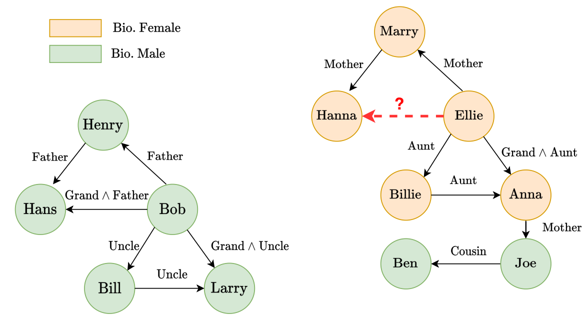
Double Permutation Equivariance for Knowledge Graph Completion

Anonymous Authors[1]

Abstract

This work provides a formalization of Knowledge Graphs (KGs) as a new class of graphs that we denote *doubly exchangeable attributed graphs*, where node and pairwise (joint 2-node) representations must be equivariant to permutations of both node ids and edge (& node) attributes (relations & node features). Double-permutation equivariant KG representations open a new research direction in KGs. We show that this equivariance imposes a structural representation of relations that allows neural networks to perform complex logical reasoning tasks in KGs. Finally, we introduce a general blueprint for such equivariant representations and test a simple GNN-based double-permutation equivariant neural architecture that achieve state-of-the-art Hits@10 test accuracy in the WN18RR, FB237 and NELL995 inductive KG completion tasks, and can accurately perform logical reasoning tasks that no existing methods can perform, to the best of our knowledge.



(a) Training KG

(b) Test KG (new nodes & relations)

Figure 1. (Biological human KG) Illustrative knowledge graphs of biological human relations. Exemplar inductive task: learn on training KG (a) to inductively predict missing relation "?" over Test KG (b) with new nodes (potentially more), new relations (potentially more), and new node features (potentially more).

1. Introduction

Knowledge graphs (KGs) are generally defined as structured representations of collections of facts in the form of a set of triplets $\mathcal{S} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, where $(i, r, j) \in \mathcal{S}$ define two entities i (head entity) and j (tail entity) connected by a relation r, where both nodes and relations are finite: $N = |\mathcal{V}| < \infty$ and $R = |\mathcal{R}| < \infty$. In some applications KGs naturally define conjunctive logical statements (as in Figure 1(a)): $(i, \text{Father}, j) \wedge (j, \text{Father}, u) \wedge (i, \text{Grand}, u) \wedge (i, \text{Father}, u) \wedge$ etc., where $\mathcal{R} = \{\text{Father}, \text{Grand}, \ldots\}$ and $\mathcal{V} = \{i, j, u, \ldots\}$.

Unfortunately, KGs are often incomplete. Hence, the task of predicting missing relations (e.g., predict missing $(i, r, j) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$) is both widely-studied and a key task in knowledge base construction, often denoted as *knowledge graph completion* (Bordes et al., 2013; Nickel et al., 2015; Teru et al., 2020). As defined above (and in the literature (Lin et al., 2015; Chen et al., 2020b; Kejriwal et al., 2021; Shen et al., 2022)), knowledge graph completion is the task of predicting attributed edges, where not only we need to identify that a pair of nodes $(i, j) \in \mathcal{V} \times \mathcal{V}$ is a missing edge in the KG, but also determine which relation $r \in \mathcal{R}$ the edge (i, j) has.

Treating KGs as attributed graphs allows researchers to adapt Graph Neural Network (GNN) methods used for link prediction with only minor modifications: Distinct pooling operations for each edge type (regularized to avoid overfitting), and changing the output from binary classification (edge prediction) to multi-label classification (predicting R relation labels). Roughly, this is the formula followed by RGCN (Schlichtkrull et al., 2018), GraIL (Teru et al., 2020), NodePiece (Galkin et al., 2021), NBFNet (Zhu et al., 2021), and ReFactorGNNs (Chen et al., 2022), among others. *However, theoretically, are KGs just attributed graphs?*

Contributions. *In this work we argue that some KGs belong to a new class of graphs (which we denote as* doubly exchangeable attributed graphs*) whose node and pairwise representations must be equivariant to the action of the permutation group composed by the permutation subgroups of node ids, edge attributes (relations), and node*

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

attributes (theoretically modeled as self-edge attributes without loss of generality). This equivariance imposes a type of structural learning akin to inductively learning to answer a subset of Horn clauses in the test KG where both entities and relations are subject to universal quantifiers (Definition 4.9), e.g., for a given pair $i, j \in \mathcal{V}^{\text{test}}$, and $\forall r_1, r_2 \in \mathcal{R}^{\text{test}}, \forall u \in \mathcal{V}^{\text{test}}, (i, r_1, u) \wedge (u, r_2, j) \implies (i, r_2, j)$. If $\mathcal{S}_1^{\text{train}}, \mathcal{S}_2^{\text{train}}, \ldots \subseteq \mathcal{V}^{\text{train}} \times \mathcal{R}^{\text{train}} \times \mathcal{V}^{\text{train}}$ are the training KGs, this equivariance allows the trained predictor to perform predictions over $\mathcal{S}^{\text{test}} \subseteq \mathcal{V}^{\text{test}} \times \mathcal{R}^{\text{test}} \times \mathcal{V}^{\text{test}}$ in the test KG, where $\mathcal{V}^{\text{test}}$ and $\mathcal{V}^{\text{train}}$, $\mathcal{R}^{\text{test}}$ and $\mathcal{R}^{\text{train}}$ are all potentially distinct sets (with potentially distinct sizes).

We believe our double-permutation equivariance could be similarly impactful for KG machine learning as (single) permutation-equivariance has been for graph machine learning. Our work will focus on inductive KG completion tasks, but the same methods can also be applied transductively. There is no existing KG completion task in the literature that can test the full power of our approach. Our experimental results use the inductive WN18RR, FB237 and NELL995 KG completion tasks, which have new nodes in test but not more and completely new relations. We also design two harder synthetic tasks. No existing KG completion method is able to perform one of our synthetic tasks (see Table 1 and Table 2), where the test KG has more and new relations than the training KG.

2. Theory review: What are attributed graphs? An exchangeability perspective

Theoretically, a multigraph —we will refer to multigraphs as graphs— with $N \geq 2$ vertices is a sequence of edges $\mathbf{A}^{(N,R)} = (A_{1,1}, A_{1,2}, \ldots, A_{N,N}) \in \mathbb{A}_R^{N^2}$, where $\mathbb{A}_R$ is some arbitrary domain that encodes $R \geq 1$ relation attributes —e.g., in KGs $\mathbb{A}_R$ is a finite set representing multiple edges and their attributes (i.e., multiple relations). Without loss of generality, node attributes will be defined as a special type of edge reserved for self-loops $A_{i,i}$. In most applications, what distinguishes a graph from a sequence is the assumption that the choice of node ids to create this sequence is arbitrary. Hence, any prediction that uses $(A_{1,1}, \ldots, A_{N,N})$ as input should be invariant[1] to the permutation of node ids (Srinivasan & Ribeiro, 2020). In statistics, this property is known as joint (array) exchangeability (Aldous, 1981). GNNs (without positional encoding) are permutation-equivariant representation functions, possessing the correct invariances for node and graph-wide classification tasks (Xu et al., 2019a; Morris et al., 2019; Srinivasan & Ribeiro, 2020). Link prediction is better served by equivariant pairwise representations (Srinivasan & Ribeiro, 2020).

[1] Refer to the theory in (Srinivasan & Ribeiro, 2020) for the sufficiency of invariances.

More precisely, and without loss of generality, let $\mathcal{V}^{(N)} = \{1, \ldots, N\}$ be the set of nodes (i.e., node ids). For consistency of the notation with knowledge graph, we denote $(\mathbf{A}^{(N,R)})_{i,r,j} = A_{i,j,r}$. Let $\pi \in \mathbb{S}_N$ be a permutation from the symmetric group $\mathbb{S}_N$ with degree N, and $(\pi \circ \mathbf{A}^{(N,R)})_{\pi \circ i, r, \pi \circ j} = (\mathbf{A}^{(N,R)})_{i,r,j}$ be the action of permutation π on the sequence $(A_{1,1}, \ldots, A_{N,N})$, which permutes node ids according to π, that is, $\pi \circ i = \pi_i$, $\forall (i, r, j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$. A function that outputs a d-dimensional node representations of any-size graphs is defined as $\Gamma_{\text{node}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} (\mathcal{V}^{(N)} \times \mathbb{A}_R^{N^2}) \to \mathbb{R}^d$, $d \geq 1$, should be invariant to node id permutations. That is, $\Gamma_{\text{node}}(i, \mathbf{A}^{(N,R)}) = \Gamma_{\text{node}}(\pi \circ i, \pi \circ \mathbf{A}^{(N,R)}), \forall i \in \mathcal{V}$.

Similarly, a neural network that outputs d-dimensional pairwise representations of any-size graphs is defined as $\Gamma_{\text{pair}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} (\mathcal{V}^{(N)} \times \mathcal{V}^{(N)} \times \mathbb{A}_R^{N^2}) \to \mathbb{R}^d$. Pairwise representation should also be invariant to the action of any $\pi \in \mathbb{S}_N$, i.e., $\Gamma_{\text{pair}}((i, j), \mathbf{A}^{(N,R)}) = \Gamma_{\text{pair}}((\pi \circ i, \pi \circ j), \pi \circ \mathbf{A}^{(N,R)})$.

We can also define a graph-wide representation $\Gamma_{\text{gra}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2} \to \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}$, noting that only the mappings between the domain and image that have the same values of R and N are possible. The representation Γ_{gra} is equivariant, that is, for $\pi \in \mathbb{S}_N$, $\pi \circ \Gamma_{\text{gra}}(\mathbf{A}^{(N,R)}) = \Gamma_{\text{gra}}(\pi \circ \mathbf{A}^{(N,R)})$.

GNNs and nearly all recent advances in graph representation learning are driven by the above invariances and equivariances (Bronstein et al., 2017; Chen et al., 2020a; Defferrard et al., 2016; Gilmer et al., 2017; Gori et al., 2005; Hamilton et al., 2017; Maron et al., 2019; Morris et al., 2019; Murphy et al., 2019b;a; Srinivasan & Ribeiro, 2020; Teru et al., 2020; Xu et al., 2019a; Zhu et al., 2021).

3. Brief Related Work: Knowledge graphs as attributed multigraphs

To the best of our knowledge the term *knowledge graph* was first introduced by Schneider (1973) to describe a tutoring system, where each node describes a concept and each arc (direct edge) describes an attributed association between concepts. By 2012, KGs received renewed interest when Google revealed them as a key ingredient in its successful search engine, "things not strings" as described in Singhal (2012). In light of recent advances in large language models (Schulman et al., 2022), the discussion whether knowledge can be described by things or strings gain renewed interest. And we believe our work sheds new light into this discussion, since we show that complex logical relations are the product of forcing an invariance (and not directly learned from associations in the data).

The view of KGs as attributed (multi)graphs —sometimes denoted *heterogeneous graphs*— was somewhat consolidated in the *semantic web* literature around 2016 (Kroetsch

& Weikum, 2016; Paulheim, 2017) and by early work on knowledge bases (Bordes et al., 2011), that later was able to integrate classical AI methods (based on knowledge bases and logic), statistical relational learning (SRL) (De Raedt, 2008; Koller et al., 2007; Kersting & De Raedt, 2007; Heckerman et al., 2007; Neville & Jensen, 2007), and attributed graph completion methods KGs (Bordes et al., 2013; Nickel et al., 2015; Teru et al., 2020).

In the SRL literature (e.g., Raedt et al. (2016)) the attribute of an edge $(i, j) \in \mathcal{V} \times \mathcal{V}$ is sometimes instantiated as either a node $r \in \mathcal{R}$ or a node $r(i, j)$, where r is the edge attribute (relation) (e.g., Heckerman et al. (2007)). The drawback of adding edge attributes (relations in $\mathcal{R}$) as nodes in a Bayesian network is that Bayesian networks are sequences (non-exchangeable), but, if treated as a graph (exchangeable), nodes and relations would be exchangeable among themselves (which in many KG applications would be incorrect, since $\mathcal{V}$ and $\mathcal{R}$ are fundamentally distinct sets). Exchangeability w.r.t. node ids in SRL appears in the form of lifting for parameterized templated graphical models, see Koller & Friedman (2009) and Raedt et al. (2016, Chapter 3.1). In practice, automatically finding these templates is difficult and tends to underperform when compared to more modern attributed graph methods for KG completion.

State-of-the-art methods for KG completion treat KGs as attributed (multi)graphs (i.e., only node id exchangeable). They include tensor factorization methods (Bordes et al., 2013; Trouillon et al., 2016; 2017; Sun et al., 2019) (mostly applied in transductive KG tasks) and graph neural network methods (GNNs) (Chen et al., 2022; Schlichtkrull et al., 2018; Galkin et al., 2021; Teru et al., 2020; Wang et al., 2021; Zhu et al., 2021) (mostly applied in inductive KG tasks), among others. Interestingly, out of those, the most successful embedding methods (tensor or GNNs) tend to impose some form of TransE-style (Bordes et al., 2013) translation equivariance in the embeddings (or impose rotation invariance). This embedding equivariance is *markedly different* from relation equivariance (which we will define later), since here each relation has its own personalized shift. Due to space constraints, a more detailed discussion of related work can be found in Appendix B.

4. Proposal: Define some KGs as double exchangeable attributed (multi)graphs

In the following text, we provide definitions and theoretical statements of our proposal in the main paper, while referring all proofs to Appendix A. Our model is intended for a broad class of KGs (but not all KGs may satisfy our conditions). The proposal starts with defining the concept of knowledge graph used in this paper:

Definition 4.1 (Knowledge Graph (KG))**.** A knowledge graph is a multigraph $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ sampled as $\mathbf{A} \sim \mu$, where μ is some unknown distribution, and $\mathbb{A}_R$ is a set encoding edge attributes (relations) [2]. For instance, if $\mathbf{A}$ has no node attributes, we can define $\mathbb{A}_R \in \{0, 1\}^R$, where $\mathbf{A}_{i,r,j} = 1$ iff the relation (i, r, j) exists in the knowledge graph $\mathbf{A}$. For homogeneous graph without node features, $\mathbb{A}_1 = \{0, 1\}$. W.l.o.g. we define $\mathcal{V}^{(N)} = \{1, \ldots, N\}$ and $\mathcal{R}^{(R)} = \{1, \ldots, R\}$. If the KG has node attributes, $\mathbb{A}_R$ also encodes them, to be used by the set of self-loops $\{\mathbf{A}_{i,r,i} : i \in \mathcal{V}^{(N)}, r \in \mathcal{R}^{(R,\text{self})}\}$ for a special subset of relations $\mathcal{R}^{(R,\text{self})} \subsetneq \mathcal{R}^{(R)}$. Often $\mathcal{R}^{(R)}$ is described through a bijection to a set of sentences (e.g., $1 \to$ Father, $2 \to$ Grand, $\ldots$). *What distinguishes our KG definition from an attributed multigraph (Section 2) is the assumption that the KG distribution μ is such that $\mu(\mathbf{A}_G) = \mu(\mathbf{A}_H)$ for any isomorphic KGs $\mathbf{A}_G$ and $\mathbf{A}_H$ ($\mathbf{A}_G \simeq_{\text{KG}} \mathbf{A}_H$ as in Definition 4.2).* In this paper we denote this property of μ as *double exchangeability*.

We then define the concept of KG isomorphism as:

Definition 4.2 (KG Isomorphism)**.** We say two multigraphs $\mathbf{A}_G, \mathbf{A}_H \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ are isomorphic (denoted as $\mathbf{A}_G \simeq_{\text{KG}} \mathbf{A}_H$) if there exists a node bijection $\phi : \mathcal{V}_G \to \mathcal{V}_H$ and a relation bijection $\tau : \mathcal{R}_G \to \mathcal{R}_H$ preserving the set of relations, i.e., $\forall (i, r, j) \in \mathcal{V}_G \times \mathcal{R}_G \times \mathcal{V}_G$, $(\mathbf{A}_G)_{i,r,j} = (\mathbf{A}_H)_{\phi(i),\tau(r),\phi(j)}$.

Remark (vertex and relation set sizes): Note that by Definition 4.1, the set of all knowledge graphs is $\cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$. Common GNN representations can be learned and applied to graphs of different sizes. Similarly, our representations can be learned and applied to KGs with any number of nodes ($N \geq 2$) **and** any number of relations ($R \geq 1$).

Invariant KG representations. It follows from Definition 4.1 that any (statistical) loss function (e.g., likelihood, regression (via energy-based models using distances), cross-entropy) defined over a knowledge graph $\mathbf{A}_G^{\text{train}}$ must be the same over any isomorphic KGs $\mathbf{A}_H \simeq_{\text{KG}} \mathbf{A}_G^{\text{train}}$, i.e., the loss over $\mathbf{A}_G^{\text{train}}$ must be invariant to permutations of the node ids, edge attributes (relations), and node attributes (types). Consequently, we will design representations that are invariant to these two permutations, as we see later.

Definition 4.3 (Permutation actions on KGs)**.** For any KG $\mathbf{A}^{(N,R)} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$. As before, let $\phi \in \mathbb{S}_N$ be an element of the symmetric group $\mathbb{S}_N$ (a permutation). The operation $\phi \circ \mathbf{A}^{(N,R)}$ is the action of ϕ on $\mathbf{A}^{(N,R)}$, defined as $(\phi \circ \mathbf{A}^{(N,R)})_{\phi \circ i, r, \phi \circ j} = (\mathbf{A}^{(N,R)})_{i,r,j}$, $\forall (i, r, j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$. In our definition of KGs, we also need relation permutations, where $\tau \in \mathbb{S}_R$ is a relation permutation and the action of τ on $\mathbf{A}^{(N,R)}$ is defined as $\forall (i, r, j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, $(\tau \circ \mathbf{A}^{(N,R)})_{i,\tau \circ r,j} = (\mathbf{A}^{(N,R)})_{i,r,j}$, where

[2] We use $\mathbf{A}$ to denote arbitrary KG of any size instead of $\mathbf{A}^{(N,R)}$, where N and R can be automatically inferred from $\mathbf{A}$.

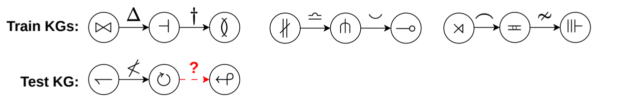


Figure 2. (Alien KG) Illustrative inductive knowledge graph completion task of our alien KG. The task is to inductively predict the missing relation "?" in red. Note that relations are all unique.

we define the action of a permutation $\tau \in \mathbb{S}_R$, as $\tau \circ r = \tau_r$.

The node and relation permutation actions on $\mathbf{A}$ are commutative, i.e., $\phi \circ \tau \circ \mathbf{A} = \tau \circ \phi \circ \mathbf{A}$. We now define isomorphic triplets based on the notion of KG isomorphism.

Definition 4.4 (Isomorphic triplets in KGs)**.** $\forall \mathbf{A}_G, \mathbf{A}_H \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$, we say two triplets $(i, r, j) \in \mathcal{V}_G \times \mathcal{R}_G \times \mathcal{V}_G$, $(i', r', j') \in \mathcal{V}_H \times \mathcal{R}_H \times \mathcal{V}_H$ are isomorphic triplets iff $\mathbf{A}_G$ and $\mathbf{A}_H$ have the same graph sizes and relation sizes, and $\exists \phi \in \mathbb{S}_N, \exists \tau \in \mathbb{S}_R$, such that $\phi \circ \tau \circ \mathbf{A}_G = \mathbf{A}_H$ and $(i', r', j') = (\phi \circ i, \tau \circ r, \phi \circ j)$.

Definition 4.4 implies isomorphic triplets can only exist between (a) two (distinct) isomorphic KGs $\mathbf{A}_G$ and $\mathbf{A}_H$, or (b) in the same graph $\mathbf{A}_G = \mathbf{A}_H$ if $\exists \phi \in \mathbb{S}_N, \exists \tau \in \mathbb{S}_R$ that are non-trivial automorphism, i.e., $\phi \circ \tau \circ \mathbf{A}_G = \mathbf{A}_G$ and ϕ and τ are not identity maps. Now we can finally define our invariant triplet representation for KGs, which is invariant over isomorphic triplets.

Definition 4.5 (Invariant triplet representation for KGs)**.** For any KG $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$, a double invariant representation of a triplet (i, r, j) is a function $\Gamma_{\text{tri}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} (\mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)} \times \mathbb{A}_R^{N^2}) \to \mathbb{R}^d, d \geq 1$, where $\Gamma_{\text{tri}}((i, r, j), \mathbf{A}) = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ r, \phi \circ j), \phi \circ \tau \circ \mathbf{A})$, $\forall (i, r, j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}, \forall \phi \in \mathbb{S}_N, \forall \tau \in \mathbb{S}_R$.

Remark (normal subgroup for preserving relations (and node attributes) that do not permute): One can also trivially extend our definitions to restrict exchangeability to a subset of relations. This is achieved by redefining the permutation group $\mathbb{S}_R$ as its normal subgroup encompassing just the relations that permute, which then implies a trivial change to the definition of KG isomorphism in Definition 4.2. This is a straightforward modification of our approach.

Remark (scoring losses): For $d = 1$, $\Gamma_{\text{tri}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} (\mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)} \times \mathbb{A}_R^{N^2}) \to \mathbb{R}$ can be seen as a *scoring function*, which returns the log-likelihood of an energy-based probability that the corresponding triplet appears in the knowledge graph. For knowledge graph completion tasks, we aim to assign high scores for triplet edges that appears or are missing in the KG. Different from traditional *scoring function* in the literature (Yang et al., 2015; Trouillon et al., 2016; Chen et al., 2022), the invariant triplet representation has additional invariance properties.

Similar to (Srinivasan & Ribeiro, 2020), we can define the most expressive invariant triplet representation.

Definition 4.6 (Most-expressive invariant triplet representation)**.** An invariant triplet representation Γ_{tri} is most expressive iff $\forall \mathbf{A}_G, \mathbf{A}_H \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}, \forall (i, r, j) \in \mathcal{V}_G \times \mathcal{R}_G \times \mathcal{V}_G, \forall (i', r', j') \in \mathcal{V}_H \times \mathcal{R}_H \times \mathcal{V}_H$, we will have $\Gamma_{\text{tri}}((i, r, j), \mathbf{A}_G) = \Gamma_{\text{tri}}((i', r', j'), \mathbf{A}_H)$ iff (i, r, j) and (i', r', j') are isomorphic triplets (Definition 4.4).

In what follows we define representations for the whole KG (akin to how GNNs provide representations for a whole graph), which we denote as *double equivariant KG representations*.

Definition 4.7 (Double equivariant KG representations)**.** Let $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ be a KG following Definition 4.1. A function $\Gamma_{\text{gra}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2} \to \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}, d \geq 1$ is double equivariant w.r.t. arbitrary node $\phi \in \mathbb{S}_N$ and relation $\tau \in \mathbb{S}_R$ permutations, if $\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}) = \phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A})$. Moreover, valid mappings of Γ_{gra} must map a domain element to an image element with the same number of nodes N and relations R.

Remark The node equivariance for KG is double equivariance when relation permutation τ is fixed to be identity; The relation equivariance for KG is double equivariance when node permutation ϕ is fixed to be identity.

Next, we connect Definitions 4.5 and 4.7.

Theorem 4.8. *For all $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$, given an invariant triplet representation Γ_{tri} we can construct a double equivariant representation as $(\Gamma_{gra}(\mathbf{A}))_{i,r,j,:} := \Gamma_{tri}((i, r, j), \mathbf{A})$, $\forall (i, r, j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, and vice-versa.*

Section 5 will introduce a double equivariant neural architecture based on Theorem 4.8. However, first we want to discuss the consequences of invariant representations, and how it can benefit KG tasks.

4.1. Consequences of invariant predictors in KGs

We will now analyze two KG completion tasks that are effectively impossible for all standard KG completion methods (based on attributed multigraphs), which are relatively easy for predictors based on our invariant KG representations.

Consider the knowledge base in Figure 2, obtained from a fictional alien civilization with 3 KGs for training and one for test. Knowing nothing about alien language and costumes, we note that in training all KG relations are different. Minimally, we could predict the missing relation in red in test data is not "$\not\leq$". Note, however, that because all edge attributes are unique, assuming the KG is an attributed (multi)graph does not allow us to automatically infer this obvious logical rule, since whatever rules are learned for one relation are not directly applicable to others.

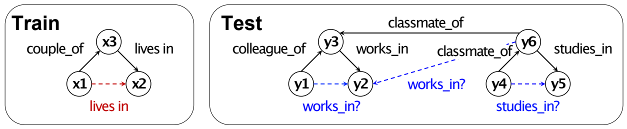


Figure 3. Example of an inductive KG completion task with new relations that can be explained by our Horn clauses.

Now let's consider sending our aliens a KG with a biological human relationships in Figure 1(a). Given a set of biological male relations as training data in Figure 1(a), under what assumptions could the alien be able to predict (without knowledge of our language or physiology) the relation "(Ellie, Grand ∧ Mother, Hanna)" between Ellie and Hanna in the (hold-out) test data of Figure 1(b)?

Thankfully, the tasks in Figures 1 and 2 can both be solved under our definition of KG (Definition 4.1). Due to required invariance, any triplet representation $(i, r, j) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ in either train or test (Definition 4.5) can only pay attention to the structural relations between nodes and their relations, not their absolute ids (node id and relation id). In the KG of Figure 2, any representation invariant to both permutations in training can only encode that any relation is unlike any other relation, that is, a self-supervised trained predictor created by removing the triplet $(\pitchfork, \smile, \multimap)$ and trying to predict it back must predict a uniform distribution over the remaining relations $\mathcal{R}^{\text{train}} \backslash \{\bumpeq\}$. If all train KGs are treated as a single (disconnected) KG, the uniform prediction is over $\mathcal{R}^{\text{train}} \backslash \{\Delta, \dagger, \bumpeq, \frown, \nsim\}$. In test, this predictor would predict the relation "?" uniformly over the set $\mathcal{R}^{\text{test}} \backslash \{\nless\}$, which is really all we know about the aliens.

In the task of Figure 1(a), once we remove (Bob, Grand ∧ Father, Hans) for training (via self-supervision), any invariant triplet predictor for the pair (Bob, Hans) that can correctly predict back the triplet (Bob, Grand ∧ Father, Hans) based on (1-hop) neighbor information from Bob and Hans in training must also be able to predict (Ellie, Grand ∧ Mother, Hanna) in the test KG of Figure 1(b). This is because, restricted to their respective 1-hop neighborhoods, the triplet (Bob, Grand ∧ Father, Hans) in the training KG of Figure 1(a) is isomorphic (Definition 4.4) to the triplet (Ellie, Grand ∧ Mother, Hanna) in the test KG of Figure 1(b).

4.2. Connection to Learning Logical Rules

We now define universally quantified entity and relation Horn clauses for our tasks, and show that any predictor that can be learned from the invariant triplet representation in Definition 4.2 has an equivalent predictor as a conjunction of such Horn clauses.

Definition 4.9 (UQER Horn clauses: Universally quantified entity and relation Horn clauses)**.** We define a subset of universally quantified Horn clauses involving K relations of M entities, defined by an indicator tensor $\mathbf{B} \in \{0,1\}^{M \times K \times M}$:

$$\begin{aligned} &\forall C_1 \in \mathcal{R}, (\forall C_r \in \mathcal{R} \setminus \{C_1, \ldots, C_{r-1}\})_{r=2}^{K}, \\ &\forall E_1 \in \mathcal{V}, (\forall E_i \in \mathcal{V} \setminus \{E_1, \ldots, E_{i-1}\})_{i=2}^{M}, \\ &\bigwedge_{\substack{i,j=1,\ldots,M, \\ r=1,\ldots,K, \\ \mathbf{B}_{i,r,j}=1}} (E_i, C_r, E_j) \implies (E_1, C_1, E_h), \end{aligned} \tag{1}$$

for any relation set $\mathcal{R}$ and entity set $\mathcal{V}$ s.t., $|\mathcal{R}| \geq K, |\mathcal{V}| \geq M$, $h \in \{1, 2\}$ (where $h = 1$ indicates a self-loop relation and/or a node attribute), where if $M \geq 3, \forall a \in \{3, \ldots, M\}$, $\sum_{m=3}^{M} \sum_{r'=1}^{K} \mathbf{B}_{m,r',a} + \mathbf{B}_{a,r',m} \geq 1$.

Note that our definition of UQER Horn clauses (Definition 4.9) is a generalization of the first order logic (FOL) clauses in (Yang et al., 2017; Meilicke et al., 2018; Sadeghian et al., 2019; Teru et al., 2020) such that the relations in the Horn clauses are also universally quantified rather than predefined constants. Note that our Horn clauses need not to form a path in the KG, since some relevant associations between relations could be in disconnected subgraphs.

Figure 3 exemplifies the connection between Definition 4.9 and our KG definition (Definition 4.1). In the training KG, we can see that $(\text{x1}, \text{couple\_of}, \text{x3}) \wedge (\text{x3}, \text{lives\_in}, \text{x2}) \implies (\text{x1}, \text{lives\_in}, \text{x2})$. According to that, we may simply learn that, in a KG, for any two different relations in $\mathcal{R}$ and any three different entities in $\mathcal{V}$, *if they form a logic chain of length* 2 *with distinct relations, then the second relation on the chain also exists between the source and destination entities of the chain.* Using Equation (1) we would write this as $\forall C_1 \in \mathcal{R}, \forall C_2 \in \mathcal{R} \setminus \{C_1\}, \forall E_1 \in \mathcal{V}, \forall E_2 \in \mathcal{V} \setminus \{E_1\}, \forall E_3 \in \mathcal{V} \setminus \{E_1, E_2\}$, $(E_1, C_1, E_3) \wedge (E_3, C_2, E_2) \implies (E_1, C_2, E_2)$.

Then, on the test KG in Figure 3, we will apply the above UQER Horn clause learned from training to predict all missing positive triplets. For instance, an arbitrary variable allocation, "classmate_of", "studies_in" and entities y4, y5, y6, allows all conjunctive conditions of our Horn clause to be satisfied, thus predicting $(\text{y4}, \text{studies\_in}, \text{y5})$ as a positive triplet. Two other triplets can similarly be predicted in dashed blue in Figure 3.

We now connect our double-invariant triplet representations in Definition 4.5 with the UQER Horn clauses in Definition 4.9.

Theorem 4.10. *Given an arbitrary triplet predictor* $\eta : \mathbb{R}^d \to \{0,1\}$ *that takes the triplet representation* $\Gamma_{\textit{tri}}((i, r, j), \mathbf{A})$ *in Definition 4.5 as input,* $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$*, and predicts if* $(i, r, j) \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$ *is a positive triplet, there exists a set of UQER Horn clauses (Definition 4.9) that predicts the same positive triplets for all* $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ *and* $(i, r, j) \in$

$\cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$.

The full proof in Appendix A shows how the universal quantification in Definition 4.9 implies a double-invariant predictor, where we can construct a set of Horn clauses and for each Horn clause, the left side are observed facts in KG and the right side is triplet predicted to be positive. By adding the UQER property, as a permutation equivariance of nodes and relations, the set of Horn clauses still holds, where the predictor η based on our invariant triplet representations Γ_{tri} (Definition 4.5) also gives the same predictions.

5. Inductive Double-Exchangeable Neural Architecture for KGs

In this section we propose a model to learn invariant triplet representation for KGs. By Theorem 4.8, one way to obtain invariant triplet representation is to learn a double equivariant function (Definition 4.7). So we propose an inductive structural doubly-exchangeable architecture to learn double equivariant functions over KG.

We start by looking at Definition 4.7 from another point of view. Consider $\mathbf{A}^{(N,R)}$ given by Definition 4.1. Denote $A^{(r)}$ as the matrix $A^{(r)}_{i,j} = (\mathbf{A}^{(N,R)})_{i,r,j}$, $r \in \mathcal{R}^{(R)}$. Note that the KG can be written as $\mathbf{A}^{(N,R)} = (A^{(1)}, \ldots, A^{(R)})$. Since the actions of the two permutation groups $\mathbb{S}_N$ and $\mathbb{S}_R$ commute, the double equivariance in Definition 4.7 over $\mathbf{A}^{(N,R)}$ can be described as a $\phi \in \mathbb{S}_N$ (graph) equivariance over $A^{(r)}, r = 1, \ldots, R$, and a $\tau \in \mathbb{S}_R$ (set) equivariance (over the set of homogeneous graphs). Hence, our double equivariance can make use of the general framework proposed by Maron et al. (2020).

We start with a linear double-equivariant layer composed by a Siamese layer to define the k-th linear double-equivariant layer $L^{(t)} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2} \to \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{R}^{N \times R \times N \times d_t}$ as follows, for each $r = 1, ..., R$:

$$(L^{(t)}(\mathbf{A}^{(N,R)}))_{:,r} = L_1^{(t)}(A^{(r)}) + L_2^{(t)}\Big(\sum_{r' \in \mathcal{R} \setminus \{r\}} A^{(r')}\Big), \tag{2}$$

where $t = 1, \ldots, T$, $T \geq 2$, $L_1^{(t)}, L_2^{(t)} : \cup_{N=2}^{\infty} \mathbb{A}_1^{N^2} \to \cup_{N=2}^{\infty} \mathbb{R}^{N \times N \times d_t}$ can be any GNN layers that outputs pairwise representations. The sum $\sum_{r' \in \mathcal{R} \setminus \{r\}} A^{(r')}$ can also be replaced by other set aggregators such as mean, max, etc.. Our implementation uses the max aggregator, where $\max\big(\{A^{(r')}\}_{r' \in \mathcal{R} \setminus \{r\}}\big)$ only cares if a pair of nodes is connected (no matter the edge attribute). Note that the proposed layer is similar to the H-equivariant layers proposed by Bevilacqua et al. (2021) for increasing the expressiveness of GNN using sets of subgraphs (a markedly different task than ours). We now can define our (double-equivariant) neural network for KGs:

Definition 5.1 (Double-equivariant neural network)**.** The double-equivariant network $\Gamma_{\text{gra}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2} \to \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}$ is defined by several linear double equivariant layers described in Equation (2) interleaved with non-polynomial activation functions to obtain triplets representations,

$$\Gamma_{\text{gra}}(\mathbf{A}) = L^{(T)}(\cdots \sigma(L^{(2)}(\sigma(L^{(1)}(\mathbf{A})))) \cdots), \tag{3}$$

where σ is the non-polynomial activation function (our implementations uses ReLU).

5.1. Implementation considerations

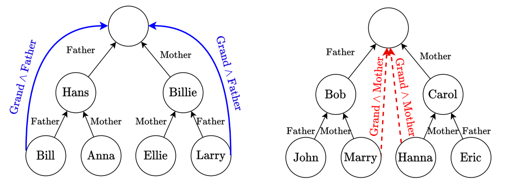
Most-expressive pairwise representations for $L_1^{(k)}, L_2^{(k)}$ are computationally expensive. Moreover, even less expressive pairwise GNN layers in Equation (2), such as Zhang & Chen (2018); Zhu et al. (2021); Zhang et al. (2021); Zhou et al. (2022), are still expensive (computationally and memory-wise). Thus, we propose inductive structural doubly-exchangeable architecture (IS-DEA), and implementation of Equation (3) that trade-offs expressivity for speed by using equivariant GNN layers (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018) for node representations. Specifically, for a KG $\mathbf{A}^{(N,R)}$, IS-DEA performs vertex message passing through two learnable functions, such as MLPs, recursively over T layers $\{L^{(t)}\}_{t=1}^{T}$.

At each iteration $t \in \{1, 2, ..., T\}$, all vertices $i \in \mathcal{V}^{(N)}$ are associated with a learned vector $h_i^{(t)} \in \mathbb{R}^{R \times d_t}, d_t \geq 1$. Since we do not assume our KGs have node attributes, we consider initializing $h_i^{(0)} = \mathbb{1}$. Then we recursively compute the update, $\forall i \in \mathcal{V}^{(N)}, \forall r \in \mathcal{R}^{(R)}$,

$$\begin{aligned} h_{i,r}^{(t+1)} = \text{MLP}_1^{(t)}\Big(h_{i,r}^{(t)}, \sum_{j \in \mathcal{N}_r(i)} h_{j,r}^{(t)}\Big) \\ + \text{MLP}_2^{(t)}\Big(\sum_{r' \neq r}^{R} h_{i,r'}^{(t)}, \sum_{j \in \cup_{r' \neq r} \mathcal{N}_{r'}(i)} \Big(\sum_{r' \neq r}^{R} h_{j,r'}^{(t)}\Big)\Big), \end{aligned} \tag{4}$$

where $\text{MLP}_1^{(t)}$ and $\text{MLP}_2^{(t)}$ denotes two multi-layer perceptron for the Siamese and aggregation function, $\mathcal{N}_r(i)$ denotes the neighborhood set of i with relation r in the graph, $\mathcal{N}_r(i) = \{j | (i, r, j) \in \mathcal{S} \text{ or } (j, r, i) \in \mathcal{S}\}$, where $\mathcal{S}$ is the set of KG triplets of $\mathbf{A}^{(N,R)}$, and $\cup_{j \neq r} \mathcal{N}_r(i)$ denotes the neighborhood set of i in the graph $A^{(r)}$. In our implementation, we use GIN (Xu et al., 2019a) as our GNN architecture, which satisfies Equation (4). At the final layers, we use standard MLPs (which does not take neighborhood information as input) to output a final prediction.

As shown by Srinivasan & Ribeiro (2020); You et al. (2019), structural node representations are not most-expressive for link prediction task in homogeneous graphs. The same issue happens for KGs. To ameliorate the issue, we concatenate i and j (double-equivariant) node representations with the



(a) Training KG (b)Test KG (new nodes)

Figure 4. **Simplified Example of FD-1 Generation.** Family tree where negative (positive) training triplets "Grand ∧ Mother" ("Grand ∧ Father") become positive (negative) triplets in test (in this task the KG is finitely exchangeable and a perfect predictor to predict dashed red test edges does not exist), while our assumption is for FD-2 is similar but adds 2 extra relations in the test KG.

distance between i and j in our triplet representation (appending distances is also adopted in the representations of Teru et al. (2020); Galkin et al. (2021)). Finally, we obtain the triplet representation $\forall (i, r, j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$

$$\Gamma_{\text{IS-DEA}}((i, r, j), \mathbf{A}) = (h_{i,r}^{(T)} \,\|\, h_{j,r}^{(T)} \,\|\, d(i, j) \,\|\, d(j, i)), \quad (5)$$

where we denote $d(i, j)$ as the length of the non-trivial shortest path from i to j without direct connection between i and j in the KG, $||$ as the concatenation operation. Since the KG is directed, we concatenate distance on both directions.

Lemma 5.2. *The triplet representation in Equation (5) is an invariant triplet representation as per Definition 4.5.*

Finally, as in previous KG works (Yang et al., 2015; Schlichtkrull et al., 2018; Zhu et al., 2021), we use negative sampling in our training procedure, where for each training triplet $(i, r, j) \in \mathcal{S}$, we randomly corrupt either the subject or object n times to generate the negative example. Following Schlichtkrull et al. (2018), we use cross-entropy loss for model optimization to obtain predictions that will score positive examples higher than negative examples:

$$\mathcal{L} = -\sum_{(i,r,j)\in\mathcal{S}} \Big(\log(\Gamma_{\text{tri}}((i, r, j), \mathbf{A})) - \frac{1}{n}\sum_{p=1}^{n} \log(1 - \Gamma_{\text{tri}}((i'_p, r, j'_p), \mathbf{A})) \Big), \quad (6)$$

where (i'_p, r, j'_p) are the p-th negative examples.

6. Experiments

We evaluate IS-DEA on two synthetic tasks (that we propose to test the generalization capabilities of our method), FD-1 and FD-2; and on three inductive knowledge graph completion datasets, WN18RR, FB237 and NELL995-v1 (each with 4 inductive splits), which are widely-used inductive knowledge graph completion benchmarks in literature (Teru et al., 2020; Zhu et al., 2021). We also construct new inductive knowledge graph completion tasks based on those datasets which requires generalization capabilities to predict new relations to show the value of our method in real world. In all results, we report mean performance over 5 runs, which is consistent with the reporting of Teru et al. (2020); Zhu et al. (2021). We also report variance of our method in real-world tasks (Table 3). More experiment details including baselines, implementation details and ablation studies can be found in Appendix D.

Model	FD-1				FD-2			
	MRR↑	Hits@1↑	Hits@2↑	Hits@4↑	MRR↑	Hits@1↑	Hits@2↑	Hits@4↑
Neural LP	0.502	0.339	0.415	0.651	N/A	N/A	N/A	N/A
DRUM	0.502	0.339	0.415	0.651	N/A	N/A	N/A	N/A
GraIL	0.422	0.181	0.416	0.740	N/A	N/A	N/A	N/A
NBFNet	0.311	0.262	0.318	0.322	N/A	N/A	N/A	N/A
IS-DEA	**0.832**	**0.700**	**0.903**	**1.000**	**0.853**	**0.730**	**0.935**	**1.000**

Table 1. **Inductive performance on Family Diagrams.** On FD-1, existing baselines clearly struggle to perform the task; On FD-2, only our method (IS-DEA) is able to perform the task.

6.1. Synthetic Experiments

In the synthetic experiments, we propose two challenging family tree completion tasks in order to verify the theoretical benefits of our model: FD-1 is used to show that our model is insensitive to relation identity; FD-2 is used to show that our model can automatically generalize to new nodes and relations. These tasks are described next.

Family Diagram 1 (FD-1). A simplified version of FD-1 is illustrated in Figure 4. Given fact and missing positive triplets in a training KG, the goal is to learn a model which can detect missing positive triplets given fact triplets on a different test KG whose relations have different meanings (Appendix D.1 gives more details). The results on FD-1 are shown at Table 1 (left). IS-DEA significantly outperforms all baselines. This task tests the relation-invariance property of IS-DEA.

Interestingly, the baselines that tend to perform better on real-world KGs (e.g., NBFNet (Zhu et al., 2021), GraIL (Teru et al., 2020)) tend to perform worse on FD-1. This is because training and test queries are conflicting: Positive triplet queries in the training graph are negative queries in test, while positive test queries become negative in training. Hence, for models that assume attributed multigraphs (i.e., exchangeable but not double exchangeable multigraphs), the better it can perform on triplets similar to the ones seeing in training, the worse it will perform on the test data.

Family Diagram 2 (FD-2). The FD-2 task is an extension of the scenario described in Figure 3. The learning goal is similar to FD-1. Besides the relation meanings are different from training to test, test KG in FD-2 has more number of relations than training KG. Please refer to Appendix D.1 for further details. On FD-2, training has 127 nodes and 2 relations, while test has 254 nodes and 4 relations (more nodes and more relations). Thus, N/A in Table 1 expresses that

Dataset	MRR↑		Hits@1↑		Hits@5↑		**Hits@10↑**	
	Random	IS-DEA	Random	IS-DEA	Random	IS-DEA	Random	IS-DEA
FB237-v1	0.090	**0.634**	0.020	**0.552**	0.100	**0.750**	0.200	**0.802**
FB237-v2	0.090	**0.656**	0.020	**0.563**	0.100	**0.736**	0.200	**0.813**
NELL995-v2	0.090	**0.806**	0.020	**0.714**	0.100	**0.929**	0.200	**0.929**
NELL995-v3	0.090	**0.680**	0.020	**0.635**	0.100	**0.709**	0.200	**0.709**
NELL995-v4	0.090	**0.615**	0.020	**0.533**	0.100	**0.700**	0.200	**0.800**

Table 2. **Performance of training with 90% relations, where test asks to predict the triplets over the discarded 10% relations.** IS-DEA is far better than random guess. **There are no other baselines in this table since baselines cannot perform this task.**

Dataset			Neural LP	DRUM	GraIL	NBFNet	IS-DEA
WN18RR	v1	Original	0.825	0.823	0.840	0.949	**0.974** (0.020)
		Permuted	0.686	0.745	0.822	0.630	**0.974** (0.020)
	v2	Original	0.839	0.847	0.816	**0.960**	0.959 (0.033)
		Permuted	0.772	0.837	0.816	0.833	**0.959** (0.033)
	v3	Original	0.607	0.611	0.633	0.907	**0.954** (0.017)
		Permuted	0.594	0.604	0.633	0.716	**0.954** (0.017)
	v4	Original	0.752	0.746	0.763	0.890	**0.909** (0.001)
		Permuted	0.628	0.698	0.763	0.474	**0.909** (0.001)
FB237	v1	Original	0.529	0.529	0.741	0.895	**0.960** (0.007)
		Permuted	0.424	0.422	0.717	0.605	**0.960** (0.007)
	v2	Original	0.589	0.587	0.884	0.977	**0.982** (0.003)
		Permuted	0.478	0.510	0.864	0.707	**0.982** (0.003)
	v3	Original	0.529	0.529	0.828	**0.980**	0.976 (0.006)
		Permuted	0.443	0.407	TL	0.741	**0.976** (0.006)
	v4	Original	0.559	0.559	0.893	**0.987**	0.981 (TL)
		Permuted	0.372	0.375	TL	0.725	**0.981** (TL)
NELL995	v1	Original	0.500	0.500	0.932	**0.995**	0.975 (0.043)
		Permuted	0.500	0.500	0.734	**0.995**	0.975 (0.043)
	v2	Original	0.787	0.786	0.963	**0.978**	0.960 (0.008)
		Permuted	0.384	0.425	0.899	**0.978**	0.960 (0.008)
	v3	Original	0.827	0.827	0.958	**0.969**	0.957 (0.033)
		Permuted	0.397	0.441	0.947	**0.968**	0.957 (0.033)
	v4	Original	0.806	0.806	0.880	**0.957**	0.906 (0.039)
		Permuted	0.340	0.352	0.806	**0.937**	0.906 (0.039)

Table 3. **Hits@10 performance on inductive knowledge graph completion.** IS-DEA reaches state-of-the-art performance on all tasks, and is the invariant to relation permutation on all tasks. TL means a baseline takes "Too Long" time to finish on a dataset. We will report the results as soon as we finish it. For our model, we also report standard deviation in parentheses.

none of our baselines can perform this task (since, as they assume an attributed multigraph as input, they all need to learn parameters for each relation). Since IS-DEA does not learn parameters specific to relations, it is the only method that can inductively infer over a KG with new and more relations in test, and achieving very good performance on FD-2 as shown in Table 1 (right).

6.2. Real-world Knowledge graphs

As far as we know, there are no real-world benchmarks where training and test KGs have distinct nodes and relations. Therefore, our real-world evaluation of inductive knowledge graph completion is limited to tasks that existing methods can also perform. We select 12 inductive knowledge graph completion benchmarks, 4 inductive splits of 3 datasets WN18RR and FB237 and NELL995 to test our proposal. In order to highlight the relation-invariance property of our proposal, we also perform a task where all relation IDs are randomly shuffled only in test.

Our results for inductive knowledge graph completion are reported in Table 3. We can see that IS-DEA results are always invariant to the permutation of relations in test, while all baselines become worse at least on one dataset if relations are permuted in test. Besides, IS-DEA obtains state-of-the-art score on the key metric Hits@10, and particularly, is mostly the best for WN18RR and FB237. More detailed results on MRR, Hits@1, Hits@5 can be found in Table 5, Table 6 and Table 7 in the appendix.

While double exchangeability may not be the right assumption for all KGs, it is clearly beneficial for some KGs. Our experiments treat all relations as exchangeable. Further research is needed to better understand which relations are exchangeable and which are not for a given KG. We also believe that using true pairwise representation can improve the performance of IS-DEA.

We also note that inductive knowledge graph completion are easy tasks that do not test the full capabilities of IS-DEA. Thus, we construct new inductive tasks with new relations based on those datasets to test IS-DEA ability to extrapolate new relations in test.

6.3. Knowledge graphs with New Test Relations

We construct our own tasks to simulate the scenario when new relations are invented to existing KGs. In our construction, we first pick a dataset with sufficient amount of relations; Then, we uniformly discard 10% relations from its training and validation triplets (remove triplets with those relations); Finally, in the test stage, we still keep all 100% relations in observed triplets, but only keep test triplets with discarded 10% relations. In our experiments, those hidden relations are $2\% \sim 11\%$ of original test inputs, and are $3\% \sim 14\%$ of original test target. The result are shown in Table 2. Since the task requires to predict new relations, existing methods are not applicable, while our method can achieve far better performance than random guess.

Ablation Study. Since negative samplings are drawn by uniformly corrupting object (without loss of generality), it is very likely that corrupted objects are far way from subject while true object is close to subject. Under such scenario, shortest distance itself will be a powerful enough feature to achieve good ranking performance in knowledge graph completion, thus we want to know if shortest distance feature augmentation contributes to the performance gain. We perform ablation study on two dataset NELL995-V2 and WN18RR-v4. As shown in Table 4, even if shortest distance is excluded from our model, it still performs quite well.Thus, we can say that double-equivariant representation itself is enough to provide good performance. Besides, we also show in Table 4 that shortest distance itself is not enough for knowledge graph completion.

Limitations. IS-DEA excels both in synthetic and real-world benchmarks. However, the simplification from pair-

Dataset	Hits@10↑		
	w/ Distance	w/o Distance	Distance MLP
WN18RR-v4	0.909 (0.001)	0.908 (0.005)	0.415 (0.000)
NELL995-v2	0.960 (0.008)	0.959 (0.009)	0.819 (0.001)

Table 4. **Performance with/without Shortest Distances in our proposed architecture, and an MLP only on Shortest Distance.** Even without shortest distance as augmented feature, our proposal still outperforms all baselines in real-world tasks. However only using shortest distance shows a clear gap between scoring triplets by a MLP only on shortest distance and scoring triplets by IS-DEA. Thus shortest distance can be a powerful feature for knowledge graph completion, but is not sufficient for good performance.

wise to node embeddings in IS-DEA limits its expressivity. In Appendix D.4, we give a synthetic counterexample how this could be an issue in some KGs. Moreover, IS-DEA has the same poor pre-processing scalability as GraIL. We leave addressing these limitations as future work (see Appendix E). We also do not envision a direct negative social impact of our work.

7. Conclusions

In this work we introduced the concept of double exchangeable attributed graphs as a formal model for KGs, challenging the view that KGs are attributed graphs (with exchangeable node ids). We showed that, similar to how node id symmetries impose learning structural node embeddings in homogeneous graphs, double symmetries (node and relation ids) impose structural rule learning in KGs. We then introduced a blueprint for double equivariant neural network architectures for KGs, which adapts permutation-equivariance to both KG entities and relations. We showed this architecture can learn logical rules that standard KG methods cannot. Finally, experiments showed that even a simple double exchangeable architecture (IS-DEA) achieves promising results in inductive KG completion tasks.

References

Aldous, D. J. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

Barceló, P., Kostylev, E., Monet, M., Pérez, J., Reutter, J., and Silva, J.-P. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

Barcelo, P., Galkin, M., Morris, C., and Orth, M. R. Weisfeiler and leman go relational. *arXiv preprint arXiv:2211.17113*, 2022.

Bevilacqua, B., Frasca, F., Lim, D., Srinivasan, B., Cai, C., Balamurugan, G., Bronstein, M. M., and Maron, H. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2021.

Bordes, A., Weston, J., Collobert, R., and Bengio, Y. Learning structured embeddings of knowledge bases. In *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

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

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, 2017.

Chamberlain, B. P., Shirobokov, S., Rossi, E., Frasca, F., Markovich, T., Hammerla, N., Bronstein, M. M., and Hansmire, M. Graph neural networks for link prediction with subgraph sketching. *arXiv preprint arXiv:2209.15486*, 2022.

Chen, Y., Mishra, P., Franceschi, L., Minervini, P., Stenetorp, P., and Riedel, S. Refactor gnns: Revisiting factorisation-based models from a message-passing perspective. In *Advances in Neural Information Processing Systems*, 2022.

Chen, Z., Chen, L., Villar, S., and Bruna, J. Can graph neural networks count substructures? In *Advances in Neural Information Processing Systems*, 2020a.

Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., and Duan, Z. Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456, 2020b.

Cheng, K., Liu, J., Wang, W., and Sun, Y. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 179–189, 2022.

De Raedt, L. *Logical and relational learning*. Springer Science & Business Media, 2008.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Galárraga, L. A., Teflioudi, C., Hose, K., and Suchanek, F. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the*

22nd international conference on World Wide Web, pp. 413–422, 2013.

Galkin, M., Denis, E., Wu, J., and Hamilton, W. L. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *International Conference on Learning Representations*, 2021.

Galkin, M., Zhu, Z., Ren, H., and Tang, J. Inductive logical query answering in knowledge graphs. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pp. 729–734, 2005.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Heckerman, D., Meek, C., and Koller, D. Probabilistic entity-relationship models, prms, and plate models. *Introduction to statistical relational learning*, 2007:201–238, 2007.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Kejriwal, M., Knoblock, C. A., and Szekely, P. *Knowledge graphs: Fundamentals, techniques, and applications*. MIT Press, 2021.

Kersting, K. and De Raedt, L. Bayesian logic programming: theory and tool. *Statistical Relational Learning*, pp. 291, 2007.

Kipf, T. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Koller, D., Friedman, N., Džeroski, S., Sutton, C., McCallum, A., Pfeffer, A., Abbeel, P., Wong, M.-F., Meek, C., Neville, J., et al. *Introduction to statistical relational learning*. MIT press, 2007.

Kroetsch, M. and Weikum, G. Special issue on knowledge graphs. *Journal of Web Semantics*, 37(38):53–54, 2016.

Lao, N. and Cohen, W. W. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

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

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, pp. 2156–2167, 2019.

Maron, H., Litany, O., Chechik, G., and Fetaya, E. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pp. 6734–6744. PMLR, 2020.

Meilicke, C., Fink, M., Wang, Y., Ruffinelli, D., Gemulla, R., and Stuckenschmidt, H. Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In *International semantic web conference*, pp. 3–20. Springer, 2018.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 4602–4609, Jul. 2019.

Murphy, R., Srinivasan, B., Rao, V., and Ribeiro, B. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *International Conference on Learning Representations*, 2019a.

Murphy, R., Srinivasan, B., Rao, V., and Ribeiro, B. Relational pooling for graph representations. In *International Conference on Machine Learning*, pp. 4663–4673. PMLR, 2019b.

Murphy, R., Srinivasan, B., Rao, V., and Ribeiro, B. Relational pooling for graph representations. In *Proceedings of the 36th International Conference on Machine Learning*, 2019c.

Neville, J. and Jensen, D. Relational dependency networks. *Journal of Machine Learning Research*, 8(3), 2007.

Nickel, M., Tresp, V., and Kriegel, H.-P. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

Nickel, M., Rosasco, L., and Poggio, T. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3): 489–508, 2017.

Qian Huang, H. R. and Leskovec, J. Few-shot relational reasoning via connection subgraph pretraining. In *Neural Information Processing Systems*, 2022.

Raedt, L. D., Kersting, K., Natarajan, S., and Poole, D. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis lectures on artificial intelligence and machine learning*, 10(2):1–189, 2016.

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

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Sadeghian, A., Armandpour, M., Ding, P., and Wang, D. Z. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Schneider, E. W. Course modularization applied: The interface system and its implications for sequence control and data analysis. 1973.

Schulman, J., Zoph, B., Kim, C., Hilton, J., Menick, J., Weng, J., Uribe, J. F. C., Fedus, L., Metz, L., Pokorny, M., Lopes, R. G., Zhao, S., Vijayvergiya, A., Sigler, E., Perelman, A., Voss, C., Heaton, M., Parish, J., Cummings, D., Nayak, R., Balcom, V., Schnurr, D., Kaftan, T., Hallacy, C., Turley, N., Deutsch, N., Goel, V., Ward, J., Konstantinidis, A., Zaremba, W., Ouyang, L., Bogdonoff, L., Gross, J., Medina, D., Yoo, S., Lee, T., Lowe, R., Mossing, D., Huizinga, J., Jiang, R., Wainwright, C., Almeida, D., Lin, S., Zhang, M., Xiao, K., Slama, K., Bills, S., Gray, A., Leike, J., Pachocki, J., Tillet, P., Jain, S., Brockman, G., and Ryder, N. ChatGPT: Optimizing language models for dialogue. Official OpenAI Blog, November 2022.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Shen, T., Zhang, F., and Cheng, J. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, pp. 109597, 2022.

Singhal, A. Introducing the Knowledge Graph: things, not strings. Official Google Blog, May 2012.

Sinha, K., Sodhani, S., Dong, J., Pineau, J., and Hamilton, W. L. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*, 2019.

Sinha, K., Sodhani, S., Pineau, J., and Hamilton, W. L. Evaluating logical generalization in graph neural networks. *arXiv preprint arXiv:2003.06560*, 2020.

Srinivasan, B. and Ribeiro, B. On the equivalence between positional node embeddings and structural graph representations. In *Eighth International Conference on Learning Representations*, 2020.

Sun, Z., Hu, W., Zhang, Q., and Qu, Y. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18, 2018.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

Sun, Z., Wang, C., Hu, W., Chen, M., Dai, J., Zhang, W., and Qu, Y. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 222–229, 2020.

Sutskever, I., Tenenbaum, J., and Salakhutdinov, R. R. Modelling relational data using bayesian clustered tensor factorization. *Advances in neural information processing systems*, 22, 2009.

Teru, K., Denis, E., and Hamilton, W. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pp. 9448–9457. PMLR, 2020.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. Complex embeddings for simple link prediction. In *International conference on machine learning*, pp. 2071–2080. PMLR, 2016.

Trouillon, T., Dance, C., Gaussier, E., Welbl, J., Riedel, S., and Bouchard, G. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research*, 18(130):1–38, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *ICLR*, 2018.

Wang, H., Ren, H., and Leskovec, J. Relational message passing for knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1697–1707, 2021.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

Wang, Z., Lv, Q., Lan, X., and Zhang, Y. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 349–357, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019a.

Xu, K., Wang, L., Yu, M., Feng, Y., Song, Y., Wang, Z., and Yu, D. Cross-lingual knowledge graph alignment via graph matching neural network. In *Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), 2019b.

Yan, Y., Liu, L., Ban, Y., Jing, B., and Tong, H. Dynamic knowledge graph alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4564–4572, 2021.

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

Yang, F., Yang, Z., and Cohen, W. W. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

You, J., Ying, R., and Leskovec, J. Position-aware graph neural networks. In *International conference on machine learning*, pp. 7134–7143. PMLR, 2019.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

Zhang, M., Li, P., Xia, Y., Wang, K., and Jin, L. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34:9061–9073, 2021.

Zhou, Y., Kutyniok, G., and Ribeiro, B. OOD link prediction generalization capabilities of message-passing GNNs in larger test graphs. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Zhu, Z., Zhang, Z., Xhonneux, L.-P., and Tang, J. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.

A. Proofs

Theorem 4.8. *For all $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$, given an invariant triplet representation Γ_{tri} we can construct a double equivariant representation as $(\Gamma_{gra}(\mathbf{A}))_{i,r,j,:} := \Gamma_{tri}((i,r,j),\mathbf{A})$, $\forall (i,r,j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, and vice-versa.*

Proof. ($\Rightarrow$) For any KG $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ with N nodes and R relations, assume $\Gamma_{\text{tri}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} (\mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)} \times \mathbb{A}_R^{N^2}) \to \mathbb{R}^d, d \geq 1$ is an invariant triplet representation as in Definition 4.5. Using the invariant triplet representation, we can define a function $\Gamma_{\text{gra}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}^{N^2} \to \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}$ such that $\forall (i,r,j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, $(\Gamma_{\text{gra}}(\mathbf{A}))_{i,r,j,:} = \Gamma_{\text{tri}}((i,r,j),\mathbf{A})$. Then $\forall \phi \in \mathbb{S}_N, \forall \tau \in \mathbb{S}_R$, $(\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}))_{\phi \circ i, \tau \circ r, \phi \circ j,:} = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ r, \phi \circ j), \phi \circ \tau \circ \mathbf{A})$. We know $\Gamma_{\text{tri}}((i,r,j),\mathbf{A}) = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ r, \phi \circ j), \phi \circ \tau \circ \mathbf{A})$. Thus we conclude, $\forall \phi \in \mathbb{S}_N, \forall \tau \in \mathbb{S}_R, \forall (i,r,j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, $(\phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}))_{\phi \circ i, \tau \circ r, \phi \circ j,:} = (\Gamma_{\text{gra}}(\mathbf{A}))_{i,r,j,:} = \Gamma_{\text{tri}}((i,r,j),\mathbf{A}) = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ r, \phi \circ j), \phi \circ \tau \circ \mathbf{A}) = (\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}))_{\phi \circ i, \tau \circ r, \phi \circ j,:}$. In conclusion, we show that $\phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}) = \Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A})$, which proves the constructed Γ_{gra} is a double equivariant representation as in Definition 4.7.

($\Leftarrow$) For any KG $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}^{N^2}$ with N nodes and R relations, assume $\Gamma_{\text{gra}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}^{N^2} \to \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}$ is a double equivariant representation as Definition 4.7. Since $\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}) = \phi \circ \tau \circ \Gamma_{\text{gra}}(G)$, then $\forall (i,r,j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, $(\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}))_{\phi \circ i, \tau \circ r, \phi \circ j} = (\phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}))_{\phi \circ i, \tau \circ r, \phi \circ j} = (\Gamma_{\text{gra}}(\mathbf{A}))_{i,r,j}$ Then we can define $\Gamma_{\text{tri}} : \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} (\mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)} \times \mathbb{A}_R^{N^2}) \to \mathbb{R}^d, d \geq 1$, such that $\forall (i,r,j) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$, $\Gamma_{\text{tri}}((i,r,j),\mathbf{A}) = (\Gamma_{\text{gra}}(\mathbf{A}))_{i,r,j}$. It is clear that $\Gamma_{\text{tri}}((i,r,j),\mathbf{A}) = (\Gamma_{\text{gra}}(\mathbf{A}))_{i,r,j} = (\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}))_{\phi \circ i, \tau \circ r, \phi \circ j} = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ r, \phi \circ j), \phi \circ \tau \circ \mathbf{A})$. Thus we show Γ_{tri} is an invariant triplet representation as in Definition 4.5. $\square$

Theorem 4.10. *Given an arbitrary triplet predictor $\eta : \mathbb{R}^d \to \{0,1\}$ that takes the triplet representation $\Gamma_{tri}((i,r,j),\mathbf{A})$ in Definition 4.5 as input, $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$, and predicts if $(i,r,j) \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$ is a positive triplet, there exists a set of UQER Horn clauses (Definition 4.9) that predicts the same positive triplets for all $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ and $(i,r,j) \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$.*

Proof. We show the proof by constructing a set of Horn clauses and prove that they have the desired properties. For any KG $\mathbf{A} \in \cup_{R=1}^{\infty} \cup_{N=2}^{\infty} \mathbb{A}_R^{N^2}$ with N nodes and R relations. We first consider triplets $(i_+, r_+, j_+) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$ that predicted to be positive by the model, i.e., $\eta(\Gamma_{\text{tri}}((i_+, r_+, j_+),\mathbf{A})) = 1$. For each of such triplets, we can construct a Horn clause where (i_+, r_+, j_+) is on the right side, and all KG facts involving all relations and entities are in the left. For each Horn clause with (i_+, r_+, j_+) on the right side, we add the node and relation permutation equivariance property by defining $\mathcal{R} = \mathcal{R}^{(R)}$, $\mathcal{V} = \mathcal{V}^{(N)}$, and $\mathbf{B}_{\phi \circ i'_+, \tau \circ r'_+, \phi \circ j'_+} = 1$ iff $(i'_+, r'_+, j'_+) \in \mathcal{S}$ and $\phi \circ i_+ = 1, \tau \circ j_+ = 2$ (without considering self relations), $\tau \circ r_+ = 1$, which makes it an UQER Horn clause. Then $\forall \phi' \in \mathbb{S}_N, \forall \tau' \in \mathbb{S}_R$, $\eta(\Gamma_{\text{tri}}(\phi' \circ i_+, \tau' \circ r_+, \phi' \circ j_+), \phi' \circ \tau' \circ \mathbf{A})) = \eta(\Gamma_{\text{tri}}((i_+, r_+, j_+),\mathbf{A})) = 1$ predicts $(\phi' \circ i_+, \tau' \circ r_+, \phi' \circ j_+)$ in $\phi' \circ \tau' \circ \mathbf{A}$ as positive, where $(\phi' \circ i_+, \tau' \circ r_+, \phi' \circ j_+)$ is also still a valid implication of the UQER Horn clauses based on the permuted KG $\phi' \circ \tau' \circ \mathbf{A}$ facts, by definition of the UQER Horn clauses (Definition 4.9). Thus, for all triplets predicted to be positive for any KG, there exists a set of UQER Horn clauses that imply the triplets given the same KG facts.

We then consider triplets $(i_-, r_-, j_-) \in \mathcal{V}^{(N)} \times \mathcal{R}^{(R)} \times \mathcal{V}^{(N)}$ that predicted to be negative by the model, i.e., $\eta(\Gamma_{\text{tri}}((i_-, r_-, j_-),\mathbf{A})) = 0$. Suppose there exists a UQER Horn clause in the set of UQER Horn clauses constructed in the above paragraph that implies (i_-, r_-, j_-) in the right side, for some (i_-, r_-, j_-) such that $\eta(\Gamma_{\text{tri}}((i_-, r_-, j_-),\mathbf{A})) = 0$. Then it means $\exists \phi \in \mathbb{S}_N, \exists \tau \in \mathbb{S}_R$, such that $(\phi \circ i_-, \tau \circ r_-, \phi \circ j_-)$ is a triplet that predicted as positive from the predictor (i.e., $\eta(\Gamma_{\text{tri}}((\phi \circ i_-, \tau \circ r_-, \phi \circ j_-), \phi \circ \tau \circ \mathbf{A})) = 1$) by construction. Now we conclude that $\eta(\Gamma_{\text{tri}}((i_-, r_-, j_-),\mathbf{A})) = \eta(\Gamma_{\text{tri}}((\phi \circ i_-, \tau \circ r_-, \phi \circ j_-), \phi \circ \tau \circ \mathbf{A})) = 1$, which contradicts $\eta(\Gamma_{\text{tri}}((i_-, r_-, j_-),\mathbf{A})) = 0$. Thus, there are no UQER Horn clauses that imply these triplets for all triplets predicted to be negative (i.e., not predicted as positive). $\square$

Lemma 5.2. *The triplet representation in Equation (5) is an invariant triplet representation as per Definition 4.5.*

Proof. From our model architecture (Equation (5)), $\Gamma_{\text{ISDEA}}((i,r,j),\mathbf{A}) = (h_{i,r}^{(T)} \parallel h_{j,r}^{(T)} \parallel d(i,j) \parallel d(j,i))$,. Using DSS layers, we can guarantee the node representations $h_{i,r}^{(T)}$ we learn achive invariance under the node and relation permutations, where $h_{i,r}^{(T)}$ in $\mathbf{A}$ is equal to $h_{\phi \circ i, \tau \circ r}^{(T)}$ in $\phi \circ \tau \circ \mathbf{A}$. It is also clear that distance function is invariant to node and relation permutations, i.e. $\forall i,j \in \mathcal{V}$, $d(i,j)$ in $\mathbf{A}$ is the same as $d(\phi \circ i, \phi \circ j)$ in $\phi \circ \tau \circ \mathbf{A}$. Thus $\Gamma_{\text{ISDEA}}((i,r,j),\mathbf{A}) = \Gamma_{\text{ISDEA}}((\phi \circ i, \tau \circ r, \phi \circ j), \phi \circ \tau \circ \mathbf{A})$ is an invariant triplet representation as in Definition 4.2. $\square$

B. Related Work

Factorization-based method for KG A widely popular way to tackle KG completion tasks is through latent representations of entities and relations. The basic principle is that the embedding should capture their relative information in the KG. Traditionally, factorization-based methods (Sutskever et al., 2009; Nickel et al., 2011; Bordes et al., 2013; Wang et al., 2014; Yang et al., 2015; Trouillon et al., 2016; Nickel et al., 2016; Trouillon et al., 2017; Dettmers et al., 2018; Sun et al., 2019) have been proposed to obtain latent embedding of entities and relations. These models try to score all combinations of relations and entities with embedding as factors, similar as tensor-factorization. Although excellence in transductive tasks, it is not possible to apply on inductive tasks on unseen entities without extensive retraining.

GNN-based model for KG In recent years, with the advancement of graph neural networks (GNNs) (Defferrard et al., 2016; Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2017; Bronstein et al., 2017; Murphy et al., 2019c), in graph machine learning fields, various works has applied the idea of GNN in relational prediction to ensure the inductive capability of the model, including RGCN (Schlichtkrull et al., 2018), GraIL (Teru et al., 2020), NodePiece (Galkin et al., 2021), NBFNet (Zhu et al., 2021), ReFactorGNNs (Chen et al., 2022) etc.. These models can be used to infer on unseen entities at test time without extensive retraining as the factorization-based methods, while most of the GNN performance are worse than FM-based methods (Ruffinelli et al., 2020; Chen et al., 2022). Specifically, Teru et al. (2020) extends the idea from Zhang & Chen (2018) to use local subgraph representations for KG link prediction. Chen et al. (2022) aims to build the connection between FM and GNNs, where they propose an architecture to cast FMs as GNNs. Galkin et al. (2021) uses anchor-nodes for parameter-efficient architecture for KG completion. Zhu et al. (2021) extends the Bellman-Ford algorithm which learns pairwise representations by all the path representations betweneen nodes. Barcelo et al. (2022) tries to understand KG-GNNs expressiveness by connecting it with the Weisfeiler-Leman test in KG. Qian Huang & Leskovec (2022) aims to perform inductive reasoning over new relations, sharing the same interest as our work. However, the difference is that they frame it as a few-shot learning problem with few examples of new relations given, while we do not have access to any new relations.

Logical Induction The relation prediction problem in knowledge graph can also be considered as the problem of learning first-order logical Horn clauses (Yang et al., 2015; 2017; Sadeghian et al., 2019; Teru et al., 2020) from the knowledge graph, where one aims to extract logical rules on binary predicates. Barceló et al. (2020) discusses the connection between the expressiveness of GNNs and first-order logical induction, but only on node GNN representation and logical node classifier. In our paper, we try to build connection between triplet representation and logical Horn clauses. Traditionally, logical rules are learned through statistically enumerating patterns observed in KG (Lao & Cohen, 2010; Galárraga et al., 2013). Neural LP (Yang et al., 2017) and DRUM (Sadeghian et al., 2019) learns logical rules in an end-to-end differentiable manner using the set of logic paths between two entities with sequence models. Cheng et al. (2022) follows a similar manner which breaks a big sequential model into small atomic models in a recursive way. Galkin et al. (2022) aims to inductively extract logical rules by devising NodePiece (Galkin et al., 2021) and NBFNet (Zhu et al., 2021).

Knowledge graph alignment Knowledge graph alignment tasks (Sun et al., 2018; 2020; Yan et al., 2021) are very common in heterogeneous, cross-lingual, and domain-specific knowledge graphs, where the task aim to align entities among different domains. For example, matching entities with there counterparts in different languages (Wang et al., 2018; Xu et al., 2019b). It is intrinsically different than our task where we aim to inductively apply on completely new entities and relations, possibly with no clear alignments between them.

C. Detailed Comparison between Different Methods

In this section, we give a detailed comparison between different factorization-based methods, GNN and logical induction methods. Suppose for a focusing triplet (i, r, j), we are provided with (i, j)'s enclosed subgraph $G_{(i,j)}$ (Zhang & Chen, 2018; Teru et al., 2020) which is a subgraph contains only nodes $\mathcal{N}^{(T)}(i) \cap \mathcal{N}^{(T)}(j)$ where $\mathcal{N}^{(T)}(i)$ are all neighbors within T-hop of node i, and direct connections between i and j inside the subgraph are removed for self-supervision. T is an arbitrary number which should be the same for all methods for comparison fairness. Given enclosed subgraph $G_{(i,j)}$ as the input, difference between all considered methods for inductive KG completion is how to achieve representation of (i, r, j) from the enclosed subgraph.

For tensor factorization methods including RotatE (Sun et al., 2019), pRotatE (Sun et al., 2019), TransE (Bordes et al., 2013), ComplEx (Trouillon et al., 2016), DistMult (Yang et al., 2015), we will have two learnable embedding matrices $H^{(\mathcal{V})}$

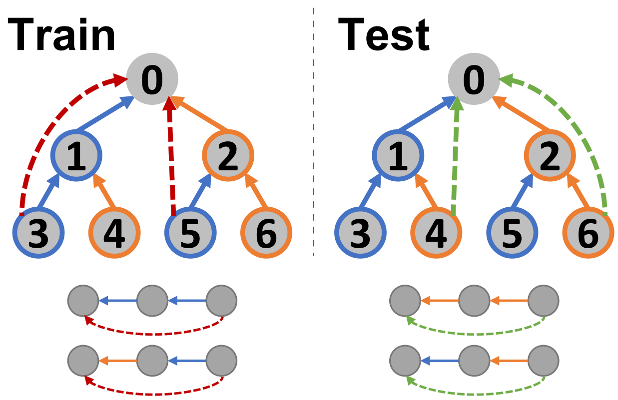


Figure 5. **FD-1 Generation Process.** Above we show generation result of depth 2 for FD-1 dataset. Below we illustrate sharing Horn clause applied in both training and test generation. A blue arrow, e.g., node 1 to node 0 means "node 1 is father of node 0"; An orange arrow, e.g., node 2 to node 0 means "node 2 is mother of node 0"; A red arrow, e.g., node 3 to node 0 means "node 3 is Grand $\wedge$ Father of node 0"; and A green arrow, e.g., node 6 to node 0 means "node 6 is Grand $\wedge$ Mother of node 0". In both training and test, direct father and mother triplet (solid arrows) are given as fact, and only dashed triplets are used for training loss or evaluation.

and $H^{(\mathcal{R})}$, the the representation of (i, r, j) is $f(H_i^{(\mathcal{V})}, H_r^{(\mathcal{R})}, H_j^{(\mathcal{V})})$ where f is a distance measurement between $H_i^{(\mathcal{V})}$ and $H_j^{(\mathcal{V})}$ given $H_r^{(\mathcal{R})}$. The difference of methods in this category is the selection of f.

For RGCN and ReFactorGNN, it is an extension of tensor factorization method with GNNs. First, it will fill $H^{(\mathcal{V})}$ and $H^{(\mathcal{R})}$ to corresponding nodes and edges in enclosed subgraph $G_{(i,j)}$ to generate an attributed enclosed subgraph $G'_{(i,j)}$, then an attributed graph neural network is applied on $G'_{(i,j)}$. We denote the final representation of node u given by GNN as $\text{GNN}\big(G'_{(i,j)}\big)_u$. Then, the representation of (i, r, j) is achieved from $f\big(\text{GNN}\big(G'_{(i,j)}\big)_i, \text{GNN}\big(G'_{(i,j)}\big)_j\big)$. RGCN is sensitive to values of $H^{(\mathcal{V})}$, while RefactorGNN is specially designed to be insensitive to that on inference.

For NodePiece, it is a variant of former category. The only difference is that the final node representation $\text{GNN}\big(G'_{(i,j)}\big)_u$ of each node u is augmented with shortest distances to several anchors node in the training graph which are selected by arbitrary strategy, e.g., uniform sampling.

For Neural LP and DRUM, they will first extract all different paths from i to j within $G_{(i,j)}$. We denote the collection of all random paths as set $\mathbb{W}_{(i,j)}$, then the representation for (i, r, j) is $\text{READOUT}(\{\text{RNN}(w) | \forall w \in \mathbb{W}_{(i,j)}\})$ where READOUT is arbitrary aggregation function, e.g., sum, and RNN is arbitrary recurrent neural network (Rumelhart et al., 1986; Hochreiter & Schmidhuber, 1997; Schuster & Paliwal, 1997). The difference between these two methods is the RNN architecture.

For GraIL and NBFNet, they will first assign node attributes for enclosed subgraph $G_{(i,j)}$ by strategy like DRNL, ZO (Zhang et al., 2021; Chamberlain et al., 2022), then fill $H^{(\mathcal{R})}$ to corresponding edge as edge attributes. This will result in an attributed enclosed subgraph $G'_{(i,j)}$. Next, an attributed graph neural network is applied on $G'_{(i,j)}$, and the representation for (i, r, j) is $\text{READOUT}\big(\text{GNN}\big(G'_{(i,j)}\big)\big)$ where READOUT is arbitrary aggregation function, e.g., average over all nodes. NBFNet has optimized on data batching for better efficiency.

For our proposed IS-DEA, the representation for (i, r, j) is $f\big(\text{DSSGNN}\big(G'_{(i,j)}\big)_i, \text{DSSGNN}\big(G_{(i,j)}\big)_j, d_{i,j}, d_{j,i}\big)$ where $d_{i,j}$ is the shortest distance from i to j in $G_{(i,j)}$ and DSS-GNN is as Equation (4).

D. Experiments

D.1. Dataset Generation

Family Diagram 1 (FD-1). The generation of FD-1 is simply based on two logic chain rules as shown in Figure 5: In training, we only have $(X, \text{Parent}, Z) \wedge (Z, \text{Father}, Y) \implies (X, \text{Grand} \wedge \text{Father}, Y)$, while in test, we only have $(X, \text{Parent}, Z) \wedge (Z, \text{Mother}, Y) \implies (X, \text{Grand} \wedge \text{Mother}, Y)$. Here (X, Parent, Z) means either (X, Father, Z) or (X, Mother, Z). For both training and test scenario, all direct father and mother relations (arcs) are provided as facts. A simplified generation process is illustrated in Figure 4. The only difference in experiment is that true dataset has a complete binary tree of depth 6 while simplified version only has depth 2.

The relation-independent Horn clauses sharing between training and test are quite simple: $\forall r_1, \forall r_2, \forall r_3, r_1 \neq r_2 \neq r_3$, there are two Horn clauses, $(X, r_1, Z) \wedge (Z, r_2, Y) \implies (X, r_3, Y)$ and $(X, r_1, Z) \wedge (Z, r_2, Y) \implies (X, r_2, Y)$. Since only father and mother appear in facts (the right side of imply symbol) in each Horn clauses, in the first Horn clause, r_3 will always be "grand".

Family Diagram 2 (FD-2). The generation of FD-2 is an extension of Figure 3 except that we break connections like $(\mathrm{y3}, \mathrm{y6})$ for the ease of generation. Detailed generation steps are as following description: First, we select the same sharing Horn clause as used in the explanation of Theorem 4.10 and Figure 3 across training and test: $\forall r_1, \forall r_2, r_1 \neq r_2, (X, r_1, Z) \wedge (Z, r_2, Y) \to (X, r_2, Y)$. Then, we generate training facts from a complete binary tree of depth 6 with only relations "a" and "b" just as we did in FD-1, and apply the only relation-independent formerly defined to collect training queries (missing positive triplets) with only relation "b". Next, we repeat the same process on another two binary trees, one of whose relations are "1" and "2", the other of whose relations are "3" and "4". Finally, we merge all facts and queries from these two binary trees together as test dataset.

Related Synthetic Dataset. Clutrr (Sinha et al., 2019) is a similar synthetic logic reasoning tasks focusing on learning family relations. However, each sample of Clutrr requires to predict a missing relation between two entities given a story of relations among multiple entities. Thus, it also requires computational linguistics techniques which is unrelated to our focus. Besides, the training and test relations of it have exactly the same semantic meanings which is also not interesting in our study.

GraphLog (Sinha et al., 2020) is another synthetic logical reasoning tasks. However, the training and test relations of it are also same which makes it less interesting in our study. Besides, its data is not available now, and we can not adapt it into our conditions.

D.2. Experiment Setup

Baselines and Implementation Details. In all experiments, we compare with four inductive and differentiable knowledge graph completion baselines, Neural LP (Yang et al., 2017), DRUM (Sadeghian et al., 2019), GraIL (Teru et al., 2020), and NBFNet (Zhu et al., 2021). Neural LP and DRUM treat KG as logical rule conjunctions, while GraIL and NBFNet treat KG as attributed graph. Same as Teru et al. (2020); Zhu et al. (2021), we run each method 5 times on each dataset, and collect mean performance whose standard deviations are omitted since they are all small. For training of each single run, we augment each triplet (i, r, j) by its inversion (i, r^{-1}, j), and sample 2 negative triplets (i', r, j') per positive in training as Sun et al. (2019); Zhu et al. (2021); For evaluation, we will not augment KG by inversions, and sample 50 negative triplets per positive to compute common metrics such as Mean Reciprocal Rank (MRR) and Hits@k as all baselines. For each positive sample, its negative samples are generated by uniformly corrupt either its subject or object by a random entity. We will filter out negative samples that collide with any positive triplets in facts and sample them again until there is no collision. Besides, we only corrupt objects in training, since corruption of subject can be achieved from corrupting object of inverse triplets (Sun et al., 2019; Zhu et al., 2021).

Hyperparameters. We follow the same configuration as Teru et al. (2020) such that hidden layer is of 32 neurons, use Adam optimizer with learning rate 0.01, and weight decay 5e-4. For all datasets, we train our model 50 epochs with batch size 256. If the model is not improving for 15 epochs, we early stop the training. For all methods, number of hops and number of layers are 2 on FD-1 and FD-2, and are 3 on real-world inductive KG completion to ensure fair comparison.

Complexity. For each layer of our method, it can be treated as running 2 homogeneous GNN $|\mathcal{R}|$ times on the KG, thus time cost is roughly $2|\mathcal{R}|$ times of adopted GNN. In our experiment, we use GIN (Xu et al., 2019a) as our GNN architecture, thus the complexity is $\mathcal{O}(|\mathcal{R}||\mathcal{S}|d^3)$ where d is the maximum size of hidden layers, $|\mathcal{R}|$ is number of relations in the KG, and $|\mathcal{S}|$ is number of fact triplets (number of edges) in KG.

Besides, our method requires the shortest distance between any two nodes without passing direct connection between two nodes for both positive and negative samples. Pay attention that this can not be simply achieved from Dijkstra or Floyd algorithm since the graph changes on computing each node pair, indeed computing such distance needs to traverse enclosed graph (Zhang & Chen, 2018; Teru et al., 2020) between each node pair once. Thus the complexity is the same as enclosed graph extraction which will be influenced by knowledge graph size and negative sampling rate. Roughly speaking, on

Dataset	Model	MRR↑		Hits@1↑		Hits@5↑		**Hits@10↑**	
		Original	Permuted	Original	Permuted	Original	Permuted	Original	Permuted
WN18RR-v1	Neural LP	0.388	0.271	0.179	0.109	0.671	0.437	0.825	0.686
	DRUM	0.395	0.330	0.201	0.157	0.645	0.527	0.823	0.745
	GraIL	0.829	0.789	0.806	0.744	0.840	0.822	0.840	0.822
	NBFNet	0.798	0.327	0.723	0.202	0.883	0.457	0.949	0.630
	IS-DEA	0.615	0.615	0.474	0.474	0.820	0.820	0.974	0.974
WN18RR-v2	Neural LP	0.397	0.299	0.194	0.124	0.651	0.498	0.839	0.772
	DRUM	0.420	0.361	0.228	0.161	0.669	0.620	0.847	0.837
	GraIL	0.816	0.810	0.808	0.796	0.816	0.816	0.816	0.816
	NBFNet	0.845	0.514	0.802	0.418	0.898	0.569	0.960	0.833
	IS-DEA	0.611	0.611	0.475	0.475	0.811	0.811	0.959	0.959
WN18RR-v3	Neural LP	0.282	0.264	0.141	0.112	0.455	0.445	0.607	0.594
	DRUM	0.307	0.288	0.169	0.152	0.469	0.450	0.611	0.604
	GraIL	0.602	0.598	0.567	0.559	0.632	0.633	0.633	0.633
	NBFNet	0.756	0.549	0.683	0.482	0.848	0.611	0.907	0.716
	IS-DEA	0.624	0.624	0.467	0.467	0.858	0.858	0.954	0.954
WN18RR-v4	Neural LP	0.387	0.242	0.198	0.088	0.620	0.394	0.752	0.628
	DRUM	0.384	0.333	0.206	0.174	0.602	0.501	0.746	0.698
	GraIL	0.760	0.741	0.748	0.708	0.763	0.763	0.763	0.763
	NBFNet	0.778	0.178	0.740	0.069	0.805	0.234	0.867	0.474
	IS-DEA	0.595	0.595	0.473	0.473	0.740	0.740	0.909	0.909

Table 5. **Results on original and relation-shuffled WN18RR dataset.**

Dataset	Model	MRR↑		Hits@1↑		Hits@5↑		**Hits@10↑**	
		Original	Permuted	Original	Permuted	Original	Permuted	Original	Permuted
FB237-v1	Neural LP	0.272	0.240	0.176	0.141	0.368	0.346	0.476	0.424
	DRUM	0.308	0.241	0.207	0.149	0.407	0.346	0.495	0.422
	GraIL	0.558	0.451	0.461	0.324	0.668	0.605	0.741	0.717
	NBFNet	0.718	0.405	0.624	0.309	0.856	0.500	0.895	0.605
	IS-DEA	0.573	0.573	0.415	0.415	0.773	0.773	0.960	0.960
FB237-v2	Neural LP	0.258	0.223	0.140	0.116	0.391	0.345	0.532	0.478
	DRUM	0.277	0.229	0.148	0.106	0.430	0.373	0.550	0.510
	GraIL	0.711	0.557	0.614	0.408	0.832	0.768	0.884	0.864
	NBFNet	0.866	0.528	0.802	0.439	0.954	0.633	0.977	0.707
	IS-DEA	0.701	0.701	0.567	0.567	0.882	0.882	0.982	0.982
FB237-v3	Neural LP	0.249	0.218	0.147	0.120	0.379	0.328	0.491	0.443
	DRUM	0.261	0.182	0.145	0.084	0.395	0.297	0.501	0.407
	GraIL	TL	TL	TL	TL	TL	TL	0.444	TL
	NBFNet	0.855	0.531	0.787	0.417	0.951	0.675	0.980	0.741
	IS-DEA	0.776	0.776	0.674	0.674	0.910	0.910	0.984	0.984
FB237-v4	Neural LP	0.222	0.177	0.133	0.093	0.320	0.260	0.445	0.372
	DRUM	0.216	0.166	0.111	0.073	0.340	0.263	0.444	0.375
	GraIL	TL	TL	TL	TL	TL	TL	0.444	TL
	NBFNet	0.873	0.524	0.810	0.413	0.961	0.664	0.987	0.725
	IS-DEA	0.793	0.793	0.696	0.696	0.926	0.926	0.981	0.981

Table 6. **Results on original and relation-shuffled FB237 dataset.**

popular transductive and inductive knowledge graph completion baselines, it takes days to months for extracting such information of a single run as a preprocessing step.

Dataset	Model	MRR↑		Hits@1↑		Hits@5↑		**Hits@10**↑	
		Original	Permuted	Original	Permuted	Original	Permuted	Original	Permuted
NELL995-v1	Neural LP	0.472	0.455	0.442	0.405	0.500	0.500	0.500	0.500
	DRUM	0.503	0.488	0.500	0.470	0.500	0.500	0.500	0.500
	GraIL	0.825	0.578	0.737	0.476	0.931	0.706	0.932	0.734
	NBFNet	0.872	0.868	0.775	0.755	0.985	0.985	0.995	0.995
	IS-DEA	0.925	0.925	0.882	0.882	0.975	0.975	0.975	0.975
NELL995-v2	Neural LP	0.237	0.222	0.144	0.139	0.310	0.291	0.410	0.384
	DRUM	0.286	0.248	0.170	0.152	0.397	0.326	0.528	0.425
	GraIL	0.785	0.700	0.686	0.599	0.914	0.834	0.963	0.899
	NBFNet	0.836	0.803	0.751	0.698	0.948	0.942	0.978	0.978
	IS-DEA	0.784	0.784	0.687	0.687	0.899	0.899	0.960	0.960
NELL995-v3	Neural LP	0.244	0.232	0.154	0.147	0.320	0.299	0.434	0.397
	DRUM	0.308	0.274	0.223	0.190	0.387	0.347	0.476	0.441
	GraIL	0.766	0.623	0.651	0.466	0.929	0.860	0.958	0.947
	NBFNet	0.865	0.862	0.789	0.788	0.961	0.958	0.969	0.968
	IS-DEA	0.778	0.778	0.690	0.690	0.895	0.895	0.957	0.957
NELL995-v4	Neural LP	0.193	0.183	0.106	0.100	0.258	0.237	0.356	0.340
	DRUM	0.211	0.195	0.124	0.110	0.282	0.254	0.367	0.352
	GraIL	0.477	0.396	0.312	0.239	0.692	0.575	0.880	0.806
	NBFNet	0.802	0.770	0.711	0.672	0.915	0.890	0.957	0.937
	IS-DEA	0.704	0.704	0.582	0.582	0.840	0.840	0.906	0.906

Table 7. **Results on original and relation-shuffled NELL995 dataset.**

NBFNet Configuration	NELL995 Dataset			
	v1	v2	v3	V4
Original	0.995	0.978	0.969	0.957
With Constant Relation Embeddings	0.740	0.949	0.928	0.877

Table 8. **Hits@10 Performance without/with Constant Relation Embeddings.** Forcing NBFNet relation embeddings to be the same constant still perform well on some NELL995 datasets. This shows that some knowledge graph completion tasks can be approximated by link predictions.

D.3. More Result Explanation

FD-1. Since FD-1 comes from an extremely simple generation process, we would expect our methods to achieve perfect performance on it (always rank positive triplets at rank 1 against all corresponding negative samples). However, it seems like that IS-DEA fails to achieve perfect performance (MRR and Hit@k all being 1.0) on this simple task. Indeed, there is no way to achieve such perfect performance on MRR, Hit@1 and Hit@2. The issue is that in the querying relations, for either father or mother relation prediction, there will be two equally good choices, e.g., $(0, \text{Father}, X), X \in \{3, 5\}$; for grand relation prediction, there will be four equally good choices, e.g., $(0, \text{Grand}, X), X \in \{3, 4, 5, 6\}$. But if we see Hit@4, IS-DEA achieves 100% accuracy.

Another minor observation is that Neural LP and DRUM has exactly the same performance on FD-1. The reason is that Neural LP and DRUM has exactly the same framework except that the neural network architecture are slightly different. This observation is also found in (Teru et al., 2020; Zhu et al., 2021), and this also happens in later real-world experiments.

Real-world Datasets. One interesting observation is that besides our proposal, some baselines are also insensitive to relation shuffling on some datasets, e.g., NBFNet on NELL995. We suppose the reason is that inductive NELL995 knowledge graph completion tasks can be simply reduced to link prediction tasks where relation ID has barely no influence. To verify this guess, we force NBFNet relation embeddings to be all-one (forcing all relations to be the same), and run the experiments again, we can see that the performance is nearly the same as original on NELL-v2 and NELL-v3, and is still fine on NELL-v1 and NELL-v4 in Table 8 which reflects that knowledge graph completion on NELL995 is nearly equivalent to link prediction.

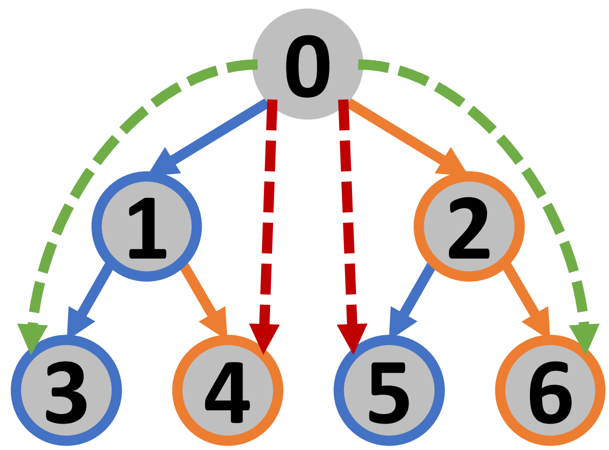


Figure 6. **Counterexample of IS-DEA limitation due to use of node embeddings rather than pairwise embeddings.** Blue means "Father"; orange means "Mother"; red means "Grand_diff" that we have two different relations on the chain path; green means "Grand_same" that we have two same relations on the chain path. Queries are all four "Grand_diff" and "Grand_same" triplets.

Ablation Study. Since negative samplings are drawn by uniformly corrupting object (without loss of generality), it is very likely that corrupted objects are far way from subject while true object is close to subject. Under such scenario, shortest distance itself will be a powerful enough feature to achieve good ranking performance in knowledge graph completion, thus we want to know if shortest distance feature augmentation contributes to the performance gain. We perform ablation study on two dataset NELL995-V2 and WN18RR-v4. As shown in Table 4, even if shortest distance is excluded from our model, it still performs quite well.Thus, we can say that double-equivariant representation itself is enough to provide good performance. Besides, we also show in Table 4 that shortest distance itself is not enough for knowledge graph completion.

D.4. Expressivity Limitation of Doubly Exchangeable Representation

In Figure 6, we denote all four relations by numbers such that "father" (blue) is 0, "mother" (orange) is 1, "grand_diff" (red) is 2 and "grand_same" (green) is 3. We are going to show that IS-DEA is incapable to distinguish triplets of relation 2 and triplets of relation 3 in Figure 6.

We denote the node representation given by IS-DEA as $H_{v,r}$ where $v \in [0, 6]$ and $r \in [0, 3]$.

Given only the fact tripelts (relation 0 and 1, or color blue and orange), we can easily see that node 3 and node 6 are symmetric, and node 4 and node 5 are symmetric by simply flip the father and mother relation IDs. Thus, based on the invariance as Definition 4.5, we should get

$$\begin{aligned} H_{3,0} &= H_{6,1} \\ H_{3,1} &= H_{6,0} \\ H_{4,0} &= H_{5,1} \\ H_{4,1} &= H_{5,0} \\ H_{3,2} = H_{3,3} &= H_{6,3} = H_{6,2} \\ H_{4,2} = H_{4,3} &= H_{5,3} = H_{5,2} \\ H_{0,2} = H_{0,3} &= H_{0,3} = H_{0,2} \end{aligned} \tag{7}$$

The representation of "grand_diff" and "grand_same" on each node is always the same because there is no facts involving these two relations, thus IS-DEA can not see their difference, thus can not distinguish them on representations. From the computation view as Equation (4), the first function $L_1^{(k)}$ always receive an empty graph, while the second function $L_2^{(k)}$ always receive the full unattributed graph (only facts) for relation "grand_diff" and "grand_same".

The representation of four triplets to be queried will be

$$
\begin{aligned}
\Gamma_{\text{tri}}((0,3,3),\mathbf{A}) &= H_{0,3} \parallel H_{3,3} \\
\Gamma_{\text{tri}}((0,4,2),\mathbf{A}) &= H_{0,2} \parallel H_{4,2} \\
\Gamma_{\text{tri}}((0,5,2),\mathbf{A}) &= H_{0,2} \parallel H_{5,2} \\
\Gamma_{\text{tri}}((0,6,3),\mathbf{A}) &= H_{0,3} \parallel H_{6,3}
\end{aligned}
$$

We omit the shortest distances in the representation since they are all 2, thus has no influence when compare with each other. Based on Equation (7), we can further notice that

$$
\begin{aligned}
\Gamma_{\text{tri}}((0,3,3),\mathbf{A}) &= H_{0,3} \parallel H_{3,3} = H_{0,3} \parallel H_{6,3} = H_{0,2} \parallel H_{6,2} = H_{0,2} \parallel H_{3,2} \\
&= \Gamma_{\text{tri}}((0,6,3),\mathbf{A}) \\
&= \Gamma_{\text{tri}}((0,6,2),\mathbf{A}) \\
&= \Gamma_{\text{tri}}((0,3,2),\mathbf{A}) \\
\Gamma_{\text{tri}}((0,4,2),\mathbf{A}) &= H_{0,2} \parallel H_{4,2} = H_{0,2} \parallel H_{5,2} = H_{0,3} \parallel H_{5,3} = H_{0,3} \parallel H_{4,3} \\
&= \Gamma_{\text{tri}}((0,5,2),\mathbf{A}) \\
&= \Gamma_{\text{tri}}((0,5,2),\mathbf{A}) \\
&= \Gamma_{\text{tri}}((0,4,3),\mathbf{A})
\end{aligned}
$$

Suppose the final MLP translating triplet representations into scores is f, and denote score $s_{i,r,j} = f(\Gamma_{\text{tri}}(i,r,j),\mathbf{A})$, we will have

$$
\begin{aligned}
s_{0,3,3} &= s_{0,6,3} = s_{0,6,2} = s_{0,3,2} \\
s_{0,4,3} &= s_{0,5,3} = s_{0,5,2} = s_{0,4,2}
\end{aligned}
$$

Suppose our model can perform well on "Grand_diff" (relation 2) completion, then it must ensure that negative cases has lower score than positive cases such that

$$
s_{0,3,2} = s_{0,6,2} < s_{0,4,2} = s_{0,5,2}.
$$

However, this also implies

$$
s_{0,3,3} = s_{0,6,3} < s_{0,4,3} = s_{0,5,3},
$$

which shows that this model is performing poorly on "Grand_same" (relation 3) completion, since it ranks node 4 and node 5 which is negative cases higher than node 3 and node 6 which is positive cases on "Grand_same".

We can see that in a case like Figure 6, if IS-DEA performs perfect for one querying relation, it must perform poorly for the other relation, thus there is no way for IS-DEA to achieve perfect performance on such tasks which reflects its expressivity limitation. However, if we only want to perform transductive learning on such cases, a tensor factorization based can easily solve this task, thus this experssivity limitation can results in a failure for knowledge graph completion.

E. Future Work

As addressed in the main paper, our implemented architecture (IS-DEA) has several limitations, which could be addressed in future work. First, IS-DEA has high pre-processing cost. This high time cost is introduced by using non-trivial shortest distance whose extraction is of the same complexity as enclosed subgraph. However, we show that non-trivial shortest distance is not fatal to our model in real-world tasks, thus it is possible that non-trivial shortest distance can be replaced by other heuristics that can be efficiently extracted.

Second, IS-DEA has high training and inference costs, since it relies on repeating GNNs for each relation. Thus, complexity IS-DEA of scales linearly w.r.t. number of relations, which is often a large number in real-world knowledge base, e.g., Wikipedia. However, fully equivariance over all relations can be too strong, and we may only want partial equivariance (Definition 4.5, Quotient Group) which may reduce the cost.

Third, IS-DEA has expressivity limitation. This limitation is related to former two cost issues since it is caused by compromising most-expressive pairwise representation to node-wise representation due to time cost. Thus if we can reduce the cost, we may be able to use more expressive graph encoder.

Finally, although we show IS-DEA representations can be explained by UQER Horn clauses, there is no algorithm to create UQER Horn clauses from IS-DEA representations. This topic is known as "explainability" which is important in knowledge graph community. We leave such an algorithm as another future work other than optimization.