Pattern Over-Generalization of Knowledge Graph Embedding

Anonymous ACL submission

Abstract

Knowledge graph embedding (KGE) demonstrates its effectiveness for predicting missing links in knowledge graphs (KGs) by projecting entities and relations into a low-dimensional vector space. It is crucial for KGE models to effectively capture inference patterns (patterns) inherent in KGs, such as symmetry/antisymmetry, inversion and composition. Although recent KGE models exhibit strong capabilities in modeling such diverse patterns, they suffer from inherent limitations stemming from *pattern over-generalization*, where embeddings learned from only a single pattern instance inevitably generalize that pattern to all related instances, i.e., generalize the pattern universally. To address this issue, we propose POGE (Pattern Over-Generalization Addressed Embedding), a simple but effective method that utilizes linear transformations and compound operations for relation representation. Our theoretical analysis demonstrates that a simple linear transformation allows a pattern to become progressively universal as more triples are observed in the pattern. Furthermore, after observing $d + 1$ linearly independent entities ($d + 1$ denotes the dimension of entity), the linear transformation guarantees universal generalization of the pattern across all related instances. Experimental results on three standard benchmark datasets show that POGE outperforms existing state-of-the-art KGE models in link prediction. Moreover, our empirical results indicate that POGE effectively addresses the negative impact of over-generalization.

1 Introduction

Knowledge graphs (KGs) store vast amounts of human knowledge in the form of factual triples (h, r, t), where h and t represent the head and tail entities and r denotes the relationship between entities. KGs have demonstrated their effectiveness in various downstream tasks (Wang et al., 2025; Sui et al., 2025; Ma et al., 2025). However, real-world KGs such as Freebase (Bollacker et al., 2008) and WordNet (Miller, 1995), even on a large scale, still suffer from incompleteness (Bordes et al., 2013). To address this issue, Knowledge graph embedding (KGE), which represents entities and relations in a low-dimensional vector space, has been widely studied as an effective method for predicting missing links.

A fundamental challenge in KGE lies in effectively capturing the inference patterns (patterns) inherent in KGs, such as symmetry/antisymmetry, inversion, and composition. To address this, existing works focus on designing specific score functions to capture these patterns. For instance, TransE (Bordes et al., 2013) represents relations as translations to model inversion and composition, while RotatE (Sun et al., 2018) employs rotations to capture symmetry/antisymmetry, inversion and composition. PairRE (Chao et al., 2021) and CompoundE (Ge et al., 2023) leverage scaling and compound operators to effectively model more patterns, including subrelation as well as complex relations such as 1-N, N-1, and N-N.

Despite their strong ability to capture various inference patterns, existing KGE models tend to over-generalize the patterns they observe. In particular, once a model observes a pattern, it generalizes the pattern universally across the entire graph, even when the pattern is supported by only a small number of observed triples. Consequently, patterns that are valid locally in the graph are treated as universally valid. We refer to this phenomenon as over-generalization, which leads to erroneous predictions.

To address this issue, we propose a simple but effective method, POGE, that prevents locally valid patterns from being generalized universally. POGE uses linear transformations and compound operations for relation representation, where the linear transformation is decomposed into a relation-specific orthogonal matrix and a shared upper-

triangular matrix This framework theoretically guarantees that patterns supported by only a small number of observed triples are generalized locally, while ensuring that any pattern supported by sufficient observed triples is generalized universally across the entire graph, when patterns are represented as connected relational paths.

Our contributions are as follows:

- We introduce pattern over-generalization, the phenomenon in which patterns supported by only a small number of observed triples are generalized universally across the entire graph in existing KGE models.
- We propose a novel KGE method, Pattern Over-Generalization Addressed Embedding (POGE), and theoretically guarantee that it effectively addresses over-generalization. In particular, POGE allows any pattern represented as a connected relational path to become progressively universal as more triples are observed in the pattern. Moreover, sufficient triples are observed, POGE guarantees universal generalization of the pattern across all related instances.
- Experimental results on three benchmark datasets demonstrate that POGE consistently outperforms baseline KGE models in link prediction and effectively addresses the negative impact of over-generalization.

2 Background

Knowledge Graph Embedding Given sets of entities and relations E and R, a KG can be defined as a collection of factual triples $G = \{(h, r, t)|h, t \in E, r \in R\}$, where h is head entity, t is tail entity, and r is the relation. KGE maps entities and relations to low-dimensional vectors in a latent embedding space and defines a score function to measure the plausibility of each triple. Distance-based models (DBMs) (Bordes et al., 2013; Sun et al., 2018; Chao et al., 2021; Ge et al., 2023) are trained to minimize the distance of the factual triple (h, r, t), while maximizing the distance of corrupted negative triples (h', r, t) or (h, r, t'), which are generated by randomly replacing the head h or tail t with other entities in E. PairRE (Chao et al., 2021), a representative model of DBMs, defines the score function as the distance between $h \circ r^H$ and $t \circ r^T$, i.e.,

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

where $h, t, r^H, r^T \in \mathbb{R}^d$, $\circ$ denotes a Hadamard product and $\|\cdot\|$ is a vector norm.

Tensor decomposition models (TDMs) (Yang et al., 2015; Trouillon et al., 2016; Balažević et al., 2019) are trained to maximize the plausibility (or semantic similarity) of the factual triple calculated via the multi-linear product of the head entity h, the relation r and the tail entity t, while minimizing the score of negative triples. DistMult (Yang et al., 2015), a representative model of TDMs, defines the score function as the multi-linear product of h, r, and t, i.e.,

$$f_r(h, t) = \langle h, r, t\rangle = \sum_{i=1}^{d} h_i r_i t_i, \quad (2)$$

where $h, r, t \in \mathbb{R}^d$, and $\langle\cdot,\cdot,\cdot\rangle$ denotes the three-way dot product, defined as the sum of element-wise products.

Inference Pattern Inference patterns (patterns) are widely used to analyze the generalization capabilities of KGEs. A pattern, notated as $\psi \Rightarrow \phi$, has the body ψ and the head ϕ, which are sets of triples composed of observed entities and relations in the data. For example, a composition pattern for relations $r_1, r_2, r_3 \in R$ is defined as $r_1(X, Y) \wedge r_2(Y, Z) \Rightarrow r_3(X, Z)$. A pattern implies that *if the body is in the graph, the head is also in the graph* (Pavlović and Sallinger, 2023).

3 Problem

3.1 Problem Formulation

Pattern Instance We further define a pattern instance as an instantiation of pattern $\psi \Rightarrow \phi$. For the composition pattern of relations $r_1, r_2, r_3 \in R$, a pattern instance $\psi_1 \Rightarrow \phi_1$ is expressed as $r_1(e_x, e_y) \wedge r_2(e_y, e_z) \Rightarrow r_3(e_x, e_z)$, where $r_1(e_x, e_y) \wedge r_2(e_y, e_z)$ corresponds to a body instance ψ_1, $r_3(e_x, e_z)$ corresponds to the head instance ϕ_1, and $e_x, e_y, e_z \in E$.

Pattern Over-Generalization Although well-known KGE models such as TransE (Bordes et al., 2013), RotatE (Sun et al., 2018), PairRE (Chao et al., 2021), and CompoundE (Ge et al., 2023) demonstrate strong generalization capabilities by modeling various patterns, they suffer from inherent limitations stemming from *pattern over-generalization*. Pattern over-generalization is the phenomenon where a model, after observing a single instance of a pattern $(\psi_1 \Rightarrow \phi_1)$ in the graph G, generalizes the pattern to every body instance that appears in the graph; for example, if

the body ψ_2 appears in graph G, the model infers that the corresponding head ϕ_2 must also exist (i.e., $\psi_2 \Rightarrow \phi_2$). This issue arises because existing models are trained to generalize a pattern universally. This phenomenon of *pattern over-generalization* is formally defined as follows.

Pattern Over-Generalization

The model generalizes a *local pattern* to all unseen triples, without observing sufficient triples, i.e., the model treats a *local pattern* as a *universal pattern*.

• Local Pattern: $\psi_i \in G_o \Rightarrow \phi_i \in G_p$ s.t. $G_p \subset G_u$ and $G_p \neq G_u$

• Universal Pattern: $\psi_i \in G_o \Rightarrow \phi_i \in G_p$ s.t. $G_p = G_u$

for a given set of instantiated relations $\{r_i\}_{i=1}^{n}$.

where

$$
\begin{aligned}
G_o &: \{f | f \text{ is a set of triples observed in the given KG data}\} \\
G_p &: \{f | f \text{ is } \psi_i \in G_o \text{ or } \phi_i \text{ that corresponds to } \psi_i, \text{ s.t. } \phi_i \text{ is semantically correct.}\} \\
G_u &: \{f | f \text{ is } \psi_i \in G_o \text{ or } \phi_i \text{ that corresponds to } \psi_i)\}
\end{aligned} \tag{3}
$$

While this phenomenon can serve as a crucial inductive bias in KGE, universally generalizing a local pattern without sufficient observations can lead to erroneous predictions by injecting incorrect information into the embeddings.

3.2 Cause and Evidence

Why Does The Problem Appear? The cause is that the pattern condition depends only on relation embeddings in KGE methods. This entity-independent pattern condition allows the model to generalize local patterns to unseen triples.

For example, in PairRE, if body instance $(e_{x_1}, r_1, e_{y_1}), (e_{y_1}, r_2, e_{z_1}) \in G$ and head instance $(e_{x_1}, r_3, e_{z_1}) \in G$, we have

$$
\begin{aligned}
e_{x_1} \circ r_1^H = e_{y_1} \circ r_1^T \ \wedge \ e_{y_1} \circ r_2^H = e_{z_1} \circ r_2^T \\
\wedge \ e_{x_1} \circ r_3^H = e_{z_1} \circ r_3^T \\
\Rightarrow \quad r_1^T \circ r_2^T \circ r_3^H = r_1^H \circ r_2^H \circ r_3^T
\end{aligned} \tag{4}
$$

Under this pattern condition, if a new body instance $(e_{x_2}, r_1, e_{y_2}), (e_{y_2}, r_2, e_{z_2})$ is observed as

$$
e_{x_2} \circ r_1^H = e_{y_2} \circ r_1^T \ \wedge \ e_{y_2} \circ r_2^H = e_{z_2} \circ r_2^T \tag{5}
$$

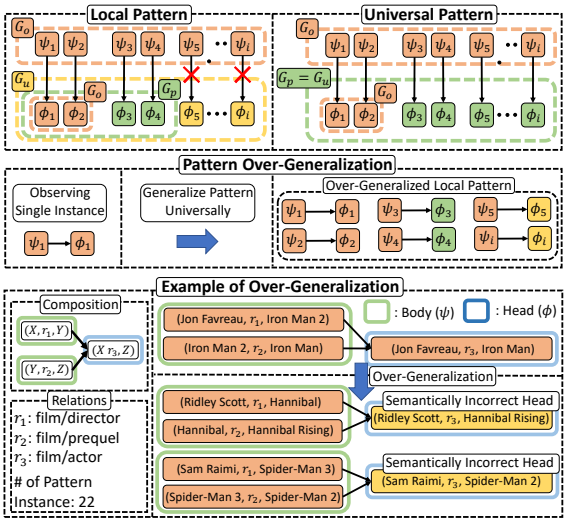


Figure 1: Illustration of local and universal patterns, and the process and examples of over-generalization. Existing models suffer from over-generalization by treating local patterns as universal patterns. The example shows that a local pattern is generalized universally, which leads to erroneous predictions.

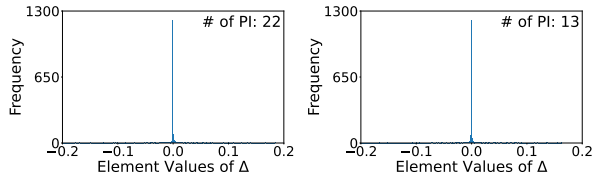
then the model guarantees that

$$
e_{x_2} \circ r_3^H = e_{z_2} \circ r_3^T, \tag{6}
$$

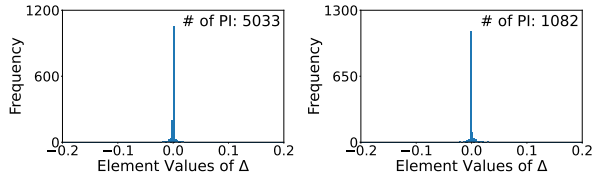
thereby leading the model to treat the corresponding head as valid for every new body instance of the same pattern. This phenomenon is further illustrated in Figure 1, where the model generalizes the pattern universally to every body instance.

However, not all patterns in real-world KGs are universally valid. Some patterns are observed across many instances in G, indicating that they represent universal patterns. In contrast, other patterns are observed across few instances in G, indicating that they represent local patterns.

Empirical Evidence Despite this fundamental difference, existing KGE models overlook the difference between local and universal patterns, treating all observed patterns as universally valid regardless of their frequency. Figure 2 shows the histograms of the embedding difference $\Delta = r_1^T \circ r_2^T \circ r_3^H - r_1^H \circ r_2^H \circ r_3^T$, that is presented in Equation 4. Elements of Δ close to zero indicate that the model recognizes the given relation set (r_1, r_2, r_3) as a valid pattern, therefore, the model generalizes the pattern universally to every body instance. Figures 2(a) and 2(b) show that the elements of Δ are concentrated near zero for both local and universal patterns, indicating that the model recognizes both as valid composition patterns re-



(a) Histograms of local patterns that are supported by scarce pattern instances. The relation sets for the left and right figures are *(film/written_by, actor/film, film/prequel)* and *(film/director, film/prequel, actor/film)*, respectively



(b) Histograms of universal patterns that are supported by many pattern instances. The relation sets for the left and right figures are *(actor/film, film/country, people/nationality)* and *(people/place_of_birth, location/country, people/nationality)*, respectively

Figure 2: Histograms of embedding difference $\Delta = r_1^T \circ r_2^T \circ r_3^H - r_1^H \circ r_2^H \circ r_3^T$ for different relation set (r_1, r_2, r_3). # of PI denotes the number of pattern instances. (r_1, r_2, r_3) are retrieved from FB15k-237.

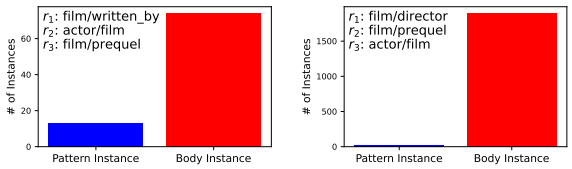


Figure 3: The number of pattern instances and body instances for the local patterns introduced in Figure 2(a).

gardless of instance frequency. This demonstrates that PairRE is trained to generalize local patterns as if they were universal, even when the supporting instances are scarce.

While local patterns are supported by only a scarce number of pattern instances, they often have a vast number of body instances. Figure 3 presents the number of pattern instances and body instances of the local patterns displayed in Figure 2(a). This indicates that a large number of body instances are affected by only a few pattern instances, leading the model to predict the corresponding head instances as valid for all body instances.

To address this issue, we propose a novel KGE framework POGE that explicitly models the distinction between pattern universality and locality. Our core idea is to generalize patterns differentially based on their observation frequency in G, rather than generalizing all patterns equally.

Model	Score Function	Over-generalization		
		Sym/Asym	Inversion	Composition
TransE	$\|h + r - t\|$	-/✓	✓	✓
RotatE	$\|h \circ r - t\|$	✓/✓	✓	✓
Rotae3D	$\|h \odot r - t\|$	✓/✓	✓	✓
PairRE	$\|h \circ r^H - t \circ r^T\|$	✓/✓	✓	✓
CompoundE	$\|M_r \cdot h - M_r \cdot t\|$	✓/✓	✓	✓
POGE (Ours)	$\|L_r h_r - t_r\|$	✗/✗	✗	✗

Table 1: Comparison between POGE and KGE models. h and t denote head and tail embeddings and h_r and t_r indicate head and tail embeddings in the relation-specific space, as presented in Equation 10.

Which Methods Are Affected? Table 1 presents representative examples of KGE methods that suffer from over-generalization. In contrast to other models that suffer from over-generalization (indicated by checkmarks), our model addresses this issue by utilizing the relation-specific linear transformation in the score function.

4 Method

In this section, we present the formulation of POGE and provide a theoretical analysis demonstrating how the proposed method effectively addresses over-generalization.

4.1 Pattern Over-Generalization Addressed Embedding (POGE)

Final Form We define the score function as the distance between the head entity h_r and tail entity t_r in relation-specific space, after the linear transformation $L_r \in \mathbb{R}^{d \times d}$:

$$f_r(h, t) = \|L_r h_r - t_r\| \tag{7}$$

where $h_r, t_r \in \mathbb{R}^d$ denote the head and tail embeddings in relation-specific space, respectively.

QR Decomposition and Partial Sharing for Efficient Parameterization Although existing models such as RESCAL (Nickel et al., 2011) and TransR (Lin et al., 2015) employ linear transformations, they suffer from overfitting or are not designed to model inference patterns. Furthermore, representing relations as $\mathbb{R}^{n \times n}$ linear transformations incurs significant computational costs. To address this limitation, inspired by QR decomposition, we decompose L_r into a relation-specific orthogonal matrix Q_r and an upper-triangular matrix R that is shared across all relations:

$$\begin{aligned} L_r &= Q_r R \\ Q_r &= H_1 H_2 ... H_k \end{aligned} \tag{8}$$

In addition, Q_r is approximated using a product of k Householder reflections (where $k \ll d$). This

approximation significantly reduces the number of parameters from $n_r d^2$ to $d(d+1)/2 + n_r k d$, where n_r denotes the number of relations, thereby effectively reducing the model complexity. Details about computational complexity with respect to k are presented in Appendix C.

Additionally, let $\bar{R}$ be the learnable upper-triangular parameter matrix. The final shared matrix R is formulated as:

$$R = \frac{\bar{R}}{\|\bar{R}\|_2} \tag{9}$$

where $\|\cdot\|_2$ denotes the spectral norm. We argue that even for local patterns, the pattern should be generalized to entities that are not observed in the patterns but are semantically similar to entities that are observed in the patterns. Spectral Normalization (SN) enables this generalization by bounding the Lipschitz constant of the transformations to one (Miyato et al., 2018). Detailed derivations are provided in the Appendix B.

Relation-Specific Affine Mapping for Expressive Power Sharing an upper-triangular matrix R reduces the expressive power of relation-specific transformations. To address this limitation and enhance the model capacity, following (Ge et al., 2023), each entity is mapped into an relation-specific space via three affine operators before applying L_r. By employing homogeneous coordinates, these operators can be unified into a single matrix multiplication:

$$\begin{aligned} h_r &= M_r h, & t_r &= \hat{M}_r \cdot t \\ M_r &= S_r \cdot R_r \cdot T_r, & \hat{M}_r &= \hat{S}_r \cdot \hat{R}_r \cdot \hat{T}_r \end{aligned} \tag{10}$$

where h, t are head and tail embeddings, S_r, R_r, and T_r denote the scaling, rotation, and translation operators, and $\hat{S}_r, \hat{R}_r$, and $\hat{T}_r$ denote the scaling, rotation, and translation operators for tail entity embedding, respectively. This mapping strategy ensures that each relation has sufficient expressive power despite the shared components in L_r.

Optimization Following Sun et al. (2018), we adopt self-adversarial negative sampling for training. The loss function can be written as:

$$\begin{aligned} L = &-\log \sigma(\gamma - f_r(h,t)) \\ &- \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(f_r(h'_i, t'_i) - \gamma) \end{aligned} \tag{11}$$

where σ is the sigmoid function, γ is a fixed margin, (h'_i, r, t'_i) is the i-th negative triple and $p(h'_i, r, t'_i)$ is the weight of the negative triple, defined as:

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(h'_j, t'_j)}{\sum_i \exp \alpha f_r(h'_i, t'_i)} \tag{12}$$

where α is the temperature of sampling.

4.2 How Is Pattern Over-Generalization Addressed?

To analyze how POGE addresses over-generalization, we first consider a simple case using only linear transformation L_r, and then extend this to our framework, which incorporates the affine operators.

Theoretical Analysis: Linear Transformation To the best of our knowledge, all patterns studied in existing research are based on connected paths formed by relations. This implies that a body (ψ) and head (ϕ) can be represented as a relational path between the start entity e_u and the end entity e_v, where each path is formulated as a product of linear matrix multiplications. Consequently, the body (ψ) and head (ϕ) of a pattern can be expressed as:

$$\begin{aligned} \text{Body } (\psi): &\quad L_\psi e_u = L_{r_n} \dots L_{r_2} L_{r_1} e_u = e_v, \\ \text{Head } (\phi): &\quad L_\phi e_u = L_{r'_m} \dots L_{r'_2} L_{r'_1} e_u = e_v. \end{aligned} \tag{13}$$

where $L \in \mathbb{R}^{d \times d}$ and $e \in \mathbb{R}^d$ denote the transformation matrix and entity vector, respectively.

From Equation 13, since both paths map e_u to the same entity e_v, we explicitly have $L_\psi e_u = L_\phi e_u$, which is equivalent to:

$$(L_\psi - L_\phi) e_u = 0 \quad \Longleftrightarrow \quad E e_u = 0 \tag{14}$$

where $E = L_\psi - L_\phi$ denotes the constraint matrix. Next, consider a set of d linearly independent entities $\{e_{u_1}, e_{u_2}, \dots, e_{u_d}\}$ that satisfy the pattern, such that:

$$E e_{u_i} = 0, \quad \text{for all } i = 1, 2, \dots, d \tag{15}$$

For any arbitrary entity $a \in \mathbb{R}^d$, since $\{e_{u_i}\}$ forms a basis in $\mathbb{R}^d$, a can be expressed as a linear combination $a = c_1 e_{u_1} + c_2 e_{u_2} + \dots + c_d e_{u_d}$. By the linearity of the transformation E, it follows that:

$$E a = c_1 E e_{u_1} + c_2 E e_{u_2} + \dots + c_d E e_{u_d} = 0 \tag{16}$$

This result implies that POGE enables any pattern to become progressively universal as more linearly independent entity e_u are observed. Additionally, when the number of observed entities reaches d, POGE guarantees the universal generalization of the pattern across all related instances.

Extension to Relation-Specific Affine Mapping This analysis can be extended to our proposed framework by employing homogeneous coordinates. By representing entities in an augmented $(d+1)$-dimensional space, the integration of affine operators and linear transformations for a relation r can be unified into a single linear matrix $A_r \in \mathbb{R}^{(d+1)\times(d+1)}$ when $\hat{M}_r$ is non-singular:

$$A_r = \hat{M}_r^{-1} L_r M_r \tag{17}$$

Therefore, the relational path can be expressed as a product of linear transformation A_r. Consequently, the same proof used in the linear case can be applied, demonstrating that the pattern becomes universal only when $d+1$ linearly independent entities are observed in the augmented space. [1] The linear independence of entity embeddings is discussed in Section 6.3.

5 Related Work

5.1 Distance-based Model

TransE (Bordes et al., 2013), RotatE (Sun et al., 2018) and Rotate3D (Gao et al., 2020) represent relations as translation, rotation and 3D rotation between entities, respectively. To model semantic hierarchies, HAKE (Zhang et al., 2020) maps entities into the polar coordinate system. DualE (Cao et al., 2021) models relations as a combination of translation and rotation to satisfy the key pattern and the multiple relations pattern. PairRE (Chao et al., 2021) applies a scaling operation to both head and tail entities. DFieldE (Nayyeri et al., 2021) embeds entities on the trajectories of Ordinary Differential Equations and represents each relation as a vector field on manifolds to preserve complex structures. HopfE (Bastos et al., 2021) models the structural embedding in 3D Euclidean space, and maps the entity embedding from a 3D Euclidean space to a 4D hypersphere using the inverse Hopf Fibration. DensE (Lu et al., 2022) decomposes each relation into rotation and scaling operations in 3D Euclidean space. ReflectE (Zhang et al., 2022) regards each relation as a Householder transformation to model complex relations. TranSHER (Li et al., 2022) utilizes relation-specific translation to address the constraint of hyper-ellipsoid restrictions. ExpressivE (Pavlović and Sallinger, 2023) embeds entities as points and each relation as hyper-parallelograms in the latent space, while Octagon Embedding (Charpenay and Schockaert, 2024) embeds each relation as composed of axis-aligned octagons. CompoundE (Ge et al., 2023) applies compound operators, including translation, rotation, and scaling to both head and tail entities. SpeedE (Pavlović and Sallinger, 2024) designs a Euclidean geometric KGE that is efficient under low-dimensional conditions. To reduce the computational cost of rotation-based methods, OrthogonalE (Zhu and Shimodaira, 2024) employs matrices for entities and block-diagonal orthogonal matrices with Riemannian optimization for relations.

5.2 Tensor Decomposition Model

DistMult (Yang et al., 2015) embeds entities as vectors and each relation as a diagonal matrix, computing the score via the product of entities and relation. ComplEx (Trouillon et al., 2016) extends DistMult to complex embeddings. HolE (Nickel et al., 2016) uses the circular correlation of entity embeddings to capture rich interactions between entities and relation. ANALOGY (Liu et al., 2017) models relations as normal linear operators, and utilizes the commutativity property of these matrices to capture analogical structures. SimplE (Kazemi and Poole, 2018) adapts Canonical Polyadic decomposition to allow independent embeddings for both head and tail entities. TuckER (Balažević et al., 2019) employs Tucker decomposition to capture interactions between entity and relation embeddings. QuatE (Zhang et al., 2019) extends the rotation to 4D space using quaternion and the Hamilton product.

6 Experiments

6.1 Experimental Setting

Dataset We evaluate POGE on three widely used KG datasets: WN18RR (Dettmers et al., 2018), FB15k-237 (Toutanova and Chen, 2015) and YAGO3-10 (Mahdisoltani et al., 2013). The statistics of the three benchmarks are presented in Appendix E

Evaluation Protocol We evaluate link prediction performance in the filtered setting (Bordes et al., 2013). In this setting, test triples are ranked against all other candidate triples that are generated by corrupting subjects or objects: (h', r, t) or (h, r, t'), and all the triples that appear either in the training, validation or test set are removed from the candidate triples, except the test triple of interest. We adopt MRR, Hits@1, and Hits@10 to compare

[1] Refer to Appendix B for constraint matrices for various patterns.

Model	WN18RR			FB15k-237			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE	.226	-	.501	.294	-	.465	-	-	-
DistMult	.430	.390	.490	.241	.155	.419	-	-	-
ComplEx	.440	.410	.510	.247	.158	.428	-	-	-
RotatE	.476	.428	.571	.338	.241	.533	.495	.402	.670
TuckER	.470	.443	.526	.358	.266	.544	-	-	-
QuatE	.488	.438	.582	.348	.248	.550	-	-	-
Rotate3D	.489	.442	.579	.347	.250	.543	-	-	-
HAKE	.497	.452	.582	.346	.250	.542	.545	.462	.694
DualE	.492	.444	.584	.365	.268	.559	-	-	-
PairRE	-	-	-	.351	.256	.544	-	-	-
DFieldE	.48	.44	.57	.36	.27	.55	.51	.41	.68
HopfE	.472	.413	.586	.343	.247	.534	.529	.438	.695
DensE	.492	-	.586	.351	-	.544	.541	-	.678
ReflectE	.488	.450	.559	.358	.263	.546	-	-	-
TranSHER	-	-	-	.360	.264	.551	.490	.404	.647
ExpressivE	.482	.407	**.619**	.350	.256	.535	-	-	-
CompoundE	.491	.450	.576	.357	.264	.545	-	-	-
SpeedE	.493	.446	-	.320	.227	-	.413	.332	-
OctagonE	.479	.436	.561	.332	.241	.517	-	-	-
OrthogonalE	.494	.446	.573	.334	.242	.518	-	-	-
POGE	**.506** ±.001	**.461** ±.001	.595 ±.000	**.369** ±.001	**.273** ±.001	**.562** ±.001	**.556** ±.000	**.474** ±.001	**.699** ±.000

Table 2: Link prediction results on WN18RR, FB15k-237 and YAGO3-10. Bold indicates the best result and underline indicates the second best result. ± indicates standard deviation.

Model	MRR		
	WN18RR	FB15k-237	YAGO3-10
POGE	.506	.369	.556
+ w/o R	.505	.365	.517
+ w/o Q_r	.501	.364	.540
+ w/o SN	.505	.360	.523
+ w/o L_r	.491	.357	.477

Table 3: Ablation study of POGE on WN18RR, FB15k-237 and YAGO3-10. MRR is used for performance comparison. R, Q_r, SN, and L_r denote the shared upper-triangular parameter matrix, relation-specific Householder reflection, Spectral Normalization, and linear transformation respectively

the performance of different KGE models. MRR denotes the mean reciprocal rank of the correct entities, and Hits@N represents the proportion of correct entities ranked within the top N.

Baselines We compare POGE with KGE baselines including TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2018), TuckER (Balažević et al., 2019), QuatE (Zhang et al., 2019), HAKE (Zhang et al., 2020), Rotate3D (Gao et al., 2020), DualE (Cao et al., 2021), PairRE (Chao et al., 2021), DFieldE (Nayyeri et al., 2021), HopfE (Bastos et al., 2021), DensE (Lu et al., 2022), ReflectE (Zhang et al., 2022), TranSHER (Li et al., 2022), ExpressivE (Pavlović and Sallinger, 2023), CompoundE (Ge et al., 2023), SpeedE (Pavlović and Sallinger, 2024), Octagon Embedding (Charpenay and Schockaert, 2024) and OrthogonalE (Zhu and Shimodaira, 2024).

6.2 Main Results

Link Prediction Performance As shown in Table 2, POGE exhibits superior or competitive performance compared with the baselines. Specifically, POGE outperforms other baselines on FB15k-237 and YAGO3-10. On WN18RR, it achieves the best results in all metrics except for Hits@10. These results indicate the effectiveness and robustness of POGE across diverse datasets.

Ablation Study Table 3 summarizes the results of an ablation study conducted to verify the effectiveness of each proposed component. As shown in the results, POGE consistently outperforms the ablated models across all datasets. This result demonstrates that every component of POGE contributes to the effectiveness of the proposed model.

6.3 Analysis

Entity Independence As discussed in Section 4.2, POGE ensures that any pattern becomes progressively universal as more linearly independent entities are observed in the pattern. This implies that if the entity embeddings trained by POGE are linearly independent, POGE can achieve such progressive universality in practice. To investigate entity independence, we randomly sample entity embeddings trained by POGE and compute the rank of the space spanned by the sampled enti-

Number of Sample Vector	Rank WN18RR	FB15k-237	YAGO3-10
100	100.0	100.0	100.0
200	200.0	200.0	200.0
500	500.0	500.0	500.0
1,000	994.4 $\pm$ 1.5	1000.0	999.5 $\pm$ 0.5
1,500	1000.0	1497.4 $\pm$ 0.6	1000.0
2,000	1000.0	1500.0	1000.0

Table 4: Mean rank of the subspace spanned by randomly sampled entities over 100 random trials across three benchmarks. The entity dimensions of POGE are 1,000 on WN18RR and YAGO3-10, and 1,500 on FB15k-237.

# of Pattern Instances (n)	WN18RR # of Patterns	WN18RR Prop. (%)	FB15k-237 # of Patterns	FB15k-237 Prop. (%)	YAGO3-10 # of Patterns	YAGO3-10 Prop. (%)
n = 1	17	48.6	1,546	26.3	56	17.6
$1 < n \leq 10$	14	40.0	2,288	39.0	112	35.2
$10 < n \leq 10^2$	4	11.4	1,439	24.5	91	28.6
$10^2 < n \leq 10^3$	-	-	489	8.3	53	16.7
$n > 10^3$	-	-	111	1.9	6	1.9
Total	35	100%	5,873	100%	318	100%

Table 5: Distribution of composition patterns according to the number of pattern instances (n) across three benchmark datasets. # of Pattern Instances and # of Patterns indicate the number of pattern instances and the number of patterns, respectively. Prop. (%) is calculated as the number of patterns within each range of n divided by the total number of patterns, within each dataset.

ties. Table 4 shows the mean rank of the subspace spanned by sampled entities over 100 random trials. We empirically observe that the rank of the space spanned by the randomly sampled entities is approximately equal to the number of sampled entities, demonstrating that the sampled entities are linearly independent. In addition, with 1,000 sampled entities, the mean rank is 994.4 in WN18RR, and 999.5 in YAGO3-10. These results indicate that, since entities in practice are shown to be linearly independent, the pattern becomes increasingly universal as the number of observed entities increases, and that universal generalization is achieved when around $d + 1$ entities are observed.

Distribution of Patterns by Number of Pattern Instances Table 5 presents the distribution of composition patterns according to the number of their pattern instances n across three KG benchmarks. We compute the number of composition patterns as the number of relation sets $(r_x, r_y, r_z) \in R$ that have at least one observed composition pattern instance $(\psi_1 \Rightarrow \phi_1)$ in the KG. In WN18RR, FB15k-237 and YAGO3-10, 88.6%, 65.3% and 52.8% of composition patterns have 10 or fewer pattern instances, respectively. This result indicates that a substantial proportion of patterns in real-world KGs are observed in only a few instances. These patterns can be generalized universally when

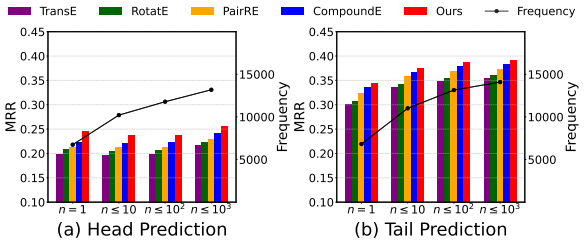


Figure 4: MRR comparison between POGE and baseline models across different local pattern definitions on FB15k-237. The black line indicates the number of test triples of G_{over}.

a model suffers from over-generalization. The distributions of symmetry/antisymmetry and inversion are presented in Appendix D.

Quantified Impact of Over-generalization To investigate the impact of over-generalization, we extract G_{over}, a set of triples (h, r, t), where candidates (h, r, t') or (h', r, t) (with $t' \neq t$ and $h' \neq h$) are heads of pattern instances whose bodies are in the training set, i.e., candidates (h, r, t') or (h', r, t) are $\phi_i \in G_u \setminus G_o$ for which there is a corresponding $\psi_i \in G_o$. Intuitively, if the model suffers from over-generalization, the rank of (h, r, t) is lower than the rank of the candidate triples. To extract G_{over}, we only consider three representative inference patterns (symmetry, inversion, composition) where the number of pattern instances is $n = 1$, $n \leq 10$, $n \leq 10^2$, and $n \leq 10^3$. We compare the MRR of POGE with other baseline models: TransE, RotatE, PairRE, and CompoundE on this extracted triple set. Figure 4 presents the results on FB15k-237. We observe that POGE consistently outperforms the baselines, regardless of the number of pattern instances. These results demonstrate that POGE effectively addresses the negative impact of over-generalization. [2]

7 Conclusion

In this paper, we propose POGE, a novel KGE method that utilizes linear transformations and compound operations. POGE is theoretically guaranteed to address over-generalization, in which a model generalizes a pattern to every body instance in the graph after observing only a single instance. Experimental results on three benchmark datasets demonstrate the effectiveness of our proposed method.

[2]The detailed definitions of G_{over} and the comparison results for WN18RR and YAGO3-10 are presented in Appendix D.

Limitations

To universally generalize patterns, POGE does not utilize the semantics of patterns, which can be a useful inductive bias for pattern generalization. In future work, we will leverage the semantics of patterns for pattern generalization.

References

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194.

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

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

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

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual quaternion knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6894–6902.

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

Victor Charpenay and Steven Schockaert. 2024. Capturing knowledge graphs and rules with octagon embeddings. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 3289–3297.

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

Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 385–394.

Xiou Ge, Yun Cheng Wang, Bin Wang, and C-C Jay Kuo. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6947–6965.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31.

Narayanan Asuri Krishnan and Carlos R Rivero. 2024. A method for assessing inference patterns captured by embedding models in knowledge graphs. In *Proceedings of the ACM Web Conference 2024*, pages 2030–2041.

Yizhi Li, Wei Fan, Chao Liu, Chenghua Lin, and Jiang Qian. 2022. Transher: Translating knowledge graph embedding with hyper-ellipsoidal restriction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8517–8528.

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

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *International conference on machine learning*, pages 2168–2178. PMLR.

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

Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025. Large language models meet knowledge graphs for question answering: Synthesis and opportunities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24589–24608, Suzhou, China. Association for Computational Linguistics.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

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

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

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

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 809–816.

Aleksandar Pavlović and Emanuel Sallinger. 2023. Expressive: A spatio-functional embedding for knowledge graph completion. In *The Eleventh International Conference on Learning Representations*.

Aleksandar Pavlović and Emanuel Sallinger. 2024. Speede: Euclidean geometric knowledge graph embedding strikes back. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 69–92.

Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. 2025. Fidelis: Faithful reasoning in large language models for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8315–8330.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Shijie Wang, Wenqi Fan, Yue Feng, Lin Shanru, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. 2025. Knowledge graph retrieval-augmented generation for llm-based recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27152–27168.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.

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

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

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3065–3072.

Yihua Zhu and Hidetoshi Shimodaira. 2024. Block-diagonal orthogonal relation and matrix entity for knowledge graph embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16956–16972.

A Implementation Details

For the experiments, we adopt the hyperparameter settings from RotatE (Sun et al., 2018) for WN18RR and YAGO3-10, and from PairRE (Chao et al., 2021) for FB15k-237. Additionally, for POGE, the number of Householder reflections k is selected from $\{2, 4, 8, 12, 20\}$. Our presented results represent the mean of three independent runs for each dataset. Furthermore, Scaling S_r and Rotation R_r are used for FB15k-237 and YAGO3-10, whereas Translation T_r and Rotation R_r are used for WN18RR. Finally, following Rotate3D (Gao et al., 2020), an L_2 regularizer is applied to entity embeddings for WN18RR.

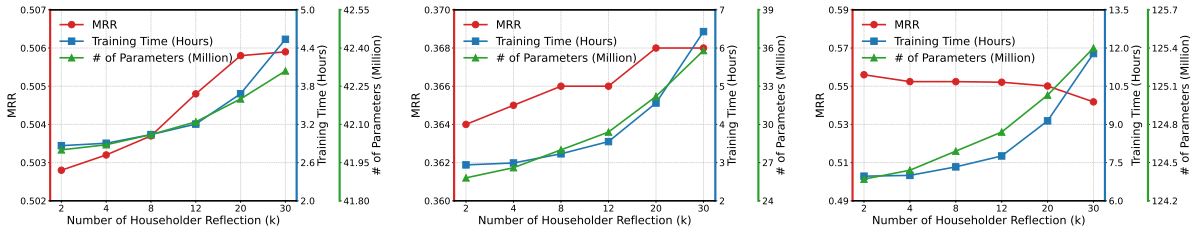
B Local Pattern Generalization and Constraint Matrices for Patterns

Bounding Lipschitz Constants for Local Pattern Generalization to Unobserved Entities By applying Spectral Normalization to the shared matrix R, we ensure that the spectral norm of each relation-specific linear transformation is bounded: $\|L_r\|_2 \leq 1$. Since the constraint matrix E is defined as $L_\psi - L_\phi$, the spectral norm of E is also bounded by the triangle inequality:

$$\|E\|_2 = \|L_\psi - L_\phi\|_2 \leq \|L_\psi\|_2 + \|L_\phi\|_2 \leq 2$$

This bound ensures that the transformation defined by the constraint matrix is Lipschitz continuous. For an entity e_{obs} that is known to satisfy the pattern (i.e., $\|Ee_{obs}\| \approx 0$) and a semantically similar but unobserved entity e_{unobs}, the pattern error $\|Ee_{unobs}\|$ for e_{unobs} is bounded as follows:

$$\|Ee_{unobs}\| \leq \|E\|_2 \|e_{unobs} - e_{obs}\| + \|Ee_{obs}\|$$

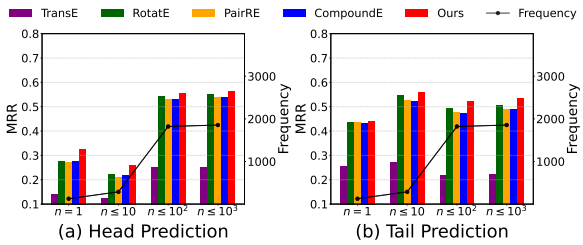


(a) WN18RR (b) FB15k-237 (c) YAGO3-10

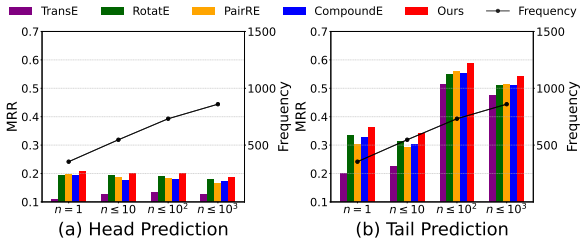
Figure 5: MRR, training time, and number of parameters of POGE on three benchmark datasets.

Pattern	Constraint Matrices ($E = A_\psi - A_\phi$)
Hierarcy $r_1(X,Y) \Rightarrow r_2(X,Y)$	$(A_{r_1} - A_{r_2})X = 0$
Symmetry $r(X,Y) \Rightarrow r(Y,X)$	$(A_r^2 - I)X = 0$
Antisymmetry $r(X,Y) \Rightarrow \neg r(X,Y)$	$(A_r^2 - I)X \neq 0$
Inversion $r_1(X,Y) \Rightarrow r_2(Y,X)$	$(A_{r_2}A_{r_1} - I)X = 0$
Intersection $r_1(X,Y) \wedge r_2(X,Y) \Rightarrow r_3(X,Y)$	$(A_{r_1} - A_{r_3})X = (A_{r_2} - A_{r_3})X = 0$
Transitivity $r(X,Y) \wedge r(Y,Z) \Rightarrow r(X,Z)$	$(A_r^2 - A_r)X = 0$
Composition $r_1(X,Y) \wedge r_2(Y,Z) \Rightarrow r_3(X,Z)$	$(A_{r_2}A_{r_1} - A_{r_3})X = 0$
Gen. Intersection $r_1(X,Y) \wedge r_1(Y,X) \Rightarrow r_2(X,Y)$	$(A_{r_1}A_{r_2} - I)X = (A_{r_1} - A_{r_2})X = 0$
B. Transitive $r(Y,Z) \wedge r(Z,X) \Rightarrow r(X,Y)$	$(A_r^3 - I)X = 0$
Equality $r(X,Z) \wedge r(Y,Z) \Rightarrow r(X,Y)$	$(I - A_r)X = 0$
B. Composition $r_1(Y,Z) \wedge r_2(Z,X) \Rightarrow r_3(X,Y)$	$(A_{r_2}A_{r_1}A_{r_3} - I)X = 0$
Commonality $r_1(X,Z) \wedge r_2(Y,Z) \Rightarrow r_3(X,Y)$	$(A_{r_2}^{-1}A_{r_1} - A_{r_3})X = 0$

Table 6: Constraint matrices corresponding to various inference patterns. The patterns are presented in (Krishnan and Rivero, 2024)



(a) WN18RR



(b) YAGO3-10

Figure 6: MRR comparison between POGE and baseline models across different local pattern definitions on three benchmarks: WN18RR (a), and YAGO3-10 (b). The black line indicates the number of test triples for each group.

As shown in the inequality, if the distance $\|e_{unobs} - e_{obs}\|$ is small, the error $\|Ee_{unobs}\|$ remains small. This mathematically guarantees that the model generalizes the learned pattern from observed entities to semantically similar entities with similar embeddings.

Constraint Matrices for Various Patterns Table 6 summarizes the derived constraint matrices for various patterns widely used in KG. Note that for patterns having multiple paths (e.g., Intersection), E represents a set of matrices $\{E_1, E_2, \dots\}$ to be satisfied simultaneously.

C Computational Complexity

Figure 5 presents computational complexity and performance with respect to Householder reflection k. In WN18RR and FB15k-237, performance improves as k increases but shows no significant improvement after $k = 20$. In YAGO3-10, the MRR is highest at $k = 2$ and decreases as k increases. These results suggest that while a larger k can improve performance by increasing the expressive power, excessive complexity may lead to a decrease in performance due to overfitting.

D Impact of Over-generalization

Detailed Definition of Group G_{over} Group G_{over} consists of test triples (h, r, t) where at least one candidate triple (h, r, t') (where $t' \neq t$) is $\phi_i \in G_u \setminus G_o$ for which there is a corresponding $\psi_i \in G_o$. In this case, the body instances corresponding to the candidate appear in the training set. For instance, consider a local composition pattern $\psi \Rightarrow \phi$ consisting of the relation triplet

Pattern	# of Pattern Instances (n)	WN18RR		FB15k-237		YAGO3-10	
		# of Patterns	Prop. (%)	# of Patterns	Prop. (%)	# of Patterns	Prop. (%)
Symmetry	n = 1	-	-	6	13.6	-	-
	$1 < n \leq 10$	1	20.0	4	9.1	5	41.7
	$10 < n \leq 10^2$	1	20.0	10	22.7	-	-
	$10^2 < n \leq 10^3$	1	20.0	17	38.6	4	33.3
	$n > 10^3$	2	40.0	7	15.9	3	25.0
	Total	**5**	**100%**	**44**	**100%**	**12**	**100%**
Antisymmetry	n = 1	1	9.1	1	1.8	-	-
	$1 < n \leq 10$	2	18.2	13	22.8	3	20.0
	$10 < n \leq 10^2$	2	18.2	26	45.6	5	33.3
	$10^2 < n \leq 10^3$	2	18.2	13	22.8	2	13.3
	$n > 10^3$	4	36.4	4	7.0	5	33.3
	Total	**11**	**100%**	**57**	**100%**	**15**	**100%**
Inversion	n = 1	3	30.0	52	18.6	6	35.3
	$1 < n \leq 10$	2	20.0	86	30.7	4	23.5
	$10 < n \leq 10^2$	5	50.0	83	29.6	4	23.5
	$10^2 < n \leq 10^3$	-	-	51	18.2	1	5.9
	$n > 10^3$	-	-	8	2.9	2	11.8
	Total	**10**	**100%**	**280**	**100%**	**17**	**100%**
Composition	n = 1	17	48.6	1,546	26.3	56	17.6
	$1 < n \leq 10$	14	40.0	2,288	39.0	112	35.2
	$10 < n \leq 10^2$	4	11.4	1,439	24.5	91	28.6
	$10^2 < n \leq 10^3$	-	-	489	8.3	53	16.7
	$n > 10^3$	-	-	111	1.9	6	1.9
	Total	**35**	**100%**	**5,873**	**100%**	**318**	**100%**

Table 7: Distribution of inference patterns: Symmetry/Antisymmetry, Inversion, and Composition according to the number of pattern instances (n) across three benchmark datasets.

(r_1, r_2, r). If the training set contains the body instances (h, r_1, x) and (x, r_2, t') for at least one candidate t' and some entity $x \in E$, then the test triple (h, r, t) is assigned to G_{over}. If KGE models suffer from over-generalization, they are likely to assign a high score to such a candidate (h, r, t'), treating it as a valid triple. For simplicity, we only describe the case of tail prediction, but the same procedure applies to head prediction. For pattern, we only consider symmetry, inversion, and composition patterns, as they are the most representative inference patterns extensively investigated across a wide range of KGE models. Antisymmetry is not considered because it ensures the absence of a head, rather than ensuring the presence of head.

Quantified Impact of Over-generalization Figure 6 presents the performance comparison on the G_{over} across WN18RR and YAGO3-10. Also in WN18RR and YAGO3-10, POGE consistently outperforms the baselines regardless of the local pattern criteria and the dataset. These results demonstrate that POGE effectively addresses the negative impact of over-generalization, and shows robustness across datasets.

Distribution of Patterns by Number of Pattern Instances Table 7 presents the distribution of each pattern according to the number of its pattern instances (n) across three KG benchmark datasets.In WN18RR, FB15k-237, and YAGO3-10, respectively, 20.0%, 22.7%, and 41.7% of symmetry patterns, 27.3%, 24.6%, and 20.0% of antisymmetry patterns, and 50.0%, 49.3%, and 58.8% of inversion patterns have 10 or fewer pattern instances. This result indicates that for symmetry, antisymmetry, and inversion, a substantial proportion of patterns in real-world KGs are observed in only a few instances.

Dataset	Entities	Relations	Triples		
			Train	Valid	Test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Table 8: Statistics of three benchmark datasets

E Datasets

WN18RR, FB15k-237 and YAGO3-10 are used to evaluate POGE. WN18RR and FB15k-237 are subsets of WN18 (Bordes et al., 2013) and FB15k (Bordes et al., 2013) with inverse relations removed, and YAGO3-10 is a subset of YAGO3 (Mahdisoltani et al., 2013) containing only entities with a minimum of 10 relations each. The statistics are summarized in Table 8.