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 dense linear transformations and compound operations for relation representation. Our theoretical analysis demonstrates that a dense 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 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 (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., 2019) 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.

Despite their strong ability to capture various 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 dense 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 and t are the head and tail entities, and r is the relation. KGE maps E and R to low-dimensional vector space and defines a score function to measure triple plausibility.

Distance-based models (DBMs) 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 follows:

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

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) are trained to maximize the score (or semantic similarity) of the factual triple calculated via the multilinear 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 follows:

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

where $h, r, t \in \mathbb{R}^d$, and $\langle \cdot, \cdot, \cdot \rangle$ denotes 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., 2019), 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. We formally define the phenomenon of *pattern over-generalization* as well as *local pattern*, and *universal pattern* as follows:

Pattern Over-Generalization

The model generalizes a *local pattern* to all unseen triples without sufficient evidence, 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

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

While this phenomenon can serve as a crucial inductive bias in KGE, universally generalizing a local pattern without sufficient evidence 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 that is fully trained to satisfy Equation 1 for all triples in G, 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

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

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 KGs are universally valid, especially those with a few pattern instances.[1]

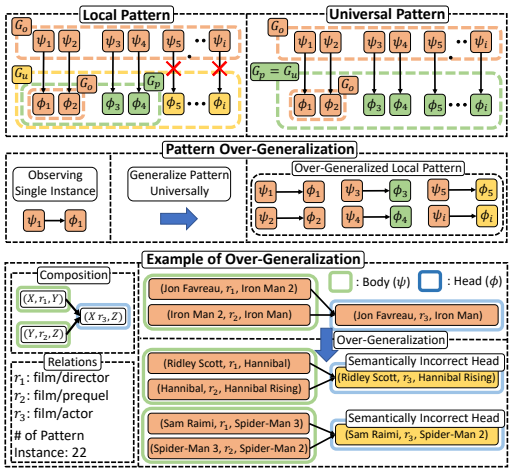
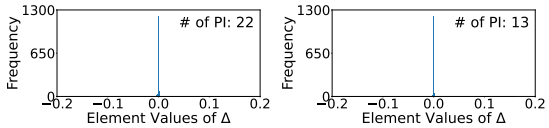


Figure 1: Illustration of local and universal patterns, and the process and examples of over-generalization. Existing models suffer from over-generalization by treating local patterns as universal patterns. The example shows that a local pattern is generalized universally, which leads to erroneous predictions.

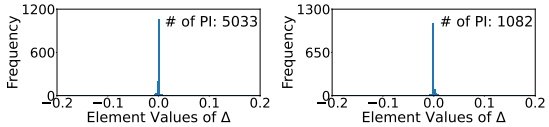
Empirical Evidence Nevertheless, existing KGE models overlook the difference between local and universal patterns, treating all observed patterns as universally valid regardless of their frequency. Figure 2 shows the histograms of the embedding difference $\Delta = r_1^T \circ r_2^T \circ r_3^H - r_1^H \circ r_2^H \circ r_3^T$, 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 regardless of instance frequency. This empirically 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 in Figure 2(a). This indicates

[1] In Appendix I, we present examples of semantically local patterns that have low frequency, along with failure cases where pattern frequency does not match the pattern semantics.



(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.

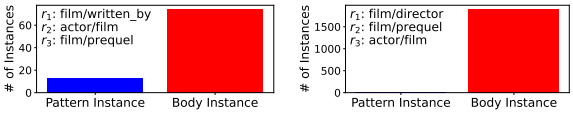


Figure 3: The number of pattern instances and body instances for the local patterns introduced in Figure 2(a).

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. The distribution of pattern instances and body instances of the universal patterns in Figure 2(b) and empirical evidence for another pattern type are presented in Appendix J.

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.

Which Methods Are Affected? Table 1 presents representative examples of KGE methods that suffer from over-generalization. To verify whether these models actually suffer from over-generalization, we propose the Over-Generalization (OG) ratio. Specifically, for the local patterns presented in Figure 3, we extract the head instances corresponding to the body instances and categorize them into True triples (if triples are in G) and False triples (others). The OG ratio is defined as the average score produced by a model for the True triples divided by the average score for the False triples. An OG ratio close to 1 indicates that a model suffers from over-generalization, as it assigns similar scores to both True and False triples. Conversely, an OG ratio closer to 0 implies that the model effectively avoids this issue by assigning higher scores to False triples than to True triples. POGE exhibits a lower OG ratio than other models. This indicates that POGE effectively addresses over-generalization. For more details of the OG ratio, please refer to Appendix H.

Model	Score Function	Over-generalization Sym/Asym/Inves/Comp	OG ratio (↓)
TransE	$\|h + r - t\|$	-/✓/✓/✓	.927
RotatE	$\|h \circ r - t\|$	✓/✓/✓/✓	.921
PairRE	$\|h \circ r^H - t \circ r^T\|$	✓/✓/✓/✓	.917
CompoundE	$\|M_r \cdot h - \hat{M}_r \cdot t\|$	✓/✓/✓/✓	.910
POGE (Ours)	$\|L_r h_r - t_r\|$	✗/✗/✗/✗	**.869**

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.

4 Method

In this section, we present the formulation of POGE and provide a theoretical analysis demonstrating how POGE 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.

Comparison Between Existing Linear Transformation Models Although existing models such as RESCAL (Nickel et al., 2011) and TransR (Lin et al., 2015) employ dense linear transformations, they suffer from overfitting and representing relations as $\mathbb{R}^{n \times n}$ dense linear matrix incurs significant computational costs. As a result, recent KGE models rarely adopt such dense linear transformations. CompoundE (Ge et al., 2023) utilizes sparse affine operators; consequently, it suffers from over-generalization, as presented in Table 1. In contrast, POGE addresses over-generalization

by employing dense linear transformations through a QR decomposition-inspired method, which reduces computational costs. Detailed differences are presented in Appendix L.

QR Decomposition and Partial Sharing for Efficient Parameterization Linear transformation L_r of POGE is decomposed 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. (2019), we adopt self-adversarial negative sampling for training. The loss function can be written as:

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

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 using only the 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[2]:

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

[2]Appendix B details constraint matrix for various patterns.

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 $a = c_1 e_{u_1} + c_2 e_{u_2} + \cdots + c_d e_{u_d}$. By the linearity of the transformation E, it follows that:

$$Ea = c_1 E e_{u_1} + c_2 E e_{u_2} + \cdots + c_d E e_{u_d} = 0 \quad (16)$$

These results indicate that as POGE observes more linearly independent entities e_u, the dimension of the space spanned by these entities increases. Consequently, when the number of observed entities reaches d, POGE guarantees the universal generalization of the pattern across all related instances. In other words, it can be expected that a pattern becomes progressively universal as the number of observed entities increases.

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:

$$A_r = \hat{M}_r^{-1} L_r M_r \quad (17)$$

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. The linear independence of entity embeddings is discussed in Section 6.3. Our theoretical guarantees rely on the ideal assumption that $\|Ee_i\| = 0$. Since satisfying this exact constraint is challenging in practice, we provide further analysis in Appendix K, proving that an approximate constraint ($\|Ee_i\| < \epsilon$) still bounds the pattern constraint of unseen entities.

5 Related Work

5.1 Distance-based Models

Distance-based models capture patterns through various relational operations. TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), Rotate3D (Gao et al., 2020), DualE (Cao et al., 2021), ReflectE (Zhang et al., 2022) and, RotatQ (Xie et al., 2025) model relations through translation, rotation, 3D rotation, a combination of translation and rotation, reflection transformation, and quaternion-based transformation, respectively. Other models enrich these operations: HAKE (Zhang et al., 2020b) uses polar coordinates for semantic hierarchies, PairRE (Chao et al., 2021) and CompoundE (Ge et al., 2023) apply scaling and compound operators, and DensE (Lu et al., 2022) decomposes relations into rotation and scaling in 3D Euclidean space. Recent models further diversify relation modeling: ExpressivE (Pavlović and Sallinger, 2023) and OctagonE (Charpenay and Schockaert, 2024) represent relations as hyper-parallelograms and axis-aligned octagons, respectively. SpeedE (Pavlović and Sallinger, 2024) improves efficiency in low-dimensional Euclidean settings, OrthogonalE (Zhu and Shimodaira, 2024) adopts block-diagonal orthogonal matrices with Riemannian optimization, and Charpenay and Schockaert (2025) theoretically analyze the ability of MuRE to capture inference patterns. Although these models effectively capture patterns, their pattern conditions are determined by relation embeddings, which can generalize patterns even when the pattern is supported by only a few instances.

5.2 Tensor Decomposition Models

Tensor decomposition models capture patterns through interactions among entity and relation embeddings. DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) use bilinear scoring functions, whereas HolE (Nickel et al., 2016) employs circular correlation. ANALOGY (Liu et al., 2017), SimplE (Kazemi and Poole, 2018), and TuckER (Balažević et al., 2019) use normal linear operators, enhanced CP decomposition, and Tucker decomposition, respectively. QuatE (Zhang et al., 2019) extends interactions with quaternion representations, while CustomizE (Guan et al., 2025) introduces customized embeddings to address the long-tail problem. Although these models provide strong representation capacity, they are not explicitly designed to distinguish between local patterns and universal patterns. As a result, they may capture observed patterns, but they do not directly control the scope of pattern generalization based on supporting evidence.

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 these datasets are presented in Appendix E

Knowledge Graph Embedding	WN18RR			FB15k-237			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE (Bordes et al., 2013)	.226	-	.501	.294	-	.465	-	-	-
DistMult (Yang et al., 2015)	.430	.390	.490	.241	.155	.419	-	-	-
ComplEx (Trouillon et al., 2016)	.440	.410	.510	.247	.158	.428	-	-	-
RotatE (Sun et al., 2019)	.476	.428	.571	.338	.241	.533	.495	.402	.670
TuckER (Balažević et al., 2019)	.470	.443	.526	.358	.266	.544	-	-	-
QuatE (Zhang et al., 2019)	.488	.438	.582	.348	.248	.550	-	-	-
Rotate3D (Gao et al., 2020)	.489	.442	.579	.347	.250	.543	-	-	-
HAKE (Zhang et al., 2020b)	.497	.452	.582	.346	.250	.542	.545	.462	.694
DualE (Cao et al., 2021)	.492	.444	.584	.365	.268	.559	-	-	-
PairRE (Chao et al., 2021)	-	-	-	.351	.256	.544	-	-	-
HopfE (Bastos et al., 2021)	.472	.413	.586	.343	.247	.534	.529	.438	.695
DensE (Lu et al., 2022)	.492	-	.586	.351	-	.544	.541	-	.678
ReflectE (Zhang et al., 2022)	.488	.450	.559	.358	.263	.546	-	-	-
ExpressivE (Pavlović and Sallinger, 2023)	.482	.407	**.619**	.350	.256	.535	-	-	-
CompoundE (Ge et al., 2023)	.491	.450	.576	.357	.264	.545	-	-	-
SpeedE (Pavlović and Sallinger, 2024)	.493	.446	-	.320	.227	-	.413	.332	-
OctagonE (Charpenay and Schockaert, 2024)	.479	.436	.561	.332	.241	.517	-	-	-
OrthogonalE (Zhu and Shimodaira, 2024)	.494	.446	.573	.334	.242	.518	-	-	-
CustomizE (Guan et al., 2025)	.486	.446	-	.351	.261	.504	-	-	-
RotatQ (Xie et al., 2025)	.489	.450	.552	.356	.254	**.619**	-	-	-
MuRE variant (Charpenay and Schockaert, 2025)	.469	.427	.553	.307	.212	.503	-	-	-
POGE (Ours)	**.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 (CompoundE)	.491	.357	.477
+ w/o QR Decomposition	OOM	OOM	OOM

Table 3: Ablation study of POGE on WN18RR, FB15k-237 and YAGO3-10. MRR is used for performance comparison. R, Q_r, SN, L_r, and OOM denote the shared upper triangular parameter matrix, relation specific Householder reflection, Spectral Normalization, linear transformation of POGE, and Out of Memory, respectively. In w/o QR Decomposition, $n \times n$ dense linear transformations are used for L_r.

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 (H@1), and Hits@10 (H@10) to compare the performance of different KGE models. MRR denotes the mean reciprocal rank of the correct entities, and H@N represents the proportion of correct entities ranked within the top N. For performance comparison, we evaluate POGE against all KGE models discussed in Section 5.

6.2 Main Results

Link Prediction Performance As shown in Table 2, POGE exhibits superior or competitive performance compared with the baselines. For instance, POGE achieves MRR improvements of 0.009, 0.004, and 0.011 over the second-best models, DualE and HAKE, on WN18RR, FB15k-237 and YAGO3-10, respectively. These results indicate the effectiveness and robustness of POGE across diverse datasets. In addition to the standard benchmarks presented above, Appendix F provides link prediction results on large-scale KG 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. Specifically, w/o L_r (equivalent to CompoundE) exhibits significant performance degradation. w/o L_r does not employ a dense matrix and thus suffers from over-generalization, which suggests that overlooking this problem results in significant performance loss. Furthermore, employing dense linear transformations without QR decomposition was infeasible across all datasets; this demonstrates that models such as TransR (Lin et al., 2015), which rely on dense linear transformations, lack scalability due to their high computational costs.

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

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	Prop. (%)	FB15k-237 # of Patterns	Prop. (%)	YAGO3-10 # of Patterns	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.

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 entities. 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. These results indicate that, since entities in practice are shown to be linearly independent, the dimension of the space spanned by the entities increases as the number of observed entities increases, and that universal generalization is achieved when around $d+1$ entities are observed.[3]

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

[3]We also present empirical results on the independence of entities observed in specific patterns in Appendix G.

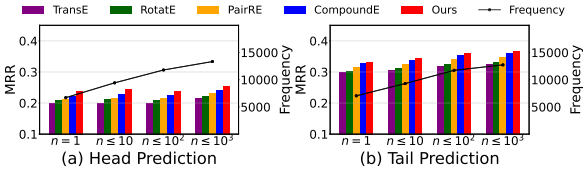


Figure 4: MRR comparison between POGE and baseline models for various sparsity conditions of pattern instances on FB15k-237. The black line indicates the number of test triples of G_{over}.

pattern instances, respectively. This result indicates that a substantial proportion of patterns in KGs are observed in only a few instances. These patterns can be generalized universally when a model suffers from over-generalization. The distributions of other patterns 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 consider symmetry, inversion, composition, hierarchy, intersection, transitive, g.intersection, b. transitive and b. 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 baselines: TransE, RotatE, PairRE, and CompoundE on G_{over}. 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 show that POGE effectively addresses the negative impact of over-generalization.[4]

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 POGE.

[4]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. Therefore, for universal but low-frequency patterns, POGE may fail to generalize them universally, and for local but high-frequency patterns, POGE may generalize them universally, resulting in inappropriate generalization. To address this limitation, in future work, we will leverage the semantics of patterns for pattern generalization.

Furthermore, our model is limited to the transductive setting, where the goal is to learn and improve embedding structures for a fixed set of known entities and relations. While extending POGE to the inductive setting is an important problem for handling unknown entities and relations, addressing it requires substantially different assumptions and architectural designs. For this reason, we leave extending POGE to the inductive setting as future work.

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.

Victor Charpenay and Steven Schockaert. 2025. Less is mure: Revisiting shallow knowledge graph embeddings. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15428–15454.

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.

Zhanpeng Guan, Zhao Zhang, Yiqing Wu, Fuwei Zhang, and Yongjun Xu. 2025. Should we use a fixed embedding size? customized dimension sizes for knowledge graph embedding. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9006–9012.

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

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.

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*.

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. 2019. 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.

Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, Samuel Broscheit, and Christian Meilicke. 2019. On evaluating embedding models for knowledge base completion. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 104–112.

Shiwen Xie, Yongfang Xie, Cheng Hu, and Tingwen Huang. 2025. Rotatq: Knowledge graph embedding based on quaternion unit. *Neurocomputing*, page 132413.

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.

Siheng Zhang, Zhengya Sun, and Wensheng Zhang. 2020a. Improve the translational distance models for knowledge graph embedding. *Journal of Intelligent Information Systems*, 55(3):445–467.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020b. 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.

Dataset	B	N	D	γ	α	lr	k
WN18RR	512	1024	1000	6.0	0.5	0.00005	20
FB15k-237	1024	256	1500	6.0	1.0	0.00005	20
YAGO3-10	1024	400	1000	24.0	1.0	0.0002	2
ogbl-biokg	512	128	2000	12.0	1.0	0.001	12
ogbl-wikikg2	4096	250	100	7.0	1.0	0.005	20

Table 6: The best hyperparameter settings of POGE for link prediction. B, N, D, γ, α, lr, and k denote batch size, negative sampling size, dimension, gamma (presented in Equation 11), alpha (presented in Equation 12), learning rate, and number of Householder reflections, respectively.

A Implementation Details

For the experiments, we adopt the hyperparameter settings from RotatE (Sun et al., 2019) 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 $\{2, 4, 8, 12, 20\}$. More specifically, we utilized the official implementations of RotatE (Sun et al., 2019) and PairRE (Chao et al., 2021) as our codebase. For the datasets, we used WN18RR, FB15k-237, and YAGO3-10 as provided in the official RotatE repository, and the biokg and wikikg2 datasets as provided in the PairRE repository. Table 6 presents the exact batch size, negative sampling size, embedding dimension, γ, learning rate, and k used for each dataset. 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. Experiments for the POGE were conducted on an NVIDIA 3090 with 24GB of memory.

Constraint Matrices for Various Patterns Table 7 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.

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:

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

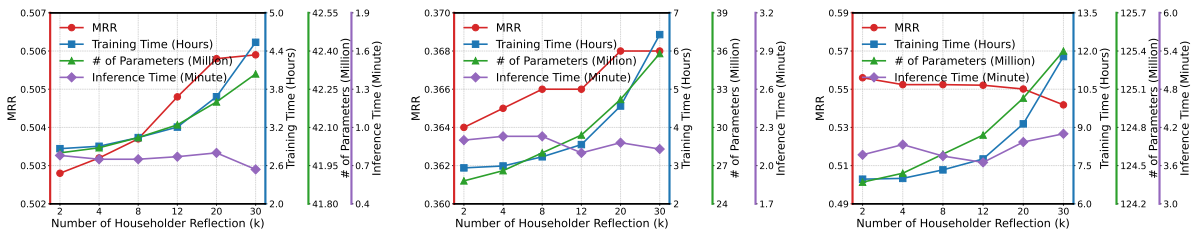
As shown in the inequality, if the distance $\|e_{unobs} - e_{obs}\|$ is small, the error $\|Ee_{unobs}\|$ remains small. This mathematically guarantees that the model generalizes the learned pattern from observed entities to semantically similar entities with similar embeddings.

Pattern	Constraint Matrices ($E = A_\psi - A_\phi$)
Hierarchy $r_1(X,Y) \Rightarrow r_2(X,Y)$	$(A_{r_1} - A_{r_2})X = 0$
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 7: Constraint matrices corresponding to various inference patterns. The patterns are presented in (Krishnan and Rivero, 2024)

C Computational Complexity

Figure 5 presents computational complexity and performance with respect to the Householder reflection k. In WN18RR and FB15k-237, performance



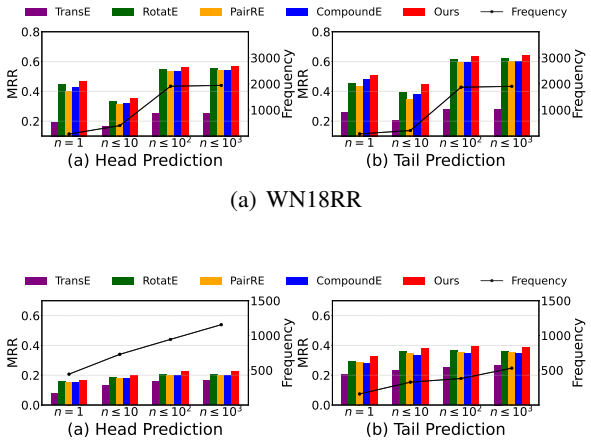
(a) WN18RR (b) FB15k-237 (c) YAGO3-10

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

improves as k increases but shows no significant improvement after $k = 20$. This performance gain is accompanied by an increase in computational cost as k grows. 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. Furthermore, there is almost no variation in inference time across different k values, implying that k can be selected during training without concerns regarding inference time.

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 (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 consider symmetry, inversion, composition, hierarchy, intersection, transitive, g.intersection, b. transitive and b. 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.



(a) WN18RR

(b) YAGO3-10

Figure 6: MRR comparison between POGE and baseline models for various sparsity conditions of pattern instances on two benchmarks: WN18RR (a), and YAGO3-10 (b). The black line indicates the number of test triples for each group.

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 17 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,

Model	ogbl-biokg			ogbl-wikikg2		
	Dim	Valid MRR	Test MRR	Dim	Valid MRR	Test MRR
TransE	2000	0.7456±0.0003	0.7452±0.0004	500	0.4272±0.0030	0.4256±0.0030
DistMult	2000	0.8055±0.0003	0.8043±0.0003	500	0.3506±0.0042	0.3729±0.0045
ComplEx	1000	0.8105±0.0001	0.8095±0.0007	250	0.3759±0.0016	0.4027±0.0027
RotatE	1000	0.7997±0.0002	0.7989±0.0004	250	0.4353±0.0028	0.4332±0.0025
PairRE	2000	0.8172±0.0005	0.8164±0.0005	200	0.5423±0.0020	0.5208±0.0027
ComopoundE	-	-	-	100	0.6716±0.0009	0.6473±0.0022
POGE	2000	**0.8198**±0.0001	**0.8190**±0.0003	100	**0.6841**±0.0010	**0.6606**±0.0006

Table 8: Link prediction results on large-scale KGs, including ogbl-biokg and ogbl-wikikg2. Bold indicates the best result, and underline indicates the second best result. ± indicates the standard deviation.

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
ogbl-biokg	93,773	51	4,763,814	162,870	162,886
ogbl-wikikg2	2,500,604	535	16,109,182	429,456	598,543

Table 9: Statistics of three benchmark datasets

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	996.1±2.0	1000.0	999.4±0.5
1,500	1000.0	1492.8±1.2	1000.0
2,000	1000.0	1500.0	1000.0

Table 10: Mean rank of the subspace spanned by randomly sampled entities in the specific pattern, 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.

antisymmetry, and inversion, a substantial proportion of patterns in real-world KGs are observed in only a few instances.

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. ogbl-biokg (Hu et al., 2020) is a large-scale biomedical KG, and ogbl-wikikg2 (Hu et al., 2020) is a Wikidata knowledge graph that contains a large number of triples. The statistics are summarized in Table 9.

F Link Prediction Performance on Large-scale Knowledge Graph

To verify the effectiveness of POGE in large-scale KGs, we conduct additional experiments on biokg and wikikg2. As shown in the table 9, biokg and wikikg2 contain 4.7 million and 16.1 million triples, respectively, making them significantly larger than the standard benchmarks. The comparative results on biokg and wikikg2 are presented in Table 8. POGE achieves the highest performance in both valid and test MRR compared to the baselines. For instance, POGE achieves MRR improvements of 0.0027 and 0.0133 over the second-best models, PairRE and CompoundE, on biokg and wikikg2 in Test MRR, respectively. Furthermore, POGE exhibits a low standard deviation (±0.001) across all datasets and settings, indicating that it consistently maintains stable performance regardless of initialization. This comparison demonstrates the robustness of our method across different datasets, especially in large-scale KGs.

G Linear Independence of Entities Observed Within a Pattern

In Section 6.3, we empirically observed that the dimension of the space spanned by entities randomly sampled from the entire KG is approximately equal to the number of sampled entities. In this section, we further investigate whether entities observed within specific patterns—rather than across the entire KG—are also linearly independent of each other. Our experimental procedure is as follows: First, we randomly select a pattern containing at least 2,000 instances. We then randomly sample N entities from this pattern (varying N from 100 to 2,000) and measure the rank of the space spanned by these sampled entities. Finally, we repeat this process for 100 independent trials and report the average rank in Table 10. Our results demonstrate that the rank of the space spanned by these entities is approximately equal to the number of sampled entities, indicating that linear independence is indeed preserved even within specific patterns.

Model	Over-Generalization Ratio		
	$\mathbf{n \leq 10}$	$\mathbf{n \leq 10^2}$	$\mathbf{n \leq 10^3}$
TransE	.851	.850	.795
RotatE	.866	.852	.793
PairRE	.749	.719	.651
CompoundE	.744	.711	.638
POGE	**.707**	**.681**	**.618**

Table 11: OG ratio comparison between KGE baselines and POGE on FB15k-237. n indicates the number of pattern instances.

Universal but Low Frequency	Local but High Frequency
_similar_to (74, WN)	isLocatedIn (5742, YAGO)
_also_see (828, WN)	isLocatedIn, hasCapital (1743, YAGO)
dealsWith (160, YAGO)	
hasNeighbor (550, YAGO)	
.../legislative_sessions (668, FB)	
.../military_combatant_group (620, FB)	
.../canoodled/participant (368, FB)	
.../marriage/spouse (342, FB)	
.../friendship/friend (204, FB)	
.../friendship/participant (1216, FB)	
.../dated/participant (1134, FB)	

Table 12: Representative examples of patterns categorized into the two failure case sets: universal but low frequency, and local but high frequency. The numbers in parentheses indicate the frequency of pattern instances, while WN, FB, and YAGO denote the WN18RR, FB15k-237, and YAGO3-10, respectively.

H Detailed Definition of the OG Ratio and Its Comparison on Various Patterns

In this section, we provide a detailed definition of the OG Ratio and additional experiments on the OG Ratio.

Detailed Definition of OG Ratio Since KGE is a relative distance-based method, if the score of a triple (h, r, t) is closer to 0 than that of another triple (h, r, t'), it implies that the model considers (h, r, t) to be more plausible than (h, r, t'). Based on this relative property, we define the OG ratio as follows. First, we extract head instances corresponding to the body instances of patterns in the KG. We then divide them into two groups—True triples and False triples—and measure the average score of each group (we exclude head instances that are in the training data, as they are used to train the models). Next, we define the OG ratio as the average score of True triples divided by that of False triples. An OG ratio ≈ 1 indicates that the model suffers from over-generalization, as it assigns similar scores to both True and False triples. An OG ratio ≈ 0 implies that the model effectively avoids over-generalization by assigning higher scores to False triples compared to True triples.

OG Ratio Comparison on Various Patterns We use the OG ratio to verify whether our model effectively addresses over-generalization in the entire KG. Specifically, we extract True and False triples for patterns whose number of pattern instances n satisfies $n \leq 10$, $n \leq 10^2$, and $n \leq 10^3$ in FB15k-237, and compare the OG ratio of POGE against KGE baselines, including TransE, RotatE, PairRE, and CompoundE. Table 11 presents a comparison of the OG ratios between POGE and the KGE baselines. Our model exhibits a lower OG ratio than all other models across all settings. This demonstrates that our model effectively addresses pattern over-generalization in the entire KG. This analysis was not conducted for WN18RR and YAGO3-10; due to their small validation/test sets and data sparsity, these datasets contain very few True triples (fewer than 10).

I Empirical Analysis of Local and Universal Patterns

In Table 18, we present examples of local and universal patterns in real-world KGs based on human verification. Because symmetry and inversion patterns can be easily classified as either universal or local, we extract these patterns from three benchmark datasets: WN18RR, FB15k-237, and YAGO3-10. For FB15k-237, due to the large number of patterns, we randomly extract only 40 patterns. We then manually classify each pattern into two groups: semantically universal and semantically local. We further categorize these patterns based on the number of pattern instances (n). Empirically, we observe that, in general, patterns that have low frequencies ($n \leq 10$) tend to be semantically local, which can cause over-generalization in existing KGE models, leading to erroneous predictions.

However, we also note that, due to the semantic complexity of real-world KGs, this general tendency may not always hold. We identify two representative failure cases for our proposed method.

- First, **universal but low-frequency patterns** may appear when semantically universal patterns are observed in only a few instances due to dataset sparsity. In this case, POGE may fail to generalize them universally.

- Second, **local but high-frequency patterns** may appear when semantically local patterns have many observed instances. In this case, POGE may generalize them universally, resulting in inappropriate generalization.

Models	WN18RR	FB15k-237	YAGO3-10
TransE	0.242	0.103	0.213
RotatE	0.661	0.121	**0.377**
PairRE	0.657	0.125	0.303
CompoundE	0.660	**0.166**	0.301
POGE	**0.686**	0.150	0.283

Table 13: MRR comparison between POGE and baseline KGE models for the universal but low frequency case set across WN18RR, FB15k-237, and YAGO3-10. Bold indicates the best performance.

Models	MRR
TransE	0.088
RotatE	0.131
PairRE	0.155
CompoundE	**0.185**
POGE	0.172

Table 14: MRR comparison between POGE and baseline KGE models for the local but high frequency case set. Note that the evaluated patterns for this case are observed exclusively within YAGO3-10. Bold indicates the best performance.

Table 12, a subset of Table 18, shows examples of patterns in these two case sets. Each pattern is reported with its frequency and dataset. The universal but low-frequency set indicates semantically universal patterns whose frequency is lower than the entity embedding dimension. The local but high-frequency set indicates semantically local patterns whose frequency is higher than the entity embedding dimension.

We compare POGE with KGE baselines on these cases. Specifically, for each dataset, we extract test triples that contain relations included in each failure case set and measure MRR. Table 13 reports the MRR results for the universal but low frequency set. POGE underperforms compared to CompoundE on FB15k-237 and YAGO3-10. This demonstrates the negative impact of the universal but low frequency failure case on performance of POGE.

Interestingly, POGE still outperforms CompoundE on WN18RR. We analyze the reasons for this as follows. For WN18RR, we measured the MRR for two patterns: *_similar_to* and *_also_see*. First, despite the *_similar_to* pattern having only 74 instances, all models except TransE achieved an MRR of 1.0. This pattern is a potential failure case, but not empirically harmful because of dataset-specific or relatively easy test structure. Furthermore, the *_also_see* pattern has 828 instances. Although these instances may not span the entire embedding space, they still provide observed instances that can support generalization. As theoretically

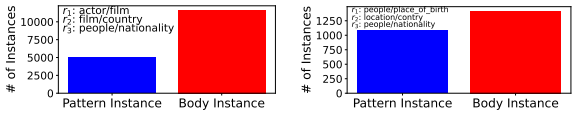


Figure 7: The number of pattern instances and body instances for the universal patterns introduced in Figure 2(b).

shown in Appendix B, our spectral normalization can help keep the pattern error bounded for unobserved entities if they are semantically similar to the observed entities. Therefore, POGE can achieve generalization to semantically similar entities for *_also_see*, which can explain its superior performance on WN18RR.

Table 14 reports the MRR results for the local but high-frequency set (only evaluated on YAGO3-10). In this case, POGE also does not outperform CompoundE. This result indicates that POGE may underperform existing models when semantically local patterns have high frequencies.

Through this analysis, we clarify the applicability boundary of POGE.

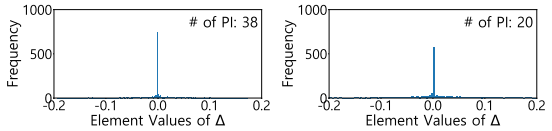
- Our method is designed to prevent the over-generalization of local patterns.
- As shown in Table 18, most local patterns have low frequency, and therefore POGE can generally improve performance on the overall datasets.
- However, POGE may underperform existing models in failure cases where pattern frequency does not match pattern semantics, such as universal but low-frequency patterns or local but high-frequency patterns.
- Therefore, POGE is most suitable for KGs where low-frequency patterns are likely to be local and high-frequency patterns are likely to be universal.

J Additional Empirical Evidence for Over-Generalization

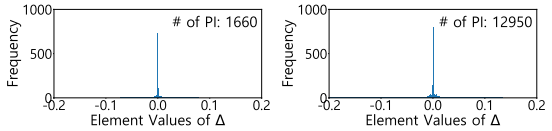
Comparison of the Number of Pattern and Body Instances Between Local and Universal Patterns In Figure 7, we present the number of pattern and body instances for the universal patterns introduced in Figure 2(b). Additionally, Table 15 compares the PI and BI of the local and universal patterns from Figure 2. As shown in Figure 7, the universal

Pattern Type	# of PI	# of BI	Ratio (PI/BI)
Local Pattern *(film/written_by, actor/film, film/prequel)*	13	74	0.176
Local Pattern *(film/director, film/prequel, actor/film)*	22	1,895	0.012
Universal Pattern *(actor/film, film/country, people/nationality)*	5,033	11,659	0.432
Universal Pattern *(people/place_of_birth, location/country, people/nationality)*	1,082	1,422	0.761

Table 15: Comparison of Pattern Instances (PI) and Body Instances (BI) between the local and universal patterns introduced in Figure 2. '# of PI' and '# of BI' represent the number of pattern instances and body instances, respectively.

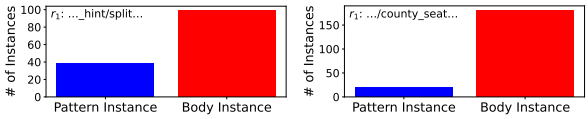


(a) Histograms of local patterns that are supported by scarce pattern instances. The relations for the left and right figures are *(.../gardening_hint/split_to)* and *(.../us_county/county_seat)*, respectively.

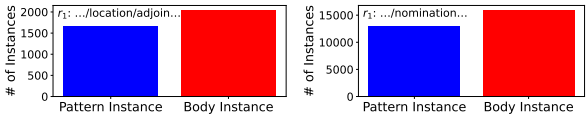


(b) Histograms of universal patterns that are supported by many pattern instances. The relations for the left and right figures are *(.../location/adjoining_relationship...)* and *(.../award/award_nomination...)*, respectively.

Figure 8: Histograms of embedding difference $\Delta = (r_1^H)^2 - (r_1^T)^2$ for different symmetric relations r_1. # of PI denotes the number of pattern instances. r_1 are retrieved from FB15k-237.



(a) The number of pattern instances and body instances for the local patterns introduced in Figure 8(a).



(b) The number of pattern instances and body instances for the universal patterns introduced in Figure 8(b).

Figure 9: The number of pattern instances and body instances for the local and universal symmetry patterns introduced in Figure 8

patterns are supported by 5,033 and 1,082 pattern instances, respectively. This significantly exceeds the number of pattern instances of local patterns (13 and 22), demonstrating that universal patterns are supported by a much larger number of observations. Furthermore, as shown in Table 15, universal patterns exhibit higher PI/BI ratios (0.432 and 0.761) compared to local patterns (0.176 and 0.012). This demonstrates that the patterns in Figure 2(b) exhibit universal characteristics.

Empirical Evidence for the Symmetry Pattern To demonstrate that the over-generalization effect is not limited to the composition pattern discussed in Section 3, we conduct an additional analysis on the symmetry pattern using the same framework in Section 3, as follows.

Similar to the composition pattern, PairRE also induces a relation-level pattern condition for symmetry patterns. If a relation r_1 is symmetric, PairRE satisfies the following condition:

$$(r_1^H)^2 = (r_1^T)^2 \tag{18}$$

This condition is determined only by relation embeddings. Therefore, once this condition is learned from observed triples, the model can generalize the symmetry pattern to other entities, even when the pattern is supported by only a small number of instances. This can lead to pattern over-generalization for local symmetry patterns. We empirically verify this by measuring the distribution of $\Delta = (r_1^H)^2 - (r_1^T)^2$. Figure 8 shows the histograms of the embedding difference (Δ) of local and universal symmetry patterns. Figures 8(a) and 8(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 symmetry patterns regardless of instance frequency. The numbers of pattern instances and body instances for these symmetry patterns are shown in Figure 9. This analysis empirically shows that pattern over-generalization also occurs in another pattern type.

Model	Score Function	Pattern Modeling	Over-Generalization	Scalability	MRR	
					WN18RR	FB15k-237
RERSCAL	$< h^T W_r t >$	✗	-	✗	.420	.270
TransR	$-\|P_r h + r - P_r t\|$	✗	-	✗	.220	.299
CompoundE	$-\|M_r \cdot h - \hat{M}_r \cdot t\|$	✓	✓	✓	.491	.357
POGE	$-\|L_r h_r - t_r\|$	✓	✗	✓	**.506**	**.369**

Table 16: Summary of differences between POGE and the existing linear transformation models. RESCAL is reproduced in Wang et al. (2019), and TransR is reproduced in Zhang et al. (2020a). W_r and P_r are $\mathbb{R}^{d \times d}$ dense linear transformations, where d is the dimension of entities and relations.

K Bounding Constraint Errors Under Practical Training Dynamics

In Section 4.2, we provide a theoretical analysis showing that a dense linear transformation can address over-generalization. However, in practice, KGE embeddings are optimized with negative sampling and a margin-based loss; therefore, the constraint $Ee_i = 0$ is approximate. That is, satisfying the exact linear constraint $Ee_i = 0$ is challenging due to the approximate nature of margin-based optimization with negative sampling. However, we can mathematically guarantee that the constraint error for unseen entities is bounded by the constraint error of observed entities. Specifically, let us assume the model is sufficiently trained such that the constraint error is minimized within a small margin ϵ for the observed linearly independent entities $e_1, \ldots, e_d$ (i.e., we consider the practical scenario in which $\|Ee_i\| < \epsilon$, rather than the exact condition $\|Ee_i\| = 0$.) Since we empirically verified that the learned entity embeddings form a basis (Section 6.3), any unseen entity e_{new} can be represented as a linear combination of the observed entities: $e_{new} = \sum_{i=1}^{d} c_i e_i$. By the linearity of the transformation E and the triangle inequality, the error for the unseen entity is bounded as follows:

$$\begin{aligned} \|Ee_{new}\| &= \left\| \sum_{i=1}^{d} c_i (Ee_i) \right\| \\ &\leq \sum_{i=1}^{d} |c_i| \cdot \|Ee_i\| \qquad (19) \\ &< \left(\sum_{i=1}^{d} |c_i| \right) \epsilon \end{aligned}$$

This inequality demonstrates that minimizing the constraint error of observed entities ($\epsilon \to 0$) directly suppresses the constraint error for unseen entities. Therefore, even under the approximate optimization of margin-based loss, the constraint error of unseen entities remains bounded. Through this analysis, we clarify the practical scope of the theoretical guarantees of our method. In practice, exact constraint satisfaction, corresponding to the ideal zero-error case, i.e., $\|Ee_i\| = 0$, is not guaranteed after training. Rather, when the constraint errors of observed entities are small, corresponding to the approximate case, i.e., $\|Ee_i\| < \epsilon$, the errors of unseen entities can also be bounded, thereby ensuring the practical effectiveness of our proposed method.

L Differences between the existing linear transformation model and POGE

We present a comparison between existing linear transformation models and POGE in Table 16. Existing models that use dense linear transformation such as RESCAL (Nickel et al., 2011) and TransR (Lin et al., 2015) are not designed for pattern modeling; moreover, when the entity and relation dimensions are n, they assign an $\mathbb{R}^{n \times n}$ matrix to each relation, leading to high computational costs as the dimension of entities and relations increases. While CompoundE is capable of pattern modeling and avoids these computational issues by using sparse affine transformation, it suffers from over-generalization. In contrast, POGE is designed for pattern modeling, addresses the over-generalization problem, and avoids the computational cost issue using a QR-decomposition-inspired method. In addition, POGE outperforms existing models on WN18RR and FB15k-237, demonstrating its effectiveness.

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%
Hierarchy	n = 1	1	12.5	72	21.4	11	19.6
	$1 < n \leq 10$	2	25.0	94	28.0	16	28.6
	$10 < n \leq 10^2$	5	62.5	114	33.9	14	25.0
	$10^2 < n \leq 10^3$	-	-	50	14.9	12	21.4
	$n > 10^3$	-	-	6	1.8	3	5.4
	Total	**8**	100%	**336**	100%	**56**	100%
Intersection	n = 1	-	-	438	29.6	18	15.0
	$1 < n \leq 10$	-	-	642	43.3	54	45.0
	$10 < n \leq 10^2$	-	-	378	25.5	42	35.0
	$10^2 < n \leq 10^3$	-	-	24	1.6	6	5.0
	$n > 10^3$	-	-	-	-	-	-
	Total	**-**	**-**	**1,482**	100%	**120**	100%
Transitive	n = 1	-	-	2	4.1	-	-
	$1 < n \leq 10$	1	16.7	8	16.3	5	33.3
	$10 < n \leq 10^2$	2	33.3	16	32.7	4	26.7
	$10^2 < n \leq 10^3$	3	50.0	11	22.4	2	13.3
	$n > 10^3$	-	-	12	24.5	4	26.7
	Total	**6**	100%	**49**	100%	**15**	100%
G.Intersection	n = 1	2	33.3	21	13.0	2	18.2
	$1 < n \leq 10$	-	-	54	33.5	5	45.5
	$10 < n \leq 10^2$	4	66.7	70	43.5	2	18.2
	$10^2 < n \leq 10^3$	-	-	14	8.7	2	18.2
	$n > 10^3$	-	-	2	1.2	-	-
	Total	**6**	100%	**161**	100%	**11**	100%
B. Transitive	n = 1	-	-	6	15.8	-	-
	$1 < n \leq 10$	-	-	2	5.3	1	12.5
	$10 < n \leq 10^2$	1	33.3	11	28.9	2	25.0
	$10^2 < n \leq 10^3$	2	66.6	9	23.7	3	37.5
	$n > 10^3$	-	-	10	26.3	2	25.0
	Total	**3**	100%	**38**	100%	**8**	100%
B. Composition	n = 1	12	40.0	1,278	28.4	3	7.1
	$1 < n \leq 10$	18	60.0	1,710	38.1	15	35.7
	$10 < n \leq 10^2$	-	-	1,053	23.4	12	28.6
	$10^2 < n \leq 10^3$	-	-	345	7.7	12	28.6
	$n > 10^3$	-	-	108	2.4	-	-
	Total	**30**	100%	**4,494**	100%	**42**	100%

Table 17: Distribution of inference patterns: Symmetry/Antisymmetry, Inversion, Composition, Hierarchy, Intersection, Transitive, G.Intersection, B. Transitive and B. Composition according to the number of pattern instances (n) across three benchmark datasets.

# of Pattern Instances	Semantically Universal (n)	Semantically Local (n)
$n \leq 10$	-	*owns* (4) (S) *isAffiliatedTo* (10) (S) *isKnownFor* (4) (S) *created* (2) (S) *hasAcademicAdvisor* (4) (S) *influences*, *created* (3) (I) *influences*, *isInterestedIn* (8) (I) *participatedIn*, *isCitizenOf* (1) (I) *created*, *hasAcademicAdvisor* (1) (I) *isAffiliatedTo*, *isMarriedTo* (1) (I) *isLocatedIn*, *isConnectedTo* (1) (I) *influences*, *isKnownFor* (5) (I) *isLocatedIn*, *dealsWith* (1) (I) *influences*, *hasChild* (1) (I) *isMarriedTo*, *influences* (8) (I) *_synset_domain_topic_of* (2) (S) *_synset_domain_topic_of*, *_has_part* (9) (I) *_hypernym*, *_derivationally_related_form* (17) (I) *_derivationally_related_form*, *_member_meronym* (23) (I) *_hypernym*, *_also_see* (38) (I) *_hypernym*, *_synset_domain_topic_of* (4) (I) *_also_see*, *_verb_group* (1) (I) *_hypernym*, *_verb_group* (17) (I) *_has_part*, *_member_of_domain_region* (1) (I) *_instance_hypernym*, *_member_of_domain_region* (1) (I) *_derivationally_related_form*, *_synset_domain_topic_of* (23) (I) */location/administrative_division/country* (4) (S) */location/country/capital* (3) (S) */film/film/prequel* (2) (S) */people/person/profession* (1) (S) */film/film/genre* (1) (S) */film/film_subject/films* (1) (S) */base/aareas/schema/administrative_area/capital* (1) (S) */medicine/symptom/symptom_of* (1) (S) */music/instrument/family* (1) (S) *.../sibling_relationship/sibling*, *.../influenced_by* (2) (I) *.../influenced_by ../romantic_relationship/celebrity* (1) (I)
$10 < n \leq 100$	*_similar_to* (74) (S)	*isLocatedIn*, *participatedIn* (50) (I) *isMarriedTo*, *hasChild* (59) (I) *influences*, *hasAcademicAdvisor* (63) (I) *isLocatedIn*, *owns* (60) (I)
$100 < n \leq 1000$	*dealsWith* (160) (S) *hasNeighbor* (550) (S) *_also_see* (828) (S) *.../legislative_sessions* (668) (S) *.../military_combatant_group/combatants* (620) (S) *.../canoodled/participant* (368) (S) *.../marriage/spouse* (342) (S) *.../friendship/friend* (204) (S)	*influences* (180) (S) *hasChild* (414) (S) *dealsWith*, *hasNeighbor* (165) (I) *.../award_nomination/nominated_for* (592) (S) *.../recording_contribution/performance_role* (402) (S) *.../award_honor/honored_for* (392) (S) *.../education/major_field_of_study* (282) (S) *.../sports_team_roster/position* (106) (S) *.../location/contains*, *.../first_level_division_of* (124) (I) *.../genre/titles*, *...film/country* (157) (I) *.../performance/film*, *.../award_honor/award_winner* (963) (I) *.../sports_team_roster/team*, *.../american_football/sports_team_roster/position* (470) (I) *...location/contains*, *.../mailing_address/state_province_region* (359) (I) *.../regular_tv_appearance/actor*, *.../award_nomination/nominated_for* (445) (I)

		...location/contains, *.../mailing_address/country* (103) (I) *.../award_nomination/nominated_for*, *.../produced_by* (497) (I) *.../dated/participant*, *../romantic_relationship/celebrity* (136) (I) *.../award_nomination/nominated_for*, */film/film/music* (314) (I) *.../music/group_membership/role*, *.../performance_role* (246) (I) *.../dated/participant*, *.../marriage/spouse* (108) (I)
$n > 1000$	*isMarriedTo* (3674) (S) *happenedIn*, *participatedIn* (1468) (I) *_derivationally_related_form* (27701) (S) *_verb_group* (1060) (S) *.../award_nomination/award_nominee* (12950) (S) *.../award_honor/award_winner* (6860) (S) *.../track_contribution/role* (3068) (S) *.../group_membership/role* (2170) (S) *.../adjoining_relationship/adjoins* (1660) (S) *.../friendship/participant* (1216) (S) *.../dated/participant* (1134) (S)	*isLocatedIn* (5742) (S) *isLocatedIn*, *hasCapital* (1743) (I)

Table 18: Human verification results of semantically universal and semantically local patterns, categorized by the number of pattern instances. The number next to each pattern indicates the number of pattern instances, and (S) and (I) indicate symmetry and inversion, respectively.