A Comprehensive Study on Knowledge Graph Embedding over Relational Patterns Based on Rule Learning

Long Jin[1], Zhen Yao[1], Mingyang Chen[2], Huajun Chen[2,3], and Wen Zhang[1⋆]

[1] School of Software Technology, Zhejiang University
[2] College of Computer Science and Technology, Zhejiang University
[3] Donghai laboratory
{longjin,yz0204,mingyangchen,huajunsir,zhang.wen}@zju.edu.cn

Abstract. Knowledge Graph Embedding (KGE) has proven to be an effective approach to solving the Knowledge Graph Completion (KGC) task. Relational patterns which refer to relations with specific semantics exhibiting graph patterns are an important factor in the performance of KGE models. Though KGE models' capabilities are analyzed over different relational patterns in theory and a rough connection between better relational patterns modeling and better performance of KGC has been built, a comprehensive quantitative analysis on KGE models over relational patterns remains absent so it is uncertain how the theoretical support of KGE to a relational pattern contributes to the performance of triples associated to such a relational pattern. To address this challenge, we evaluate the performance of 7 KGE models over 4 common relational patterns on 2 benchmarks, then conduct an analysis in theory, entity frequency, and part-to-whole three aspects and get some counterintuitive conclusions. Finally, we introduce a training-free method Score-based Patterns Adaptation (SPA) to enhance KGE models' performance over various relational patterns. This approach is simple yet effective and can be applied to KGE models without additional training. Our experimental results demonstrate that our method generally enhances performance over specific relational patterns. Our source code is available from GitHub at `https://github.com/zjukg/Comprehensive-Study-over-Relational-Patterns`.

Keywords: Relational patterns · Knowledge graph embedding · Rule mining · Knowledge graph completion.

1 Introduction

Knowledge Graphs (KGs) are used to organize triples and represent various types of information about the real world. A typical triple consists of a head entity, a relation, and a tail entity, expressed in the format (h, r, t). Several well-known KG projects, including FreeBase [1], WordNet [22], YAGO [31], and

---

⋆ Corresponding author.

DBpedia [19], have gained attention for their successful use in natural language processing [41], question answering [3], recommendation systems [36], and other downstream tasks.

Despite the vast number of triples in large-scale KGs, they suffer from the problem of incompleteness. To address this problem, the Knowledge Graph Completion (KGC) task, such as link prediction, aims to predict missing triples based on known triples. Knowledge Graph Embedding (KGE) [35] has proven to be an effective approach to solving the KGC task by capturing semantic representations of entities and relations in a low-dimensional vector space.

Most KGE methods utilizing triples as learning resources derive the semantics of entities and relations from graph structures. Relations with specific semantics typically exhibit corresponding graph patterns, which we call relational patterns in this paper, such as symmetry/antisymmetry, inversion, and composition [32]. The performance of KGE models is widely regarded as being closely tied to their capacity for capturing relational patterns within the KG [45,32,28,20,4]. Previous studies have endeavored to explore whether KGE models truly learned the relational patterns among triples. Some works [42,45] use the learned relation embeddings to mine rules corresponding to different relational patterns and prove that the mined rules are of high quality, showing KGE models successfully learned the relational patterns among triples. Some works [32,46] utilize distribution histograms to reveal that embeddings of relations associated with relational patterns tend to converge towards specific positions within the vector space. Although previous studies have theoretically analyzed KGE models' capabilities in addressing various relational patterns and established a rough connection between better relational patterns modeling and better performance of KGC, a comprehensive quantitative analysis of KGE models to relational patterns remains absent. In the absence of such research, it is uncertain how the theoretical support of KGE for a specific relational pattern contributes to the prediction results of triples associated with that pattern. Consequently, quantifying KGE models' performance over particular relational patterns poses a significant challenge.

In this paper, we propose a methodology to classify triples into relational patterns based on rules mined from training data, then the capacity of KGE models in reasoning over different patterns can be quantified with the performance of triples belonging to specific patterns. We conduct numerous experiments, in theory, entity frequency, and part-to-whole aspects, to assess KGE models' performance over relational patterns, leading to the following conclusions: **1)** Theoretical support for a relational pattern in a KGE model does not guarantee superior performance compared to another KGE model lacking such support. **2)** The influence of entity frequency on the performance of different relational patterns varies. Performance for symmetric patterns diminishes as entity frequency increases, while for other patterns, performance improves with increasing frequency. **3)** If one KGE model significantly outperforms another, the superior model will exhibit better performance overall relational patterns. Conversely, when two KGE models exhibit similar overall performance, their perfor-

mance over relational patterns may diverge considerably. Lastly, we introduce a training-free method, Score-based Patterns Adaptation (SPA), to enhance KGE models' performance over various relational patterns. Our experimental results demonstrate that SPA generally improves performance over specific relational patterns.

The contributions of this paper are summarized as follows:

1. To our best knowledge, we are the first to conduct a comprehensive quantitative analysis over relational patterns;
2. We evaluate the performance of 7 KGE models over 4 common relational patterns on 2 benchmarks, then provide an analysis in theory, entity frequency, and part-to-whole three aspects, and get some counterintuitive conclusions;
3. We introduce a training-free method, Score-based Patterns Adaptation (SPA), designed to enhance KGE models' performance over various relational patterns. This approach is simple yet effective and can be applied to KGE models without additional training.

This article is structured as follows: Section 2 introduces the related work and Section 3 introduces preliminaries and background. Sections 4 and 5 are the main part of the paper, presenting our methodology, the comprehensive quantitative analysis of patterns, and SPA results. Section 6 presents conclusions and an outlook for future work.

2 Related Work

In this section, we concentrate on the related work of this paper, which encompasses the process of KGE, the definition of various relational patterns, and rule mining for relational patterns.

Knowledge Graph Embedding. Knowledge Graph Embedding (KGE) models strive to capture the semantic meanings of entities and relations by mapping them to continuous vectors, allowing for effective information retrieval and knowledge discovery. The process of KGE generally initializes the entity and relation embeddings and subsequently updates them with the score function and loss function. **Score function** measures the plausibility of a triple (h, r, t) with embeddings. These functions can be classified into two categories: translational distance based and semantic matching based models [35,16]. The translational distance based model primarily includes TransE and its extensions (such as TransH [37], TransR [21], TransD [15]), RotatE [32], and others. Semantic matching based models mainly comprise RESCAL [25], DistMult [42], ComplEx [34] and more. Table 3 presents some common score functions of KGE models. **Negative sampling** is the process of generating negative samples as most KGs predominantly contain positive triples. Negative triples are produced by corrupting a positive triple (h, r, t) through the replacement of either h or t. Established methods encompass uniform negative sampling [2], Bernoulli negative

sampling [37], and so on. **Loss functions** strive to minimize the scores of negative triples while maximizing those of positive triples. Principal loss function methods mainly include pointwise logistic loss [34], pairwise hinge loss [2], softplus loss [24], self-adversarial negative sampling loss [32] and others.

Relational Patterns. Relational patterns serve as a crucial metric for evaluating the performance of KGE models. Our comprehension of relational patterns becomes deepening over time. Wang et al. [37] highlight several mapping properties of relations that ought to be considered when embedding a knowledge graph, including **reflexive**, **one-to-many**, **many-to-one**, and **many-to-many**. Xie et al. [38] introduce a method called Type-embodied Knowledge Representation Learning (TKRL) that leverages **hierarchical entity types** to enhance the representation learning of knowledge graphs. Minervini et al. [23] incorporate **equivalence and inversion axioms** to improve the training of neural embeddings for knowledge graphs. The purpose of these axioms is to improve the accuracy and generalization abilities of neural embeddings by utilizing external background knowledge. Sun et al. [32] discuss **relational patterns** such as **symmetry/antisymmetry**, **inversion**, and **composition**. Then RotatE is proposed and demonstrates higher performance on various benchmarks. Qu et al. [28] examine the **subrelational pattern** in the context of exploring the impact of different rule patterns on knowledge graph reasoning. Cao et al. [4] suggest that **multiple relations** and propose DualE to model multiple relations using a combination of translation and rotation with greater performance. Some common relational patterns with their conditions are listed in Table 1.

Rule Mining. Rule mining can be employed to uncover non-obvious structures in data with logical rules [43]. The logical rules serve as a flexible declarative language for conveying high-level cognition [14,40], which can enhance the accuracy of reasoning or contribute to the generation of new triples [7,44]. Various rule mining methods have been developed to efficiently extract rules from large-scale knowledge graphs. The WARMR [9] and ALEPH [30] discover association rules over a limited set of queries. AMIE [11] and Ontological Path-finding [6] mine rule based on an exhaustive top-down search with pruning strategies. AMIE+ [10] improves the precision of the forecasts by using joint reasoning and type information. RARL [27] uses relatedness between predicates to improve search efficiency. These mined logical rules can correspond to the majority of relational patterns we proposed (the relationships between them will be detailed in Section 3). In this paper, we employ the latest version of AMIE called AMIE3 [18] to achieve length control and a trade-off between efficiency and quality.

3   Preliminaries and Background

Knowledge Graph. With a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$, Knowledge Graph $\mathcal{G}$ can be represented as a set of triplets $\mathcal{G} = \{(h, r, t)\}$ in which $h \in \mathcal{E}$

and $t \in \mathcal{E}$ represent the head and tail entity respectively, $r \in \mathcal{R}$ represents the relationship between h and t. A triple (h, r, t) can also be represented as $r(h, t)$. Most KGs are far from complete. KGC comes into play as a powerful application to infer missing links. For example, predicting the missing head or tail entities given (h, r) or (r, t) pairs.

Relational Patterns. We gave definitions of six key patterns [32,28] that could be written in regular rule form concerned in Table 1, including symmetric, anti-symmetric, inverse, equivalent, subrelation, and compositional patterns.

Table 1. Conditions for relational patterns and its rule formulation. $EN(r)$ refers to the set of entity pairs with $(head\ entity, tail\ entity)$ that satisfy $r(head\ entity, tail\ entity)$. $\emptyset$ refers to the empty set. The r in each row is represented as the relation belonging to the corresponding pattern. Note that the n in parentheses after the compositional indicates the number of relations in the hypothesis.

Relational Patterns	Condition	Rule Form
symmetric	$EN(r) = \{(t, h)\|(h, t) \in EN(r)\}$	$r(H, T) \leftrightarrow r(T, H)$
antisymmetric	$EN(r) \cap \{(t, h)\|(h, t) \in EN(r)\} = \emptyset$	$r(H, T) \not\leftrightarrow r(T, H)$
inverse	$EN(r') \subseteq \{(t, h)\|(h, t) \in EN(r)\}$	$r(H, T) \leftarrow r'(T, H)$
equivalent	$EN(r') = EN(r)$	$r(H, T) \leftrightarrow r'(H, T)$
subrelation	$EN(r') \subseteq EN(r)$	$r(H, T) \leftarrow r'(H, T)$
compositional(n)	$\{(h, t)\|(h, a_1) \in EN(r_1)\ldots$ $(a_n, t) \in EN(r_n)\} \subseteq EN(r)$	$r(H, T)$ $\leftarrow r_1(H, X_1), \ldots r_n(X_n, T)$

Closed-Path Rules The rule has the property of a closed path [42] if and only if the sequence in the hypotheses creates a path from the subject argument to the object argument of the conclusion predicate without any cycles or repeated nodes. Closed-path rules are a type of rule that is used to capture complex relationships between entities in KGs. A closed-path rule τ is of the form:

$$r(H, T) \leftarrow r_1(H, X_1) \wedge r_2(X_1, X_2) \wedge \ldots \wedge r_n(X_{n-1}, T) \tag{1}$$

where H, T, X_i are variables. We usually represent hypotheses $r_1(H, X_1) \wedge r_2(X_1, X_2) \wedge \ldots \wedge r_n(X_{n-1}, T)$ as the body and denote the conclusion $r(H, T)$ as the head of the rule. Rule quality could be evaluated through statistical metrics such as standard confidence, partial-close-world assumption (PCA) Confidence, and Head Coverage [11].

4   Methodology

The overall architecture of our methodology is shown in Fig. 1. Our work pipeline can be divided into the following three steps. **1)** In the training stage, train

Fig. 1. The overall architecture of our methodology. Sym, inv, mul, comp2, and comp3 are abbreviations for symmetric, inverse, multiple, compositional2, and compositional3 respectively.

the KGE model on the training set to get entities' and relations' embeddings. **2)** In the classification and analysis stage, classify triple (h, r, t) into patterns datasets based on the classification of r. Then the performance of different KGE models over different patterns is analyzed comprehensively and quantitatively through experiments on specific pattern datasets. **3)** Compute the final score by combining the SPA score and KGE score without requiring additional training.

The training of KGE models in step 1 follows the standard process, thus we omit the training details of KGE models and refer readers to the original papers of these methods. Next, we will describe Steps 2, and 3 of the architecture of our methodology in detail.

4.1   Classification of Triples

In this subsection, we focus on classifying triples into different relational pattern datasets to realize the comprehensive quantitative capabilities analysis of KGE models over various relational patterns. We first give a more detailed explanation of the classification method. Then we make an analysis of the relational pattern from the perspective of rule form and KG's property.

Classification Method The rule form of relational patterns often takes the form of Horn rules with closed paths (as seen in Table 1), which are similar to the rules mined by rule mining in KGs. Therefore, we decide to utilize rule mining to classify relations $Type(r)$ based on the rule form of distinct patterns. For

example, if we get a rule $\tau : r(H,T) \leftarrow r(T,H)$ which is similar to the rule form of symmetric pattern $r(H,T) \leftrightarrow r(T,H)$, we will get $Type(r) = \{symmetric\}$ (note that one relation may belong to more than one pattern). Classification of the triple $Type((h,r,t))$ is based on the classification of its relation r, in which $Type((h,r,t)) = Type(r)$. The performance of the KGE model over a relational pattern P can be quantified according to the performance of the set of triples belonging to this relational pattern that $\{(h,r,t)|P \in Type((h,r,t))\}$.

Relational Patterns Analysis For the relational patterns mentioned in Table 1, we conduct a more detailed analysis of six relational patterns:

1. For the antisymmetric pattern, we consider it as the negative counterpart of symmetric patterns. Since the triples in knowledge graphs are all positive, the correctness of non-existing triples is unknown. Therefore, we only consider positive patterns in the experiments.
2. Subrelation and equivalent patterns share a similar rule form. The test set is classified based on the rules obtained from the training set, with the test set used as the head of the rule to reversely get the body. However, since the training and test sets are randomly divided, the prior and posterior between them are weak. Thus, we merge these two patterns into `multiple` patterns, following the expression by Cao et al. [4].
3. In the compositional pattern, if the length of the rule is 1, it can become the previous pattern form. Some works [33,7] incorporated language biases, such as restricting the length of rules up to 3 to deal with the vast search space and self-loops. We also set the length of rules to 2 and 3 in subsequent experiments.

4.2   Score-based Patterns Adaptation (SPA)

To explicitly enhance the capability of KGE models over relational patterns, we aim to propose a simple approach to specifically promote KGE models on different patterns. Therefore, we introduce a training-free method, **Score-based Patterns Adaptation (SPA)**, which combines information from specific relational patterns and KGE score to modify the score function of models. The fundamental premise of SPA is that if the head triple is true, the triples within the body are also likely to be true with a high probability as the rule's head can be inferred from the body using established rules. Consequently, we propose to utilize the result derived from the rule's body as SPA score to enhance the inference over specific patterns. We also consider the rule confidence of rule mining as the confidence of relational patterns to measure the credibility of the modified score. The details are as follows:

Table 2 shows the SPA score with rule τ on different relational patterns. Compared to Table 1, Table 2 is refined based on the relational patterns analysis in Table 1. For the `symmetric` pattern, if we have the rule $\tau : r(H,T) \leftarrow r(T,H)$, we consider this relation r in the rule to be symmetric and can contribute to reasoning by calculating `symmetric` SPA score $f(t,r,h)$. And for the

Table 2. Correspondence between SPA score and rules in relational patterns. `Multiple` is the combination of subrelation and equivalent patterns mentioned in Table 1.

Relational Patterns	Rule τ	SPA score $S_p(h,r,t)$
Symmetric	$r(H,T) \leftarrow r(T,H)$	$f(t,r,h)$
Inverse	$r(H,T) \leftarrow r'(T,H)$	$f(t,r',h)$
Multiple	$r(H,T) \leftarrow r'(H,T)$	$f(h,r',t)$
Compositional	$r(H,T) \leftarrow r_1(H,X_1),...r_n(X_n,T)$	$Path(h,r_1...r_n,t)$

`inverse` pattern, we consider the relation r to be inverse with relation r' on rule $\tau : r(H,T) \leftarrow r'(T,H)$, and the semantic information in the body is $f(t,r',h)$ when computing the credibility of (h,r,t). Similarly to `symmetric` and `inverse` patterns, we can get SPA score $f(h,r',t)$ for the `multiple` and $Path(h,r_1...r_n,t)$ for the `compositional`.

Nevertheless, the compositional rules mined by the majority of existing open-source techniques are not readily applicable, largely due to their not chain-like structure [26]. Take the rule $\tau : r(H,T) \leftarrow r_1(X,H), r_2(X,T)$ as an example, we need first convert the body part $r_1(X,H)$ into $r_1^{-1}(H,X)$, where r_1^{-1} denotes the inverse relation of r_1, then we can obtain a compositional chain rule $r(H,T) \leftarrow r_1^{-1}(H,X), r_2(X,T)$, and see Appendix A for more details on the implementation of `Compositional` SPA.

Table 3. The details of KGE models, where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the absolute-value norm and Euclidean norm respectively. The expression of **h**,**r**,**t** in bold is denoted as their embedding vectors.

Model	$f_{kge}(h,r,t)$	$Path(h,r_1...r_n,t)$
TransE	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_1$	$-\|\mathbf{h}+\mathbf{r_1}...+\mathbf{r_n}-\mathbf{t}\|_1$
RotatE	$-\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|_2$	$-\|\mathbf{h}\circ\mathbf{r_1}...\circ\mathbf{r_n}-\mathbf{t}\|_2$
HAKE	$-\|\mathbf{h}_m\circ\mathbf{r}_m-\mathbf{t}_m\|_2 - \lambda\|\sin\left((\mathbf{h}_p+\mathbf{r}_p-\mathbf{t}_p)/2\right)\|_1$	$-\|\mathbf{h}_m\circ\mathbf{r_1}_m...\circ\mathbf{r_n}_m-\mathbf{t}_m\|_2 - \lambda\|\sin\left((\mathbf{h}_p+\mathbf{r_1}_p...+\mathbf{r_n}_p-\mathbf{t}_p)/2\right)\|_1$
ComplEx	$\text{Re}\left(h^{\top}\text{diag}(r)\bar{t}\right)$	$\text{Re}\left(h^{\top}\text{diag}(r_1)...\text{diag}(r_n)\bar{t}\right)$
DualE	$<Q_h\otimes W_r^0, Q_t>$	$<Q_h\otimes W_{r_1}^0...\otimes W_{r_n}^0, Q_t>$
PairRE	$-\|\mathbf{h}\circ\mathbf{r}^H-\mathbf{t}\circ\mathbf{r}^T\|_1$	$-\|\mathbf{h}\circ\mathbf{r_1}^H...\circ\mathbf{r_n}^H-\mathbf{t}\circ\mathbf{r_1}^T...\circ\mathbf{r_n}^T\|_1$
DistMult	$\mathbf{h}^{\top}\text{diag}(\mathbf{r})\mathbf{t}$	$\mathbf{h}^{\top}\text{diag}(\mathbf{r_1})...\text{diag}(\mathbf{r_n})\mathbf{t}$

We predict any missing fact jointly with the KGE score (Table 3) and SPA score (Table 2). For a query $(h,r,?)$, we substitute the tail entity with all candidate entities (h,r,t') and compute their KGE scores $f(h,r,t)$ and compute the SPA score $s_p(h,r,t)$ if relation r belongs to specific p patterns. There may be many relations with the relation r showing specific patterns, so we need to

consider all of them. The final scores can be presented as:

$$s(h,r,t) = s_{kge}\left(h,r,t\right) + $$
$$\lambda_p \frac{1}{\sum_{\tau \in Set_p(r)} MC_\tau} \sum_{\tau \in Set_p(r)} MC_\tau \left(s_p\left(h,r,t\right) - s_{kge}\left(h,r,t\right)\right) \quad (2)$$

where the function s_{kge} represents the score function of KGE models and $s_p(h,r,t)$ represents the SPA score of the p pattern. $Set_p(r)$ represents the p pattern rule set of relation r. λ_p represents the hyper-parameters of the p pattern, and MC_τ represents the mean confidence of the rule τ to measure the credibility of the modified score.

5 Experiments

In this section, we introduce the experiment results with analysis. Specifically, in subsection 5.1, we introduce the datasets, metrics, and implementation first. Next in subsection 5.2, we analyze the performance of the KGE models over relational patterns with three questions. Last, in subsection 5.3 and 5.4, we compare the performance of using SPA to optimize the score function on specific patterns and analyze SPA with the case study.

5.1 Evaluation Setup

Datasets. We have chosen two benchmark datasets, FB15k-237 and WN18RR, which are commonly used for evaluating Knowledge Graph Embedding (KGE) models. A summary of these datasets is presented in Appendix B.

Evaluation protocol. Following the same protocol as in TransE [2], we evaluate link prediction performance by assessing the ranking quality of each test triple. For a given triple (h,r,t) in the test set, we replace either the head (h',r,t) or the tail entity (h,r,t') with all entities and rank the candidate triples in descending order according to their scores. We select Mean Reciprocal Rank (MRR) as the evaluation metric. Furthermore, we use the "filtered" setting [2] to eliminate the reconstructed triples that are already present in the KG.

Implementation Details. In our experiments, we employ the combination of PCA confidence and HC thresholds to mine rules with AMIE3 [18] and classify relations and triples then conduct quantitative analysis. The best models are chosen by early stopping on the validation set using MRR over different patterns. As our SPA method optimizes the score function based on the embedding information from the trained KGE model without retraining or parameter adjustment, we do not modify the learning rate, margin, or other hyper-parameters. Instead, we only adjust the hyper-parameter λ for the four patterns (`symmetric`, `inverse`, `multiple`, and `compositional`). Our settings for hyper-parameter selection are as follows: the hyper-parameters of `symmetric` pattern

λ_{sym}, `inverse` pattern λ_{inv}, and `multiple` pattern λ_{sub} are adjusted within the set $\{\pm1, \pm2, \pm3, \pm4, \pm5, \pm10, \pm50, \pm100\}$, while the hyper-parameter for the `compositional` pattern λ_{comp2} is adjusted within the set $\{\pm1e{-}5, \pm1e{-}4, \pm1e{-}3, \pm0.01, \pm0.02, \pm0.05, \pm0.1, \pm0.2, \pm0.5, \pm1, \pm2, \pm5\}$. It should be noted that to ensure a fair comparison, experiments with the SPA strategy have the same hyper-parameters and implementation as the original models, except for pattern hyper-parameters. For more details about hyper-parameters in different KGE models, please refer to Appendix C.

Table 4. Statistics of the number of rules and relations for different relational patterns. #Rule, #r(P) denote the number of the mined rules and relations in pattern P $\#\{r|P \in Type(r)\}$ respectively. Sym, inv, mul, comp2, and comp3 are abbreviations for `symmetric`, `inverse`, `multiple`, and `compositional` patterns respectively. The mining results of relational patterns on FB15k-237 and WN18RR with five prescribed thresholds, where the symbol means the symbolization of different thresholds. i.e., the symbol of "PCA=0.9 and HC=0.6 thresholds" can be written as θ_1.

Confidence Thresholds			FB15k-237 / WN18RR					
Symbol	PCA	HC	# Rule	#r(sym)	#r(inv)	#r(sub)	#r(comp2)	#r(comp3)
θ_1	0.9	0.5	6k/5	6/3	4/0	14/0	49/0	111/1
θ_2	0.8	0.5	6k/10	26/3	11/0	22/0	62/0	147/3
θ_3	0.6	0.3	13k/41	28/3	26/0	32/0	96/0	203/7
θ_4	0.4	0.1	39k/84	31/3	48/0	55/0	162/1	233/10
θ_5	0.2	0.1	54k/115	31/3	67/0	73/0	185/1	233/10

5.2 Quantitative Analysis over Relational Patterns

Table 4 displays the number of rules mined in the two benchmarks at different confidence thresholds. We can obviously observe that the number of rules in FB15k-237 is much more than that in WN18RR, and there are all five relational patterns. While there are only two relational patterns in WN18RR, indicating that the data in FB15k-237 is more miscellaneous than that in WN18RR.

Following the suggestions of RUGE [12] and taking HC in the middle position, the evaluation of performance over patterns will be based on θ_2 (PCA=0.8 and HC=0.5).

Fig. 2 illustrates the heat map of relation distribution to show the overlapping between patterns. More details about the heat map can be found in Appendix D. In FB15k-237, the overlap between `compositional3` and others are close to 1, while the overlap between other relations is small, which means that relations in `compositional3` are massive. The relationships in WN18RR are simpler without `inverse` or `multiple`. Furthermore, we closely examine the rules of `compositional3` in WN18RR and find that the body of rules consists of the head relation with two symmetric relations [8], which causes the `compositional3` pattern to lose its compositional meaning.

Fig. 2. The heat map of the relations' distribution in five relational patterns. From left to right are the statistical results under FB15k-237 and WN18RR datasets.

Considering the significant overlap of `compositional3` with other patterns in FB15k-237 and its meaninglessness in WN18RR, we discard the `compositional3` pattern and conduct further experiments with the remaining four patterns.

Our analysis is driven by the form of three questions.

Q1: Does a KGE model supporting a specific relational pattern in theory achieve better link prediction results on triples related to the relational pattern compared to another KGE model that does not support such a relational pattern?

Table 5. Comparison of seven KGE models over various relational patterns with θ_2. ✓ indicates that the model supports the pattern while ✗ indicates unsupported. The numbers with Star(∗) indicate the baseline in each pattern. The percentages in parentheses represent the MRR ratios compared to the baseline while the radio of baseline in each pattern is always 100%. If the model gets better performance than the baseline, it indicates in **bold**, otherwise in gray.

Model	Relation Patterns			
	Symmetric	Inverse	Multiple	Composition2
TransE	✗(100%)∗	✓(98%)	✗(105%)	✓(119%)
RotatE	✓(98%)	✓(98%)	✗(102%)	✓(120%)
HAKE	✓(98%)	✓(96%)	✗(104%)	✓(124%)
DistMult	✓(94%)	✗(100%)∗	✗(106%)	✗(100%)∗
ComplEx	✓ (61%)	✓(49%)	✗(52%)	✗(100%)
DualE	✓(90%)	✓(93%)	✓(100%)∗	✓(117%)
PairRE	✓(105%)	✓(99%)	✓(104%)	✓(125%)

We consider the answer to be **NO**. Intuition suggests that if a model itself does not support one relational pattern, it may have a lower MRR in the

link prediction task compared to other KGE models that support the related relational pattern. However, the experimental results contradict our intuition.

Table 5 shows the comparison between seven KGE models. Due to the absence of some relational patterns in WN18RR, we only consider FB15k-237 for this experiment. In `symmetric`, the benchmark is TransE which does not support this pattern, while the other six KGE models (except PairRE) that do support it perform worse than TransE. The same result also appears in `inverse` and `multiple`. Only in `compositional2`, KGE models show significantly better performance for the supported than unsupported ones. Moreover, it is difficult to judge the capability of KGE models over patterns when comparing those that support the same pattern. For instance, in `symmetric`, all supporting models perform around 90% except for ComplEx, which also achieves 90% in `inverse`, 105% in `multiple`, and 120% in `compositional2`.

We consider that one of the reasons for this phenomenon is the complexity of the relations. The complex relatedness between relations makes relational patterns less discernible. As shown in Fig. 2, expect `compositional3`, the overlap between patterns is still not small and part overlaps values up to 50-70%. This suggests that a single pattern is insufficient to construct the semantics of relations in complex KGs. Also, the issue of sparsity presents a significant challenge for KGE models in reasoning processes [29,39]. Generally, the infrequency of an entity correlates with diminished reasoning accuracy [45]. This prompts an investigation into the extent to which entity frequency impacts relational patterns, thus leading to our second research question.

Q2: What is the impact of entity frequency for different KGE models over patterns?

Fig. 3. Statistics of triples in four patterns with entity frequency under θ_2. From left to right are the results on FB15k-237 and WN18RR. (Note that the vertical axis is exponential)

The statistics for the number of triples over patterns are shown in Fig. 3. For a triple (h, r, t), the threshold limit is the total number of occurrences of h and t in the KG. The `symmetric` pattern is significantly reduced from 0 to 5 frequency and the number of triples in the `compositional2` pattern is large with

Fig. 4. MRR of seven KGE models(TransE, RotatE, HAKE, DistMult, ComplEx, DualE, and PairRE) with varying entity frequency over patterns. The two figures in the first row, from left to right, are the experimental results of the `symmetric` pattern in FB15k-237 and WN18RR respectively. The second line from left to right is the results over `inverse`, `multiple`, and `compositional2` patterns in FB15k-237 respectively.

little change in entity frequency. In WN18RR, it only contains the `symmetric` pattern without `inverse` and `multiple` patterns. The number of `symmetric` triples is large and decreases sharply with increasing frequency. The subsequent experiments in this question will be based on the data mentioned above.

Fig. 4 illustrates the performance of seven KGE models over different patterns with varying entity frequencies on two datasets. In FB15k-237, we observe that the performance of ComplEx is significantly worse than the other models. In WN18RR, TransE, which does not support `symmetric`, is considerably worse than the other models. Intuitively, we expect that as the constraints on entity frequency increase, the KGE model should be better at learning embeddings of entities and result in better performance. However, in the `symmetric` pattern, there is a noticeable downward trend as the constraints on entity frequency increase. The performance on `inverse` (the first plot in the second row of Fig. 4) and `multiple` (the middle plot in the second row of Fig. 4) in FB15k-237 is consistent with our intuition. Performance on `compositional2` shows an initial drop followed by a rise.

We find that the impact of entity frequency on different models (except ComplEx) is similar. Additionally, different patterns (except `symmetric`) generally improve as the constraint on entity frequency increases. We consider the reason for the downward trend as the increasing entity frequency is that the `symmetric` pattern relation solution is usually unitary. For example, in RotatE [32] and HAKE [46], the symmetric relation tends to be π to realize the rotation in complex space. With the increase in the entity frequency, the relations involved in the

entities may also increase, resulting in the entity is not in the center symmetric position, which makes the model less effective.

Subsequently, we aim to examine the correlation between pattern-specific performance and overall performance in KGE inference, thereby formulating our final research question.

Q3: If one KGE model performs better than another one on link prediction, is it because it improves uniformly across different relational patterns?

Fig. 5. Comparison of seven KGE models under relational patterns with the overall performance in FB15k-237. The dotted lines in each model represent the performance under the complete test set of FB15k-237.

We compare seven KGE models over relational patterns with the full set of MRR in FB15k-237, as shown in Fig. 5. We find there are two scenarios. First, if one model demonstrates overall improvement compared to another, there will be significant enhancements in all relational patterns. For instance, when comparing ComplEx to DistMult, MRR has increased from 0.26 to 0.29, and it is evident that DistMult has improved in all patterns relative to ComplEx. Second, the relational pattern capability may be poor even when the overall performance is similar. Taking TransE, RotatE, and HAKE as examples, the overall performance of the three models is similar (0.33 for TransE, 0.333 for RotatE, and 0.329 for HAKE), but the capability of HAKE is slightly weaker than the other two models overall patterns.

5.3   Score-based Patterns Adaptation Results

We compare the effectiveness of using SPA to optimize the score function on specific pattern datasets. We observe that different KGE models with SPA achieve greater performance in various pattern datasets on FB15K237 and WN18RR. Table 6 displays the experimental result on FB15k-237.

On FB15k-237, our SPA strategy attains the best performance, demonstrating that it can effectively learn all patterns in specific pattern datasets. In the performance comparison of the `symmetric` pattern, KGE models with SPA, such

Table 6. Link prediction results of several models evaluated on FB15k-237 with the metric of MRR. w/o SPA represents using only KGE score without SPA strategy and w SPA represents using SPA with KGE score. **Bold** numbers indicate the best performance over patterns.

Model	FB15k-237							
	Sym. dataset		Inv. dataset		Sub. dataset		Comp.2 dataset	
	w/o. SPA	w. SPA	w/o. SPA	w. SPA	w/o. SPA	w. SPA	w/o. SPA	w. SPA
TransE	.282	**.301**	.592	**.649**	.500	**.546**	.402	**.403**
RotatE	.277	**.293**	.590	**.644**	.485	**.543**	**.407**	**.407**
HAKE	.278	**.288**	.581	**.633**	.495	**.559**	**.418**	**.418**
DistMult	**.264**	**.264**	.603	**.622**	.505	**.529**	**.338**	**.338**
ComplEx	.174	**.179**	**.296**	**.296**	.247	**.273**	**.339**	**.339**
DualE	.253	**.254**	.559	**.564**	.475	**.483**	.397	**.399**
PairRE	.295	**.316**	.597	**.676**	.494	**.555**	.424	**.425**

as TransE, RotatE, and PairRE, achieve a 2% improvement in MRR. For the `inverse` pattern, KGE models with SPA, such as TransE, RotatE, HAKE, and PairRE, show MRR improvements of 5-8%. For the `multiple` pattern, KGE models with SPA, such as TransE, RotatE, HAKE, and PairRE, exhibit MRR improvements of around 6%. For the `compositional2` pattern, we find that SPA gets only slight improvements in DualE and PairRE. A possible explanation for this phenomenon is that during the experiment, we calculate the contribution of the `compositional2` pattern in the form of 2-hops, which may further amplify errors and make the effect less pronounced. The experimental result of WN18RR is presented in Appendix E and is consistent with our conclusions in FB15k-237.

5.4   Case Study

Score-based Patterns Adaptation (SPA) utilizes the information from relational patterns, which is simple but effective to be applied to KGE models without further training. As Table 7 shows, we provide an intuitive demonstration for SPA. We select an example case (1980 Summer Olympics, /olympics/olympic_games/sports, ?) from the FB15k-237 and list the tail prediction rank on $s_{kge}(h, r, t)$, $s_p(h, r, t)$, and $s(h, r, t)$ based on RotatE.

Firstly, we get the rule τ: /user/jg/default_domain/olympic_games/sports(H, T) → /olympics/olympic_games/sports(H, T) with PCA = 0.83 and HC = 0.72 from test set which means relations in the head and body of rule τ with high similarity. The rank of the correct answer (Rowing) is 5,6, and 2 with different scores respectively, while confusing answers such as Artistic gymnastics, Swimming, and Volleyball are decreased in the rank of the $s(h, r, t)$, which means that SPA calculates the SPA score from the body of rules with the KGE score can achieve more accurate in reasoning.

Table 7. The rank of part entities in different score functions. The reasoning task is (1980 Summer Olympics, /olympics/olympic_games/sports, ?) from the FB15k-237, and we list the tail prediction rank in KGE score $s_{kge}(h, r, t)$, SPA score $s_p(h, r, t)$, and KGE with SPA score $s(h, r, t)$ based on RotatE. The rankings are sorted from highest to lowest with "filter" and the "**Rowing**" entity is the grounding.

Q: (1980 Summer Olympics, /olympics/olympic_games/sports, ?) A: **Rowing**		
Rank	KGE score $s_{kge}(h, r, t)$ — SPA score $s_p(h, r, t)$ — KGE and SPA $s(h, r, t)$		
1	Artistic gymnastics	Artistic gymnastics	Canoe Slalom
2	Swimming	Swimming	**Rowing**
3	Volleyball	Freestyle wrestling	Swimming
4	Freestyle wrestling	Volleyball	Artistic gymnastics
5	**Rowing**	Archery	Cycling
6	Archery	**Rowing**	Volleyball

6   Conclusion and Future Work

We study KGC tasks in KGE models based on relational patterns and analyze the result in theory, entity frequency, and part-to-whole three aspects with some counterintuitive and interesting conclusions. Firstly, theoretical backing for a KGE model's relational pattern doesn't ensure its superiority over models without such support. Secondly, entity frequency differently affects relational patterns' performance; it decreases for symmetric patterns yet increases for others with rising frequency. Finally, a significantly outperforming KGE model consistently excels across all relational patterns otherwise doesn't.

In the future, we believe that the research of KGE should not be confined to relational patterns. Greater attention should be given to analyzing the correlations between overall relations from a macro perspective, negative sampling [17,13], and loss function.

Supplemental Material Statement: Our source code, datasets and results of study with SPA are all available from GitHub at `https://github.com/zjukg/Comprehensive-Study-over-Relational-Patterns`.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings

of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250 (2008)

2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems **26** (2013)

3. Bordes, A., Weston, J., Usunier, N.: Open question answering with weakly supervised embedding models. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14. pp. 165–180. Springer (2014)

4. Cao, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q.: Dual quaternion knowledge graph embeddings. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 6894–6902 (2021)

5. Chao, L., He, J., Wang, T., Chu, W.: Pairre: Knowledge graph embeddings via paired relation vectors. arXiv preprint arXiv:2011.03798 (2020)

6. Chen, Y., Goldberg, S., Wang, D.Z., Johri, S.S.: Ontological pathfinding. In: Proceedings of the 2016 International Conference on Management of Data. pp. 835–846 (2016)

7. Cheng, K., Yang, Z., Zhang, M., Sun, Y.: Uniker: A unified framework for combining embedding and definite horn rule reasoning for knowledge graph inference. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 9753–9771 (2021)

8. Cui, W., Chen, X.: Instance-based learning for knowledge base completion. arXiv preprint arXiv:2211.06807 (2022)

9. Dehaspe, L., Toivonen, H.: Discovery of frequent datalog patterns. Data Mining and knowledge discovery **3**, 7–36 (1999)

10. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with amie+. The VLDB Journal **24**(6), 707–730 (2015)

11. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on World Wide Web. pp. 413–422 (2013)

12. Guo, S., Wang, Q., Wang, L., Wang, B., Guo, L.: Knowledge graph embedding with iterative guidance from soft rules. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)

13. Hajimoradlou, A., Kazemi, M.: Stay positive: knowledge graph embedding without negative sampling. arXiv preprint arXiv:2201.02661 (2022)

14. Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E.: Harnessing deep neural networks with logic rules. arXiv preprint arXiv:1603.06318 (2016)

15. Ji, G., He, S., Xu, L., Liu, K., 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). pp. 687–696 (2015)

16. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE transactions on neural networks and learning systems **33**(2), 494–514 (2021)

17. Kamigaito, H., Hayashi, K.: Comprehensive analysis of negative sampling in knowledge graph representation learning. In: International Conference on Machine Learning. pp. 10661–10675. PMLR (2022)

18. Lajus, J., Galárraga, L., Suchanek, F.: Fast and exact rule mining with amie 3. In: The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17. pp. 36–52. Springer (2020)

19. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web **6**(2), 167–195 (2015)
20. Li, R., Cao, Y., Zhu, Q., Li, X., Fang, F.: Is there more pattern in knowledge graph? exploring proximity pattern for knowledge graph embedding. arXiv preprint arXiv:2110.00720 (2021)
21. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29 (2015)
22. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
23. Minervini, P., Costabello, L., Muñoz, E., Nováček, V., Vandenbussche, P.Y.: Regularizing knowledge graph embeddings via equivalence and inversion axioms. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10. pp. 668–683. Springer (2017)
24. Mohamed, S.K., Novácek, V., Vandenbussche, P.Y., Muñoz, E.: Loss functions in knowledge graph embedding models. DL4KG@ ESWC **2377**, 1–10 (2019)
25. Nickel, M., Tresp, V., Kriegel, H.P., et al.: A three-way model for collective learning on multi-relational data. In: Icml. vol. 11, pp. 3104482–3104584 (2011)
26. Niu, G., Zhang, Y., Li, B., Cui, P., Liu, S., Li, J., Zhang, X.: Rule-guided compositional representation learning on knowledge graphs. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 2950–2958 (2020)
27. Pirrò, G.: Relatedness and tbox-driven rule learning in large knowledge bases. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 2975–2982 (2020)
28. Qu, M., Tang, J.: Probabilistic logic neural networks for reasoning. Advances in neural information processing systems **32** (2019)
29. Sharma, A., Talukdar, P., et al.: Towards understanding the geometry of knowledge graph embeddings. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 122–131 (2018)
30. Srinivasan, A.: The aleph manual (2001)
31. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706 (2007)
32. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
33. Suresh, S., Neville, J.: A hybrid model for learning embeddings and logical rules simultaneously from knowledge graphs. In: 2020 IEEE International Conference on Data Mining (ICDM). pp. 1280–1285. IEEE (2020)
34. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International conference on machine learning. pp. 2071–2080. PMLR (2016)
35. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
36. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.S.: Explainable reasoning over knowledge graphs for recommendation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 5329–5336 (2019)

37. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence. vol. 28 (2014)
38. Xie, R., Liu, Z., Sun, M., et al.: Representation learning of knowledge graphs with hierarchical types. In: IJCAI. vol. 2016, pp. 2965–2971 (2016)
39. Xiong, C., Power, R., Callan, J.: Explicit semantic ranking for academic search via knowledge graph embedding. In: Proceedings of the 26th international conference on world wide web. pp. 1271–1279 (2017)
40. Xu, Z., Ye, P., Chen, H., Zhao, M., Chen, H., Zhang, W.: Ruleformer: Context-aware rule mining over knowledge graph. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2551–2560 (2022)
41. Yang, B., Mitchell, T.: Leveraging knowledge bases in lstms for improving machine reading. arXiv preprint arXiv:1902.09091 (2019)
42. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
43. Zhang, W., Chen, J., Li, J., Xu, Z., Pan, J.Z., Chen, H.: Knowledge graph reasoning with logics and embeddings: survey and perspective. arXiv preprint arXiv:2202.07412 (2022)
44. Zhang, W., Chen, M., Xu, Z., Zhu, Y., Chen, H.: Explaining knowledge graph embedding via latent rule learning (2021)
45. Zhang, W., Paudel, B., Wang, L., Chen, J., Zhu, H., Zhang, W., Bernstein, A., Chen, H.: Iteratively learning embeddings and rules for knowledge graph reasoning. In: The world wide web conference. pp. 2366–2377 (2019)
46. Zhang, Z., Cai, J., Zhang, Y., Wang, J.: Learning hierarchy-aware knowledge graph embeddings for link prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 3065–3072 (2020)

Appendix

A   Implementation Details on Compositional Pattern.

The `compositional` pattern differs from other patterns in that its rule length is no longer 1. We mainly discuss rules with rule length 2. Since the rules in the `compositional` pattern are usually not close-path rules, we need to get their inverse relation and define the function of compositional inference. Our approach is mainly divided into the following two steps: 1)construction of inverse relation for all relations. 2) Define the function of the compositional inference.

Construction of inverse relation for all relations To make the experiment simpler, we do not add a new reverse triple (t, r^{-1}, h) of (h, r, t) in the dataset, but derive its inverse relation from the properties of itself. Table 8 lists all the inverse relations that need to be used in our experiment.

Table 8. Reverse Relation for different KGE models. Note that TransE, RotatE, DualE represent entities and relations into real, complex, and dual-quaternion space respectively, inverse relations of them the inverse in the corresponding space. HAKE is the combine of TransE and RotatE. The relation in DistMult and ComplEx are diagonal matrixes and the inverse of relations are the inverse of the matrixes.

Model	Reverse Relation(r^{-1})
TransE	$\mathbf{r}^{-1} = -\mathbf{r}$
RotatE	$\mathbf{r}^{-1} = \bar{\mathbf{r}}$
HAKE	$\mathbf{r}_m^{-1} = -\mathbf{r}_m \quad \mathbf{r}_p^{-1} = \bar{\mathbf{r}}$
DistMult	$\mathrm{diag}(r)^{-1} = 1/\mathrm{diag}(r)$
ComplEx	$\mathrm{diag}(r)^{-1} = 1/\mathrm{diag}(r)$
DualE	$W_r^{\diamond -1} = \overline{W_r^{\diamond}}$
PairRE	$\mathbf{r}^{H-1} = 1/r^H \quad r^{T-1} = 1/\mathbf{r}^T$

Define the function of the compositional inference The original compositional rule can be encoded into a closed-path rule by inverse relation. Table 3 defines the compositional inference function $Path(h, r_1, ...r_n, t)$, the proof of them can be listed as follows:

For TransE, r_1, r_2 can compose r, we can get $\mathbf{r}_1 + \mathbf{r}_2 = \mathbf{r}$.

Proof: If $r_1(h, e_1)$, $r_2(e_1, t)$, and $r(h, t)$, we have

$$\mathbf{h} + \mathbf{r}_1 = \mathbf{e}_1 \wedge \mathbf{e}_1 + \mathbf{r}_2 = \mathbf{t} \wedge \mathbf{h} + \mathbf{r} = \mathbf{t} \Rightarrow \mathbf{r}_1 + \mathbf{r}_2 = \mathbf{r} \tag{3}$$

For RotatE, HAKE, DualE, and PairRE, the proof of the compositional inference function $Path(h, r_1, ...r_n, t)$ of relationships between relation in head

r and relations in body $r_1, ... r_n$ can be found in works [32,46,4,5] respectively. For DistMult and ComplEx, since these two models themselves do not support `compositional` pattern, we use the method of modeling as a multiplication of matrices proposed by Yang [42] to construct compositional inference function $Path(h, r_1, ... r_n, t)$.

B Dataset Detail

Table 9 shows the detail of FB15k-237 and WN18RR.

Table 9. Statistics of datasets. The symbols #E and #R denote the number of entities and relations respectively. #TR, #VA, and #TE denote the size of the train set, validation set, and test set respectively

Dataset	#E	#R	#TR	#VA	#TE
FB15k-237	14541	237	272115	17535	20466
WN8RR	40493	11	86835	3034	3134

C The Hyper-parameter Settings.

Table 10 shows hyper-parameters settings of KGE models over different pattern datasets.

Table 10. Hyper-parameters settings of KGE models over different pattern

KGE model	λ_{sym}	λ_{inv}	λ_{mul}	λ_{comp}
TransE	-2	-2	-3	0.2
RotatE	-4	-1	-4	-0.01
HAKE	-2	-1	-3	0.1
DistMult	-2	-2	-4	1e-5
ComplEx	-2	-1	-4	-0.01
DualE	-2	-1	-3	-0.01
PairRE	-10	-2	-2	0.5

D Pattern Matrix

The relationships between relations may be highly complex, and overlapping between patterns can affect the final analysis, we propose the **Pattern Matrix**

to indicate the degree of intersection between pairs of relational patterns. Each element M_{ij} of the matrix M represents the ratio of the relation overlap of relational pattern P_i to P_j that

$$M_{ij} = \frac{|R_{P_i} \cap R_{P_j}|}{|R_{P_i}|}, \quad R_{P_k} = \{r | P_k \in Type(r)\} \tag{4}$$

where R_{P_i} represents the set of relations that belong to P_i, $|X|$ represents the length of the set X, and $\cap$ denotes the intersection of sets.

E   SPA Results on WN18RR.

Table 11 displays the experimental results on WN18RR. On WN18RR, which only has a `symmetric` pattern due to the absence of complex connections between relations like in FB15k-237, we analyze the `symmetric` pattern. With SPA, TransE achieves a 40% improvement in MRR, and by comparing the Hit@N metrics, the primary improvement is in Hit@1, which increases from 0 to 0.8 with SPA. This may be because TransE does not support the symmetric model, and there is an evident symmetric relationship in WN18RR, causing TransE to learn the symmetric relation as a **0** vector with high probability, resulting in low performance in Hit@1. Except for TransE, the less conspicuous performance of other models may also be related to the obvious symmetric relationship in the dataset itself, which enables different models that support the `symmetric` pattern to better learn symmetric relationships.

Table 11. Link prediction results of several models evaluated on the WN18RR dataset with the metric of MRR. **Bold** numbers indicate the best performance over patterns.

Model	Symmetry dataset		Hyper-parameters
	w/o. SPA	w. sym. SPA	λ_{sym}
TransE	.431	**.833**	50
RotatE	**.935**	**.935**	1
HAKE	**.942**	**.942**	1
DistMult	**.928**	**.928**	1
ComplEx	.943	**.944**	1
DualE	.938	**.939**	2
PairRE	**.939**	**.939**	1