Double Equivariance for Inductive Link Prediction for Both New Nodes and New Relation Types

Anonymous Author(s)
Affiliation
Address
email

Abstract

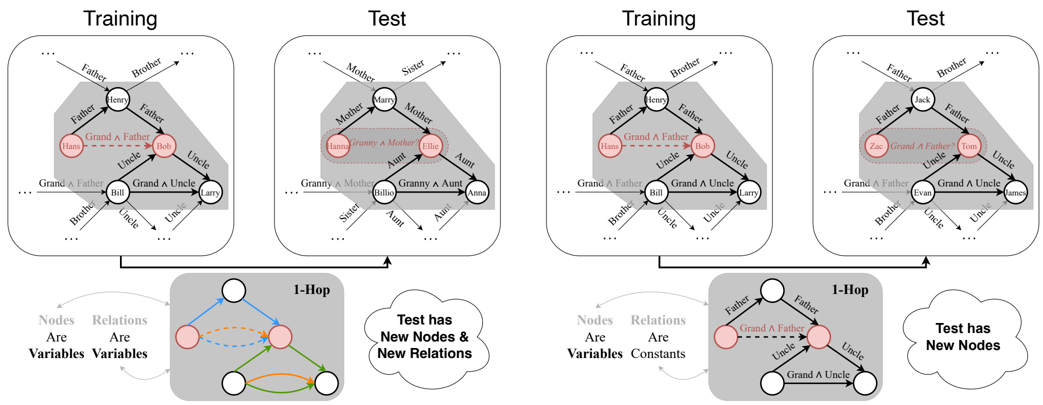
The task of inductive link prediction in discrete attributed multigraphs (e.g., knowledge graphs, multilayer networks, heterogeneous networks, etc.) generally focuses on test predictions with solely new nodes but not both new nodes and new relation types. In this work, we formally define the task of predicting (completely) new nodes and new relation types in test as a *doubly inductive link prediction* task and introduce a theoretical framework for the solution. We start by defining the concept of *double permutation-equivariant representations* that are equivariant to permutations of both node identities and edge relation types. We then propose a general blueprint to design neural architectures that impose a structural representation of relations that can inductively generalize from training nodes and relations to arbitrarily new test nodes and relations without the need for adaptation, side information, or retraining. We also introduce the concept of *distributionally double equivariant positional embeddings* designed to perform the same task. Finally, we empirically demonstrate the capability of the two proposed models on a set of novel real-world benchmarks, showcasing relative performance gains of up to 41.40% on predicting new relations types compared to baselines.

1 Introduction

This work studies what we call a *doubly inductive* (node and relation) *link prediction* task to predict missing links in unseen discrete attributed multigraphs with completely new nodes and new relation types in test (i.e. none of them are seen in training). Discrete attributed multigraphs encompass knowledge graphs [5, 76, 63, 62, 19, 58], multilayer networks (multiple graph types sharing a common set of nodes, e.g., the power grid and the road network [16, 17]), and heterogenous networks with discrete link types (e.g., recommending products to users that have distinct ways to interact with an online store [9, 79, 71]). Our experiments primarily center around knowledge graphs; however, we note that the outlined tasks and methodology can be seamlessly adapted to both multilayer and heterogeneous network data.

The main contribution of our work is a general theoretical framework for **doubly inductive link prediction** *on discrete attributed multigraphs and a blueprint to create equivariant neural networks for this task (both from structural representations and from positional embeddings).* We will introduce the concept of double equivariant graph models and distributionally equivariant positional graph embedding models, which are equivariant to the overgroup of permutations of nodes and permutations of relations (we review the necessary group theory concepts in Section 2). The majority of today's link prediction works can be broadly divided into a few categories that are either incapable of inductive reasoning over new relations in test or require side information to do so. In Section 4 we explain how the doubly inductive link prediction is different from these existing tasks in more detail.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.



(a) **Over New Nodes and New Relation Types** (b) **Over New Nodes**

Figure 1: (a) **Doubly inductive link prediction:** In this task, the goal is to learn (on training graphs) to inductively predict querying relation over test graphs with new nodes and new relation types. Sharing local **relational structure** (bottom) enables predicting the same relative relation types w.r.t. the structure of the training pair, as there is a common relational type structure (the colored links) in training that can be applied to the new nodes and new relation in test. (b) **Traditional inductive link prediction:** This task aims to inductively predict querying relation over test graph with only new nodes. Querying node pairs share the same local structure (bottom) as a training pair. Thus, we can predict the same relation type as in training. Since relation types of the local structure are assumed the same in training and test, this approach can only be applied over new nodes.

Contributions. In this work, we study the task of *doubly inductive link prediction over both new nodes and new relation types, using only information obtained from the training graphs*. Our work makes the following *three* contributions:

1. We formally introduce the *doubly inductive link prediction task* and the concept of *double equivariance* and *distributionally double equivariant positional embeddings* for graph models, whose node and pairwise representations are equivariant to the action of the permutation overgroup composed by the permutation subgroups of node identities, and edge types (relations).
2. We develop the first general *double equivariant* graph neural network (GNN) framework that is capable of performing *doubly inductive link prediction*, and introduce an approximately double equivariant representation built from distributionally double equivariant positional embeddings.
3. We introduce two real-world benchmark datasets: PediaTypes and WikiTopics, for the newly proposed doubly inductive link prediction task, and empirically verify inductive link prediction capabilities of our models over both new nodes and new relation types on these benchmarks.

2 Doubly (Node & Relation) Inductive Link Prediction

In what follows, we introduce the doubly inductive link prediction task and compare it with the traditional inductive link prediction task using two examples. We then proceed to theoretically describe the task in a general setting and propose our double equivariant modeling framework to handle doubly inductive link prediction task using structural representations and positional embeddings.

2.1 Doubly inductive link prediction examples

We now introduce doubly inductive link prediction over both new nodes and new relation types and explain the difference between the traditional inductive link prediction task in Figure 1. The traditional inductive link prediction task focuses solely on predicting new nodes in the test. To this end, standard graph neural networks (GNNs) [73, 41] force the neural network to learn structural node representations [54], which —if used appropriately— allows GNNs to perform *inductive link prediction over new nodes* [30, 60] as shown in Figure 1(b) but does not extrapolate over new relations.

However, the equivariance in GNNs is not enough to perform the doubly inductive link prediction task in Figure 1(a). Specifically, to be able to *inductively* predict the Granny $\wedge$ Mother relation on the test graph by learning from the Grand $\wedge$ Father relation on the training graph, the equivariance property needs to go beyond just node permutations. To be able to represent the structural properties of the nodes and relations with respect to the structural properties of other nodes and relations, our work defines an equivariance also in relations. For instance, via double equivariance (we will define the concept in Section 2.3) it is possible to perform the task of predicting Granny $\wedge$ Mother using the node and relation structural pattern shown at the bottom of Figure 1(a), which can be formally described through the logical formula $\forall v_1, v_2, v_3, v_4, v_5 \in \mathcal{V}^{\text{te}}, \forall r_1, r_2, r_3 \in \mathcal{R}^{\text{te}}, (v_1, r_1, v_2) \wedge (v_2, r_1, v_3) \wedge (v_4, r_2, v_3) \wedge (v_3, r_2, v_5) \wedge (v_4, r_2, v_5) \wedge (v_4, r_3, v_5) \implies (v_1, r_1, v_3) \wedge (v_1, r_3, v_3)$, where $\mathcal{V}^{\text{te}}$ and $\mathcal{R}^{\text{te}}$ are the (new) vertices and (new) relations observed in test. Additional examples and a more detailed analysis of the logical statements implied by double equivariance are in Appendices A and B.

2.2 Formalizing the doubly inductive link prediction task

We now introduce notations and definitions used throughout this paper. First, we formally define our inductive link prediction task for both new nodes and new relation types, i.e., doubly inductive link prediction, over discrete attributed multigraph. We denote $[n] := \{1, \ldots, n\}$ for any $n \in \mathbb{N}$. Let $\mathcal{G}^{(\text{tr})} = (\mathcal{V}^{(\text{tr})}, \mathcal{R}^{(\text{tr})}, \mathbf{A}^{(\text{tr})})$ be the training discrete attributed multigraph, where $\mathcal{V}^{(\text{tr})}$ is the set of $N^{(\text{tr})}$ training nodes, $\mathcal{R}^{(\text{tr})}$ is the set of $R^{(\text{tr})}$ training relation types. We also define two associated bijective mappings $v_\cdot^{(\text{tr})} : [N^{(\text{tr})}] \to \mathcal{V}^{(\text{tr})}, r_\cdot^{(\text{tr})} : [R^{(\text{tr})}] \to \mathcal{R}^{(\text{tr})}$ that enumerate the nodes and relation types in training. The tensor $\mathbf{A}^{(\text{tr})} \in \{0,1\}^{N^{(\text{tr})} \times R^{(\text{tr})} \times N^{(\text{tr})}}$ defines the adjacency of the training graph such that $\forall (i,k,j) \in [N^{(\text{tr})}] \times [R^{(\text{tr})}] \times [N^{(\text{tr})}], \mathbf{A}^{(\text{tr})}_{i,k,j} = 1$ indicates that the triplet $(v_i^{(\text{tr})}, r_k^{(\text{tr})}, v_j^{(\text{tr})})$ is present in the data (we denote (i,k,i) as the k-th attribute of node i). To simplify notation, we further refer to the collection of all discrete attributed multigraph of any sizes as $\mathbb{A} \coloneqq \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \{0,1\}^{N \times R \times N}$.

Definition 2.1 (Doubly inductive link prediction task)**.** The task of doubly inductive link prediction learns a model on $\mathcal{G}^{(\text{tr})}$ and inductively apply it to predict missing links in a test graph $\mathcal{G}^{(\text{te})} = (\mathcal{V}^{(\text{te})}, \mathcal{R}^{(\text{te})}, \mathbf{A}^{(\text{te})})$ with *completely* new nodes and new relation types, i.e., $\mathcal{V}^{(\text{te})} \cap \mathcal{V}^{(\text{tr})} = \emptyset, \mathcal{R}^{(\text{te})} \cap \mathcal{R}^{(\text{tr})} = \emptyset$, without extra context given to the model. Specifically, we aim to predict both missing relations for the given head and tail nodes $(i, ?, j)$ and missing nodes for a given relation $(i, k, ?)$.

While some real-world tasks may have overlapping relation types between training and test, Definition 2.1 forces the model to not rely on potential overlaps. In what follows, we use the superscript $(*)$ as a wildcard to describe both train and test data. For example, $\mathbf{A}^{(*)}$ is a wildcard variable for referring to either $\mathbf{A}^{(\text{tr})}$ or $\mathbf{A}^{(\text{te})}$. For brevity, we use discrete attributed multigraph and attributed multigraph interchangeably. And since there are bijections $v_\cdot^{(*)}, r_\cdot^{(*)}$ between indices and nodes and relation types, we represent the triplet $(v_i^{(*)}, r_k^{(*)}, v_j^{(*)}) \in \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)}$ with indices $(i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, and mainly use $\mathbf{A}^{(*)}$ to denote discrete attributed multigraph.

Without additional context such as textural description embeddings for the new relations or graph ontology (thoroughly discussed in Section 4), it is essential for our model to differentiate nodes and relations based only on their structural relationships in $\mathbf{A}^{(*)}$, rather than their labels in $\mathcal{V}^{(*)}, \mathcal{R}^{(*)}$, in order to make accurate predictions in doubly inductive link prediction as discussed in Section 2.1. Thus, we develop the double equivariant representations for attributed multigraphs as follows.

2.3 Double equivariant representations for attributed multigraphs

In what follows, we provide definitions and theoretical statements of our proposed double equivariant attributed multigraph representations in the main paper while referring all proofs to Appendix C. The proposal starts with defining the permutation actions on discrete attributed multigraphs as:

Definition 2.2 (Node and relation permutation actions on attributed multigraphs)**.** For any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, a node permutation $\phi \in \mathbb{S}_{N^{(*)}}$ is an element of the symmetric group $\mathbb{S}_{N^{(*)}}$, a relation permutation $\tau \in \mathbb{S}_{R^{(*)}}$ is an element of the symmetric group $\mathbb{S}_{R^{(*)}}$, and the operation $\phi \circ \tau \circ \mathbf{A}^{(*)}$ is the action of ϕ and τ on $\mathbf{A}^{(*)}$, defined as $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}], (\phi \circ \tau \circ \mathbf{A}^{(*)})_{\phi \circ i, \tau \circ k, \phi \circ j} = \mathbf{A}^{(*)}_{i,k,j}$ where $\phi \circ i = \phi_i$ and $\tau \circ k = \tau_k$. The node and relation permutation actions on $\mathbf{A}^{(*)}$ are commutative, i.e., $\phi \circ \tau \circ \mathbf{A}^{(*)} = \tau \circ \phi \circ \mathbf{A}^{(*)}$.

To learn structural representation for both nodes and relations, we first design triplet representations that are invariant to the two permutation actions on nodes and relations, as shown below.

Definition 2.3 (Double invariant triplet representations)**.** For any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, a double invariant triplet representation is a function $\Gamma_{\text{tri}} : \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} ([N] \times [R] \times [N]) \times \mathbb{A} \to \mathbb{R}^d, d \geq 1$, such that $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}], \forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}, \Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(*)}) = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)})$.

To understand the property of our double invariant triplet representations, we first introduce the notion of discrete attributed multigraph isomorphism and triplet double isomorphism.

Definition 2.4 (Attributed multigraph isomorphism and Triplet isomorphism)**.** We say two attributed multigraphs $\mathbf{A}^{(G)}, \mathbf{A}^{(H)} \in \mathbb{A}$ with number of nodes and relations $N^{(G)}, R^{(G)}$ and $N^{(H)}, R^{(H)}$ respectively, are isomorphic (denoted as "$\mathbf{A}^{(G)} \simeq_{\text{RL}} \mathbf{A}^{(H)}$") if and only if $\exists \phi \in \mathbb{S}_{N^{(G)}}, \exists \tau \in \mathbb{S}_{R^{(G)}}$, such that $\phi \circ \tau \circ \mathbf{A}^{(G)} = \mathbf{A}^{(H)}$. And we say two triplets $\left(i^{(G)}, k^{(G)}, j^{(G)}\right) \in [N^{(G)}] \times [R^{(G)}] \times [N^{(G)}]$, $\left(i^{(H)}, k^{(H)}, j^{(H)}\right) \in [N^{(H)}] \times [R^{(H)}] \times [N^{(H)}]$ are isomorphic triplets (denoted as "$\left(\left(i^{(G)}, k^{(G)}, j^{(G)}\right), \mathbf{A}^{(G)}\right) \simeq_{\text{TRI}} \left(\left(i^{(H)}, k^{(H)}, j^{(H)}\right), \mathbf{A}^{(H)}\right)$") if and only if $\exists \phi \in \mathbb{S}_{N^{(G)}}, \exists \tau \in \mathbb{S}_{R^{(G)}}$, such that $\phi \circ \tau \circ \mathbf{A}^{(G)} = \mathbf{A}^{(H)}$ and $\left(i^{(H)}, k^{(H)}, j^{(H)}\right) = \left(\phi \circ i^{(G)}, \tau \circ k^{(G)}, \phi \circ j^{(G)}\right)$.

For example, in Figure 1(a), (Hans, Grand$\wedge$Father, Bob) in train and (Hanna, Granny$\wedge$Mother, Ellie) in test are isomorphic triplets by Definition 2.4 (where "Granny" can be any arbitrary typo in the data). It is clear that our double invariant triplet representations are able to output the same representations for these isomorphic triplets, enabling doubly inductive link prediction. The connection between Definition 2.3 and logical reasoning can be found in Appendix B. In what follows, we define the structure double equivariant representations for the whole attributed multigraph $\mathbf{A}^{(*)}$ (akin to how GNNs provide representations for a whole graph).

Definition 2.5 (Double equivariant attributed multigraph representations)**.** For any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, a function $\Gamma_{\text{gra}} : \mathbb{A} \to \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}, d \geq 1$ 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 and relations.

Finally, we connect Definitions 2.3 and 2.5 by showing how to build double equivariant graph representations from double invariant triplet representations in Theorem 2.6, and vice-versa.

Theorem 2.6. *For all $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, given a double invariant triplet representation Γ_{tri}, we can construct a double equivariant graph representation as $\left(\Gamma_{gra}(\mathbf{A}^{(*)})\right)_{i,k,j} := \Gamma_{tri}((i,k,j), \mathbf{A}^{(*)})$, $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, and vice-versa.*

Next, we consider positional graph embeddings that are equivariant in distribution.

2.4 Distributionally double equivariant positional graph embeddings

To the best of our knowledge, InGram [35] is the first and only existing work capable of performing our doubly inductive link prediction task (Definition 2.1), but it does so with what we now define as *distributionally double equivariant positional embeddings*, which are permutation sensitive, as we will show in Section 3.2:

Definition 2.7 (Distributionally double equivariant positional embeddings)**.** For any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, the distributionally double equivariant positional embeddings of $\mathbf{A}^{(*)}$ are defined as joint samples of random variables $\mathbf{Z}|\mathbf{A}^{(*)} \sim p(\mathbf{Z}|\mathbf{A}^{(*)})$, where the tensor $\mathbf{Z}$ is defined as $\mathbf{Z}_{i,k,j} \in \mathbb{R}^d, d \geq 1, \forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, where we say $p(\mathbf{Z}|\mathbf{A}^{(*)})$ is a double equivariant probability distribution on $\mathbf{A}^{(*)}$ defined as $\forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}, p(\mathbf{Z}|\mathbf{A}^{(*)}) = p(\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)})$.

Prior work on (standard) link prediction tasks has shown the advantages of equivariant representations over positional embeddings [84]. Moreover, Srinivasan & Ribeiro (2020) [54] establishes the equivalence between positional embeddings and structural representations for simple graphs by proving that representations based on an expectation of the positional embeddings are equivariant to node permutations. In what follows, we extend this result to the double equivariant setting:

Theorem 2.8 (From distributional double equivariant positional embeddings to double equivariant representations)**.** *For any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$, the average $\mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}]$ is a double equivariant attributed multigraph representation (Definition 2.5) for any distributional double equivariant positional embeddings $\mathbf{Z}|\mathbf{A}^{(*)}$ (Definition 2.7).*

Later in Section 3.2, we use the result in Theorem 2.8 to introduce DEq-InGram, a double equivariant representation that builds upon InGram's distributionally double equivariant positional embeddings (Definition 2.7) that is shown to significantly outperforms the original InGram in Section 5.

3 Double Equivariant Neural Architecture

This section introduces two double equivariant neural architectures based on Sections 2.3 and 2.4. First, Section 3.1 introduces an Inductive Structural Double Equivariant Architecture (ISDEA), a model guaranteed to produce double equivariant representations (Definition 2.5). Then, Section 3.2 introduces a Monte Carlo estimate of a double equivariant representation built from a distributionally double equivariant positional graph embedding [35].

3.1 Inductive Structural Double Equivariant Architecture (ISDEA)

We start revisiting Definition 2.4. Consider an arbitrary discrete attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, and denote $A^{(*,k)}$ as the adjacency matrix such that $A^{(*,k)}_{i,j} := \mathbf{A}^{(*)}_{i,k,j}, \forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$. For each adjacency matrix $A^{(*,k)}$, it will correspond to a graph without edge relation types, thus we can also consider $A^{(*,k)}$ as an unattributed graph containing only edges with relation type $r^{(*)}_k$. Then, the attributed multigraph $\mathbf{A}^{(*)}$ can be equivalently expressed as a collection of unattributed graphs $A^{(*)} := \{\!\{ A^{(*,1)}, \dots, A^{(*,R^{(*)})} \}\!\}$. Since the actions of the two permutation groups $\mathbb{S}_{N^{(*)}}$ and $\mathbb{S}_{R^{(*)}}$ commute, the double equivariance of $\mathbf{A}^{(*)}$ (Definition 2.4) can be described as two (single) equivariances: A (graph) equivariance $\phi \in \mathbb{S}_{N^{(*)}}$ over each graph $A^{(*,k)}, k = 1, \dots, R^{(*)}$, and a (set) equivariance $\tau \in \mathbb{S}_{R^{(*)}}$ (over the set of graphs). Hence, our double equivariance can make use of the general framework using DSS layers on learning sets of symmetric elements proposed by Maron et al. (2020) [39]. We first define a double equivariant layer composed by a Siamese layer [7] as follows, $L : \mathbb{A} \to \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \mathbb{R}^{N\times R\times N\times d}$, for each $k = 1, ..., R^{(*)}$:

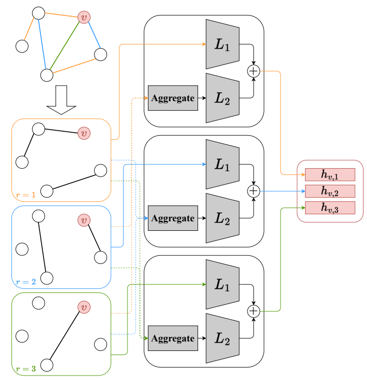


Figure 2: **Illustration of Equation (1):** Attributed multigraph input is split into a set of unattributed graphs $A^{(*,k)}$ corresponding to each relation $k = 1, \dots, R^{(*)}$. For each relation k, the representation of $A^{(*,k)}$ and the set representation of all other unattributed graphs are combined together to update representation $h_{i,k}$ for arbitrary node i. Finally, updated representations of all relations are concatenated together to generate the output h_i.

$$\left(L\left(\mathbf{A}^{(*)}\right)\right)_{:,k} = L_1\left(A^{(*,k)}\right) + L_2\left(\mathrm{AGG}^{R^{(*)}}_{k'\neq k}\left(A^{(*,k')}\right)\right), \tag{1}$$

where d is the output dimension, $L_1, L_2 : \mathbb{A} \to \cup_{N=1}^{\infty}\mathbb{R}^{N\times N\times d}$ can be any (node) equivariant layers that output pairwise representations [83, 87, 84], and the aggregation term $\mathrm{AGG}^{R^{(*)}}_{k'\neq k}$ can be any set aggregators such as sum, mean, max, DeepSets [80], etc.. Note that the proposed layer is similar to the H-equivariant layer proposed by Bevilacqua et al. (2021) [4] for increasing the expressiveness of GNN using sets of subgraphs (a markedly different task than ours). An illustration of Equation (1) is provided in Figure 2. We create our double equivariant neural network by stacking double equivariant layers.

3.1.1 Implementation Details

We use GNN layers for constructing L_1, L_2. Since most-expressive pairwise representations are computationally expensive, we propose Inductive Structural Double Equivariant Architecture (ISDEA) and trade-off expressivity in the implementation of Equation (1) for speed and memory by using node representation GNN layers [73, 65, 41]. Specifically, for an attributed multigraph $\mathbf{A}^{(*)}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, at each iteration $t = 1, ..., T$, all nodes $i \in [N^{(*)}]$ are associated with a learned vector $h_i^{(t)} \in \mathbb{R}^{R^{(*)}\times d_t}, d_t \geq 1$. If there are no node attributes, we initialize $h_i^{(0)} = \mathbb{1}$ and $d_0 = 1$. Then we recursively compute the update, $\forall i \in [N^{(*)}], \forall k \in [R^{(*)}]$,

$$h^{(t+1)}_{i,k} = \mathrm{GNN}^{(t)}_1\left(h^{(t)}_{i,k}, \{\!\{ h^{(t)}_{j,k} \,\middle|\, j\in\mathcal{N}_k(i) \}\!\}\right) + \mathrm{GNN}^{(t)}_2\left(\mathrm{AGG}^{R^{(*)}}_{k'\neq k}\left(h^{(t)}_{i,k'}\right), \left\{\!\!\left\{ \mathrm{AGG}^{R^{(*)}}_{k'\neq k}\left(h^{(t)}_{j,k'}\right) \,\middle|\, j\in \bigcup_{k'\neq k}\mathcal{N}_{k'}(i) \right\}\!\!\right\}\right),$$

where $\text{GNN}_1^{(t)}$ and $\text{GNN}_2^{(t)}$ denote two GNN layers and $\mathcal{N}_k(i) := \left\{j \middle| \mathbf{A}^{(*)}_{j,k,i} = 1\right\}$ denotes the neighborhood set of node i with relation k in the unattributed graph $A^{(*,k)}$. At the final layers, we use standard MLPs instead of GNNs to output a final prediction. We use *mean* as our aggregators.

As shown by Srinivasan & Ribeiro (2020) [54] and You et al. (2019) [78], structural node representations are not most expressive for link prediction in unattributed graphs. Hence, we concatenate i and j (double equivariant) node representations with the shortest distance between i and j in the observed graph as our triplet representations (appending distances is also adopted in the representations of prior work [60, 22]). Finally, we obtain the triplet representation,

$$\Gamma_{\text{ISDEA}}((i,k,j), \mathbf{A}^{(*)}) = \left(h_{i,k}^{(T)} \middle\| h_{j,k}^{(T)} \middle\| d(i,j) \middle\| d(j,i)\right), \forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}], \quad (2)$$

where we denote $d(i,j)$ as the length of shortest path from i to j without considering (i,k,j), $\|$ as the concatenation operation. Since our graph is directed, we concatenate them in both directions.

Lemma 3.1. *Γ_{ISDEA} in Equation* (2) *is a double invariant triplet representation as per Definition 2.3.*

As in Yang et al. (2015) [76]; Schlichtkrull et al. (2018) [53]; Zhu et al. (2021) [87], we use negative sampling in our training with the difference that we account for both predicting missing nodes and relation types (Definition 2.1). Specifically, for each positive training triplet (i,k,j) such that $\mathbf{A}^{(\text{tr})}_{i,k,j} = 1$, we first randomly corrupt either the head or the tail n_{nd} times to generate the negative (node) examples (i,k,j'). Additionally, we also want our model to learn the correct relation type $(i,?,j)$ between a pair of nodes. Thus, we corrupt relation n_{rl} times to generate negative (relation) examples (i,k',j). In our training, $n_{\text{nd}} = n_{\text{rl}} = 2$; while in evaluation, $n_{\text{nd}} = 50, n_{\text{rl}} = 0$ for node evaluation, and $n_{\text{nd}} = 0, n_{\text{rl}} = 50$ for relation evaluation. Following Schlichtkrull et al. (2018) [53], we use cross-entropy loss to encourage the model to score positive examples higher than corresponding negative examples:

$$\mathcal{L} = -\sum_{(i,k,j)\in\mathcal{S}} \left(\log\left(\Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(\text{tr})})\right) - \frac{1}{n_{\text{nd}} + n_{\text{rl}}} \sum_{p=1}^{n_{\text{nd}}+n_{\text{rl}}} \log\left(1 - \Gamma_{\text{tri}}\left(\left(i'_p, k'_p, j'_p\right), \mathbf{A}^{(\text{tr})}\right)\right)\right), \quad (3)$$

where $S = \left\{(i,k,j) \middle| \mathbf{A}^{(\text{tr})}_{i,k,j} = 1\right\}$, and $\left(i'_p, k'_p, j'_p\right)$ are the p-th negative node or relation example corresponding to (i,k,j).

3.2 Double Equivariant InGram (DEq-InGram)

ISDEA directly obtains double equivariant representations for attributed multigraphs. Alternatively, one can build these double equivariant representations from distributionally double equivariant positional embeddings (Theorem 2.8). To this end, we investigate obtaining double equivariant representations from the positional embeddings of InGram [35], as discussed in Section 2.4.

InGram [35] constructs a *relation graph* as a weighted graph consisting of relations and a heuristic to construct affinity weights between them. It then employs a GNN on the relation graph to generate relation embeddings, which are then fed into another GNN on the original attributed multigraph to generate node embeddings. Finally, InGram uses a variant of DistMult [76] to compute triplet scores from the node and relation embeddings. These embeddings, however, are permutation sensitive due to their reliances on Glorot initialization [28] in each training epoch and test-time inference.

Lemma 3.2. *The triplet representations generated by InGram [35] output distributionally double equivariant positional embeddings (Definition 2.7).*

Theorem 2.8 suggests that averaging InGram's positional embeddings can be used to construct double equivariant attributed multigraph representations. Hence, we propose a Monte Carlo method to estimate these double equivariant graph representations and denote it as DEq-InGram. Specifically, given InGram's triplet score function $\mathbf{Z}_{\text{InGram}}((i,k,j), \mathbf{A}^{(\text{te})}, \boldsymbol{V}^{(0)}, \boldsymbol{R}^{(0)})$ over a test attributed multigraph $\mathbf{A}^{(\text{te})}$, the initial random node embeddings $\boldsymbol{V}^{(0)} \in \mathbb{R}^{N^{(\text{te})} \times d}$, and the initial random relation embeddings $\boldsymbol{R}^{(0)} \in \mathbb{R}^{R^{(\text{te})} \times d'}$ (where d and d' are the dimension sizes), our DEq-InGram produces the following triplet scores:

$$\Gamma_{\text{DEq-InGram}}((i,k,j), \mathbf{A}^{(\text{te})}) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{Z}_{\text{InGram}}((i,k,j), \mathbf{A}^{(\text{te})}, \boldsymbol{V}_m^{(0)}, \boldsymbol{R}_m^{(0)}) \quad (4)$$

where $\{\boldsymbol{V}_m^{(0)}\}_{m=1}^M$ and $\{\boldsymbol{R}_m^{(0)}\}_{m=1}^M$ are M i.i.d. samples drawn from the distribution of initial node and initial relation embeddings respectively (via Glorot initialization).

4 Related Work

A more comprehensive discussion of related work can be found in Appendix D.

Transductive link prediction. In transductive link prediction task, missing links are predicted over a fixed set of nodes and relation types as in training [5, 76, 63]. These (positional) embeddings can be made inductive via Srinivasan & Ribeiro (2020) [54]'s theory but are not designed for predicting new relation types.

Inductive link prediction over new nodes (but not new relations). Rule-induction methods [76, 77, 40, 51] are inherently node-independent which aim to extract First-order Logical Horn clauses from the attributed multigraph. Recently, with the advancement of GNNs, various works [53, 60, 22, 87, 14] have applied the idea of GNN in relational prediction to learn structural node/pairwise representation. Although all these methods can be used to perform *inductive link prediction over solely new nodes*, they can not handle new relation types in test.

Inductive link prediction over both new nodes and new relations (with extra context). Existing methods for querying triplets involving both new nodes and new relations generally assume access to extra context, such as generating language embedding for textual descriptions of unseen relation types [46, 24, 81, 67], a shared background graph connecting seen and unseen relations (e.g., test graph has training relations [31, 10, 12]), or access to graph ontology [25]. Hence, these methods cannot be directly applied to test graphs that neither contain meaningful descriptive information of the unseen relation types (e.g., url links) nor connection with nodes and relation types seen in training.

Inductive link prediction over both new nodes and new relations (no extra context). We focus on this most general doubly inductive link prediction task without additional context data (just the test graph structure is available during inference). To the best of our knowledge, InGram [35] is the first and only existing method capable of performing this task. The connection between InGram and our work has been described in Sections 2.4 and 3.2.

5 Experimental Results

In this section, we aim to answer two questions: **Q1**: Can double equivariant models (ISDEA and DEq-InGram) perform doubly inductive link prediction over attributed multigraphs more accurately than existing methods? **Q2**: Will ISDEA be more stable than DEq-InGram, and will DEq-InGram in turn be more stable than the original InGram [35]? It's essential to remember that, as per our theory in Section 2, ISDEA is designed to directly produce double equivariant representations. In contrast, InGram yields positional embeddings that achieve double equivariance only in expectation.

Datasets. To the best of our knowledge, there are no existing real-world benchmarks that are specially designed to test a model's extrapolation capability for doubly inductive link prediction task with training and test graphs coming from distinct domains with distinct characteristics. Existing datasets such as NL-100, WK-100, and FB-100 from Lee et al. (2023) [35] are typically created by randomly splitting a larger graph (e.g., NELL-995 [72], Wikidata68K [26], FB15K237 [61]) into disjoint node and relation sets, implying that the test and training graphs still come from the same distribution. In contrast, we purposefully create two doubly inductive link prediction benchmark datasets: PediaTypes and WikiTopics, sampled respectively from the OpenEA library [57] and WikiData-5M [68], where by design, the test and training graphs are either from different domains or different topic groups, and are likely to possess different characteristics to fully test model's capability for doubly inductive link prediction.

Baselines. To the best of our knowledge, InGram [35] is the first and only work capable of performing doubly inductive link prediction without needing significant modification to the model. We also run RMPI [25], which is capable of reasoning over new nodes and new relations but requires extra context at test time (test graphs either contain training relations or ontology about unseen relations). In addition, we consider the state-of-the-art link prediction model NBFNet [87] capable of generalizing to new nodes but not new relations and modify its architecture to work with new relations at test time

Table 1: **Relation & Node Hits@10 performance on Doubly Inductive Link Prediction over PediaTypes.** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. The dataset name "X-Y" means training on graph X and testing on graph Y. The best values are shown in bold font, while the second-best values are underlined. The highest standard deviation within each task is highlighted in red color. "Rand" column contains unbiased estimations of the performance from a random predictor. **Both ISDEA and DEq-InGram consistently achieve better results than the baselines with generally smaller standard deviations.** N/A*: Not available due to constant crashes.

(a) **Relation prediction $(i, ?, j)$ performance in %. Higher $\uparrow$ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00
GAT	18.58$\pm$00.52	18.93$\pm$00.33	19.40$\pm$00.28	18.87$\pm$00.19	18.78$\pm$00.28	18.76$\pm$00.33	19.78$\pm$01.39	19.15$\pm$00.35
GIN	19.34$\pm$00.32	19.34$\pm$00.29	18.98$\pm$00.27	18.88$\pm$00.47	19.30$\pm$00.52	18.86$\pm$00.35	18.69$\pm$00.75	18.92$\pm$00.68
GraphConv	19.18$\pm$00.27	19.02$\pm$00.64	19.19$\pm$00.24	18.93$\pm$00.60	19.46$\pm$00.38	19.13$\pm$00.54	19.13$\pm$01.24	18.89$\pm$00.57
NBFNet	21.93$\pm$02.53	22.20$\pm$02.92	18.98$\pm$02.75	7.01$\pm$01.43	23.51$\pm$07.06	23.05$\pm$03.55	31.50$\pm$04.82	35.17$\pm$05.13
RMPI	27.91$\pm$06.48	28.62$\pm$03.75	27.51$\pm$06.48	25.59$\pm$06.48	N/A*	16.76$\pm$04.03	39.03$\pm$20.28	11.77$\pm$07.07
InGram	78.74$\pm$07.48	62.11$\pm$13.60	48.72$\pm$08.94	65.60$\pm$14.42	77.75$\pm$06.60	63.32$\pm$02.78	67.98$\pm$25.45	64.98$\pm$26.69
DEq-InGram (Ours)	**87.94**$\pm$05.68	80.47$\pm$09.90	68.89$\pm$05.45	80.79$\pm$10.51	**91.47**$\pm$01.53	77.03$\pm$04.09	77.72$\pm$21.92	**89.30**$\pm$05.53
ISDEA (Ours)	84.94$\pm$05.00	**84.75**$\pm$02.51	**95.26**$\pm$00.63	**94.23**$\pm$00.71	82.22$\pm$02.44	**88.87**$\pm$02.94	**91.42**$\pm$01.79	85.34$\pm$01.49

(b) **Node prediction $(i, k, ?)$ performance in %. Higher $\uparrow$ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00	19.60$\pm$00.00
GAT	89.77$\pm$00.41	86.83$\pm$00.41	66.24$\pm$02.81	69.08$\pm$00.66	31.08$\pm$01.07	77.05$\pm$00.36	53.51$\pm$00.29	64.13$\pm$00.31
GIN	90.10$\pm$00.61	85.32$\pm$01.18	73.32$\pm$03.35	75.66$\pm$04.85	34.87$\pm$09.12	78.67$\pm$02.46	56.87$\pm$00.44	65.27$\pm$01.14
GraphConv	92.97$\pm$00.11	**90.56**$\pm$00.04	83.58$\pm$00.68	82.64$\pm$00.65	40.59$\pm$01.72	79.28$\pm$01.29	68.91$\pm$00.51	76.50$\pm$00.14
NBFNet	87.64$\pm$01.81	89.77$\pm$00.80	85.56$\pm$02.07	59.78$\pm$03.73	63.23$\pm$03.65	78.24$\pm$00.90	49.97$\pm$01.44	66.36$\pm$02.64
RMPI	89.59$\pm$06.61	81.79$\pm$02.17	82.93$\pm$03.56	81.38$\pm$06.19	N/A*	65.76$\pm$07.45	55.67$\pm$06.61	71.03$\pm$02.12
InGram	92.32$\pm$01.00	83.71$\pm$03.53	90.82$\pm$01.84	92.15$\pm$00.90	61.44$\pm$09.84	87.60$\pm$01.21	54.79$\pm$08.81	67.84$\pm$06.38
DEq-InGram (Ours)	**94.47**$\pm$00.60	88.90$\pm$02.06	**93.85**$\pm$00.36	**94.02**$\pm$00.74	**71.94**$\pm$07.37	**91.47**$\pm$00.62	71.53$\pm$04.78	**80.53**$\pm$07.96
ISDEA (Ours)	76.28$\pm$00.05	77.51$\pm$01.46	82.24$\pm$00.94	81.80$\pm$00.68	66.69$\pm$01.01	75.19$\pm$03.12	**72.87**$\pm$01.03	76.41$\pm$01.52

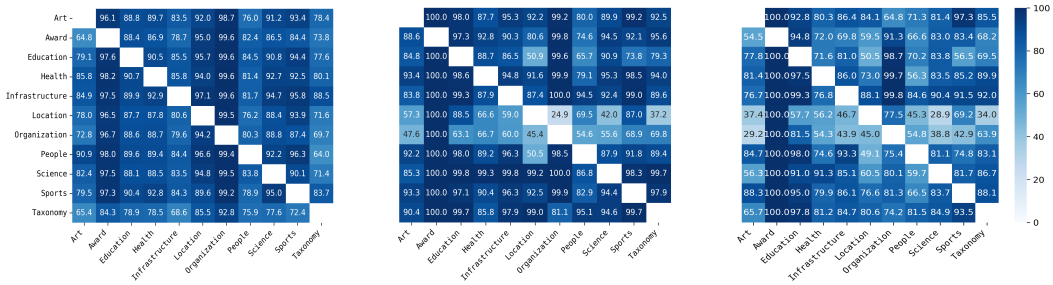
(following Lee et al. (2023) [35]'s approach). We also compare our models with message-passing GNNs, including GAT [64], GIN [73], GraphConv [41], which treats the graph as a homogeneous graph by ignoring the relation types. For fair comparisons, we add distance features as in Equation (2) to increase the expressiveness of these GNNs. Additional baseline details are in Appendix E.

Relation and Node Prediction Tasks. We report the Hits@10 performances over 5 runs of different random seeds for all models on both the relation prediction task of $(i, ?, j)$ and the more traditional node prediction task of $(i, k, ?)$. For each task, we sample 50 negative triplets for each ground-truth positive target triplet during test evaluation by corrupting the relation type or the tail node respectively. *Further experiment details on synthetic tasks, additional datasets from Lee et al. (2023) [35], baseline implementations, ablation studies, and other metrics (e.g., MRR, Hits@1) can be found in Appendix E.*

5.1 Doubly Inductive Link Prediction over PediaTypes Dataset

The OpenEA library [57] contains multiple attributed multigraphs of relational databases (i.e., knowledge graphs) from different domains on similar topics, such as DBPedia [36] in different languages (English, French and German), YAGO [48] and Wikidata [66]. We create a new dataset **PediaTypes** (details in Appendix E.1.2) by sampling from the OpenEA library [57], including pairs of attributed multigraphs such as English-to-French DBPedia (denoted as EN-FR), DBPedia-to-YAGO (denoted as DB-YG), etc.. In each graph, triplets are randomly divided into 80% training, 10% validation, and 10% test. We then train and validate the model on one of the graphs (e.g., EN) and directly apply it to another graph (e.g., DE), which has completely new nodes and new relation types.

Table 1a shows the results on the relation prediction task, and Table 1b shows the node prediction task on PediaTypes. Across all scenarios on both tasks, our models, ISDEA and DEq-InGram, obtain significantly better average performance, achieving up to 41.40% relative improvement in relation prediction and up to 13.78% relative improvement in node prediction compared to the best-performing baseline. Furthermore, ISDEA tends to have smaller standard deviations than DEq-InGram, and both demonstrate much smaller standard deviations than InGram in almost all scenarios, corroborating our theoretical predictions in Section 2 that a model directly producing double equivariant representations will be more stable than positional embeddings, which are only double equivariant in expectation.



(a) ISDEA (b) DEq-InGram (c) Original InGram

Figure 3: **Relation Hits@10 performance over WikiTopics** for ISDEA, DEq-InGram, and InGram [35]. Each row corresponds to a training graph, and each column corresponds to a test graph. A darker color means better performance. **Both ISDEA and DEq-InGram consistently show better performance than the baseline InGram. In addition, ISDEA exhibits more consistent results across different train-test scenarios than DEq-InGram.**

Interestingly, we observe that in the node prediction task, the message-passing GNNs (GAT, GIN, and GraphConv) achieve quite excellent performances, even though *they completely disregard the information carried by different relation types and treat the attributed multigraph as a homogeneous graph.* This observation corroborates with the conclusions of Jambor et al. (2021) [32]. Indeed, only 4 out of 8 scenarios did InGram outperform the message-passing GNNs on this task, suggesting the node prediction task might be too easy because a homogeneous link prediction model can do decently well.

5.2 Doubly Inductive Link Prediction over WikiTopics Dataset

WikiData-5M [68] is a large knowledge graph dataset containing over 4M entities, 20M triplets, and 822 relation types from the Wikipedia website. The vast number of relation types span a wide range of topics, such as arts and media, education and academics, sports and gaming, etc.. Hence, an interesting question arises: can a model learn on the subgraph corresponding to only one topic, e.g., arts, and be directly applicable to reasoning on the subgraph of another topic, e.g., education? To this end, we create another new dataset **WikiTopics** containing a collection of 11 different attributed multigraphs, each containing relation types specific to only a particular topic. These graphs are created by first breaking all relation types of WikiData-5M [68] into 11 non-overlapping topic groups and then selecting triplets within each topic group (details and statistics in Appendix E.1.3). We train the models on each of the 11 graphs for 5 random seeds, and for each trained model checkpoint, we cross-test it on all the other 10 graphs, resulting in a total of 550 statistics. We report the mean results across random seeds in heatmaps.

Figure 3 shows the results of ISDEA, DEq-InGram, and InGram on WikiTopics for the relation prediction task. The results on the relatively easier task of node prediction are relegated to Appendix E.1.3. In general, we observe that both ISDEA and DEq-InGram showcase darker colors than the baseline InGram on the heatmaps, indicating more accurate predictions. In addition, the results of ISDEA are more consistent than DEq-InGram across different train-test scenarios. For example, whereas the worst performance of DEq-InGram is 24.9% Hits@10 on LOCATION-ORGANIZATION, ISDEA's worst performance is 64.0% Hits@10 on PEOPLE-TAXONOMY. This further corroborates that a model directly modeling double equivariant representations will be more stable than positional methods, not only across different random seeds, *but also across different training and test scenarios.*

6 Conclusion

This work formally introduced the doubly inductive link prediction task defined over both new nodes and new relation types in the test data. It also defined *double equivariant models* and *distributionally double equivariant positional embedding* models for this task. We showed that, similar to how node equivariances impose learning structural node representations in unattributed graphs, double (node and relation) equivariances impose relational structure learning for attributed multigraphs. We then introduced a blueprint for double equivariant neural network architectures that enables inductive link prediction over new nodes and relations without the need for additional data or test-time adaptations. Finally, we proposed two real-world doubly inductive link prediction benchmarks, and empirically verified the ability of our proposed approaches to extrapolate to both new nodes and relation types.

References

[1] Ralph Abboud, Ismail Ilkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising power of graph neural networks with random node initialization. *arXiv preprint arXiv:2010.01179*, 2020.

[2] Pablo Barcelo, Mikhail Galkin, Christopher Morris, and Miguel Romero Orth. Weisfeiler and leman go relational. *arXiv preprint arXiv:2211.17113*, 2022.

[3] Pablo Barceló, Egor Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan-Pablo Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

[4] Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2021.

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

[6] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[7] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[8] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[9] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. Graph heterogeneous multi-relational recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3958–3966, 2021.

[10] Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6271–6278, 2021.

[11] Mingyang Chen, Wen Zhang, Yuxia Geng, Zezhong Xu, Jeff Z Pan, and Huajun Chen. Generalizing to unseen elements: A survey on knowledge extrapolation for knowledge graphs. *arXiv preprint arXiv:2302.01859*, 2023.

[12] Mingyang Chen, Wen Zhang, Zhen Yao, Xiangnan Chen, Mengxiao Ding, Fei Huang, and Huajun Chen. Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1966–1972. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[13] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4217–4226, 2019.

[14] Yihong Chen, Pushkar Mishra, Luca Franceschi, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Refactor gnns: Revisiting factorisation-based models from a message-passing perspective. In *Advances in Neural Information Processing Systems*, 2022.

[15] Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. Rlogic: Recursive logical rule learning from knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 179–189, 2022.

[16] Michele Coscia, Giulio Rossetti, Diego Pennacchioli, Damiano Ceccarelli, and Fosca Giannotti. " you know because i know" a multidimensional network approach to human resources problem. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 434–441, 2013.

[17] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

[18] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

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

[20] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[21] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422, 2013.

[22] Mikhail Galkin, Etienne Denis, Jiapeng Wu, and William L Hamilton. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *International Conference on Learning Representations*, 2021.

[23] Mikhail Galkin, Zhaocheng Zhu, Hongyu Ren, and Jian Tang. Inductive logical query answering in knowledge graphs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[24] Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Jeff Z. Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. Ontozsl: Ontology-enhanced zero-shot learning. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3325–3336. ACM / IW3C2, 2021.

[25] Yuxia Geng, Jiaoyan Chen, Jeff Z Pan, Mingyang Chen, Song Jiang, Wen Zhang, and Huajun Chen. Relational message passing for fully inductive knowledge graph completion. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1221–1233. IEEE, 2023.

[26] Genet Asefa Gesese, Harald Sack, and Mehwish Alam. Raild: Towards leveraging relation features for inductive link prediction in knowledge graphs. In *Proceedings of the 11th International Joint Conference on Knowledge Graphs*, pages 82–90, 2022.

[27] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 2017.

[28] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[29] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[30] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[31] Qian Huang, Hongyu Ren, and Jure Leskovec. Few-shot relational reasoning via connection subgraph pretraining. In *Neural Information Processing Systems*, 2022.

[32] Dora Jambor, Komal Teru, Joelle Pineau, and William L Hamilton. Exploring the limits of few-shot link prediction in knowledge graphs. *arXiv preprint arXiv:2102.03419*, 2021.

[33] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

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

[35] Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. InGram: Inductive knowledge graph embedding via relation graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 18796–18809, 2023.

[36] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

[37] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 631–636, New York, NY, USA, 2006. Association for Computing Machinery.

[38] Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3376–3381, 2019.

[39] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR, 2020.

[40] Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In *International semantic web conference*, pages 3–20. Springer, 2018.

[41] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, Jul. 2019.

[42] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673. PMLR, 2019.

[43] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[44] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[45] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

[46] Pengda Qin, Xin Wang, Wenhu Chen, Chunyun Zhang, Weiran Xu, and William Yang Wang. Generative adversarial zero-shot relational learning for knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8673–8680, 2020.

[47] Haiquan Qiu, Yongqi Zhang, Yong Li, and Quanming Yao. Logical expressiveness of graph neural network for knowledge graph reasoning. *arXiv preprint arXiv:2303.12306*, 2023.

[48] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*, pages 177–185. Springer, 2016.

[49] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Little Ball of Fur: A Python Library for Graph Sampling. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, page 3133–3140. ACM, 2020.

[50] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020.

[51] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

[52] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. In *Proceedings of the 2021 SIAM international conference on data mining (SDM)*, pages 333–341. SIAM, 2021.

[53] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

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

[55] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18, 2018.

[56] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 222–229, 2020.

[57] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11):2326–2340, 2020.

[58] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

[59] Ilya Sutskever, Joshua Tenenbaum, and Russ R Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. *Advances in neural information processing systems*, 22, 2009.

[60] Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pages 9448–9457. PMLR, 2020.

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

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

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

[64] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[65] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2018.

[66] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[67] Hongwei Wang, Hongyu Ren, and Jure Leskovec. Relational message passing for knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1697–1707, 2021.

[68] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

[69] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[70] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 349–357, 2018.

[71] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1120–1128, 2022.

[72] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1980–1990, 2018.

[73] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[74] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual knowledge graph alignment via graph matching neural network. In *Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), 2019.

[75] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. Dynamic knowledge graph alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4564–4572, 2021.

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

[77] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

[78] Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *International conference on machine learning*, pages 7134–7143. PMLR, 2019.

[79] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 347–350, 2013.

[80] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[81] Hanwen Zha, Zhiyu Chen, and Xifeng Yan. Inductive relation prediction by bert. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:5923–5931, Jun. 2022.

[82] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. Few-shot knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3041–3048, 2020.

[83] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[84] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34:9061–9073, 2021.

[85] Ming Zhao, Weijia Jia, and Yusheng Huang. Attention-based aggregation graph networks for knowledge graph information transfer. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part II 24*, pages 542–554. Springer, 2020.

[86] Zhaocheng Zhu, Mikhail Galkin, Zuobai Zhang, and Jian Tang. Neural-symbolic models for logical queries on knowledge graphs. In *International Conference on Machine Learning*, pages 27454–27478. PMLR, 2022.

[87] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.

A Additional example for doubly inductive link prediction

This example depicts an even harder scenario than the example in Figure 1, obtained from a fictional alien civilization. Knowing nothing about alien languages, we note that in training, all adjacent relations are different. Minimally, we could predict the missing relation in red in test data is not "$\not\measuredangle$". By introducing equivariance in relations, it is possible for a model to predict relation types uniformly over the set of other $(R-1)$ relations except for the existing relation "$\not\measuredangle$", which is all we know about the aliens.

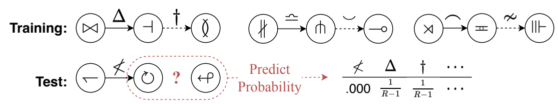


Figure 4: **Alien discrete attributed multigraph:** The task is to predict the missing relation "?" in red. Training only tells us that relations do not repeat in a path.

B Connection to Double Equivariant Logical Reasoning

In what follows, we follow the literature and connect link prediction in discrete attributed multigraph to logical induction [60, 86, 47]. Existing logical induction requires all involved relations to be observed at least once, thus, such logical reasoning can not generalize to new relation types. We propose the Universally Quantified Entity and Relation (UQER) Horn clause, a double equivariant extension of conventional logical reasoning, which is capable of generalizing to new relation types, and show that the double invariant triplet representation in Definition 2.4 is capable of encoding such set of UQER Horn Clauses.

Definition B.1 (Universally Quantified Entity and Relation (UQER) Horn clause)**.** An UQER Horn clause involving M nodes and K relations is defined by an indicator tensor $\mathbf{B} \in \{0,1\}^{M\times K\times M}$:

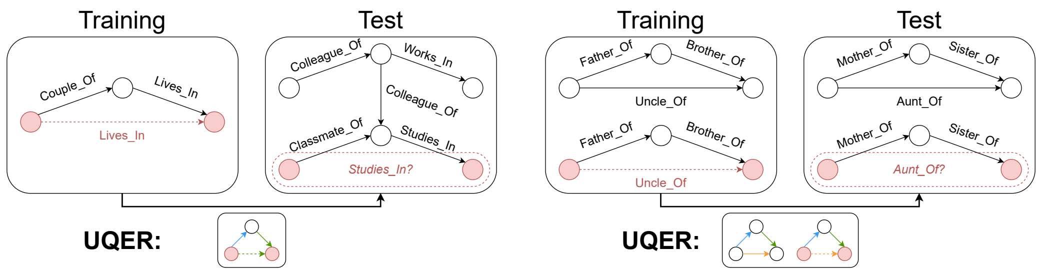
$$\forall E_1 \in \mathcal{V}^{(*)}, \Big(\forall E_u \in \mathcal{V}^{(*)} \setminus \{E_1,\dots,E_{u-1}\}\Big)_{u=2}^{M}, \forall C_1 \in \mathcal{R}^{(*)}, \Big(\forall C_c \in \mathcal{R}^{(*)} \setminus \{C_1,\dots,C_{c-1}\}\Big)_{c=2}^{K},$$
$$\bigwedge_{\substack{u,u'=1,\dots,M,c=1,\dots,K,\\ \mathbf{B}_{u,c,u'}=1}} (E_u, C_c, E_{u'}) \implies (E_1, C_1, E_h), \tag{5}$$

for any node set $\mathcal{V}^{(*)}$ and relation set $\mathcal{R}^{(*)}$ with number of nodes and relations $N^{(*)}, R^{(*)}$ s.t. $N^{(*)} \geq M, R^{(*)} \geq K$, $h \in \{1,2\}$ (where $h=1$ indicates a self-loop relation or a relational node attribute), where if $M > h$, $\forall u \in \{h+1,\dots,M\}$, $\sum_{u'=1}^{M}\sum_{c=1}^{K} \mathbf{B}_{u,c,u'} + \mathbf{B}_{u',c,u} \geq 1$, and if $K \geq 2, \forall c \in \{2,\dots,K\}, \sum_{u=1}^{M}\sum_{u'=1}^{M} \mathbf{B}_{u,c,u'} + \mathbf{B}_{u',c,u} \geq 1$ (every variable should appear at least once in the formula).

Note that our definition of UQER Horn clauses (Definition B.1) is a generalization of the First Order Logic (FOL) clauses in [77, 40, 51, 60] such that the relations in the Horn clauses are also universally quantified rather than predefined constants. UQER can be used to predict new relations in the test attributed multigraph with *pattern matching*, i.e., if the left-hand-side (condition) of a UQER can be satisfied in the test attributed multigraph, then the right-hand-side (implication) triplet should be present. In Figure 5, we illustrate two examples using UQER to predict new relations at test time.

We now connect our double equivariant representations (Definition 2.3) with the UQER Horn clauses.

Theorem B.2. *For any UQER Horn clause defined by* $\mathbf{B} \in \{0,1\}^{M\times K\times M}$ *(Definition B.1), there exists a double invariant triplet predictor* $\Gamma_{tri} : \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} ([N]\times[R]\times[N]) \times \mathbb{A} \to \{0,1\}$ *(Definition 2.3), such that for any set of truth statements* $\mathcal{S} \subseteq \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)}$ *and their equivalent tensor representation* $\mathbf{A}^{(*)} \in \mathbb{A}$ *(where* $\mathbf{A}^{(*)}_{i,k,j} = 1$ *iff* $(v_i^{(*)}, r_k^{(*)}, v_j^{(*)}) \in \mathcal{S}$*), it satisfies* $\Gamma_{tri}((i,k,j),\mathbf{A}^{(*)}) = 1$ *iff* $(i,k,j) \in \mathcal{S}'$*, where* $\mathcal{S}' = \Big\{ (i,k,j) \,\Big|\, \forall (i,k,j)$ *, such that* $(E_1, C_1, E_2) = \Big(v_i^{(*)}, r_k^{(*)}, v_j^{(*)}\Big) \in \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)}$, $\exists^{M-2} E_3, \dots, E_M \in \mathcal{V}^{(*)} \setminus \{E_1, E_2\}, \exists^{K-1} C_2, \dots, C_K \in \mathcal{R}^{(*)} \setminus \{C_1\}$*, where* $\forall (u,c,u') \in [M]\times[K]\times[M], \mathbf{B}_{u,c,u'} = 1 \Rightarrow (E_u, C_c, E_{u'}) \in \mathcal{S}\Big\}$ *is the set of true statements induced by modus ponens by the truth statements* $\mathcal{S}$ *and the UQER Horn clause, where the existential quantifier* $\exists^k$ *means exists at least* k *distinct values.*



(a) **A Simple UQER Application** (b) **A Complex UQER Application**

Figure 5: (a) The UQER (bottom) learned from training can be used to predict missing new relation "Studies_In" in red since an assignment of left-hand-side of the UQER $(E_1, \text{Classmate\_Of}, E_3) \wedge (E_3, \text{Studies\_In}, E_2)$ is satisfied in test. (b) UQER can contain disconnected components, giving more freedom to its application. For example, the UQER (bottom) can be learned from training to repeat arbitrary logical chain, which makes it possible to deal with new female relations at test time and will predict "Aunt_Of" in test just as "Uncle_Of" (red) in training.

The full proof is in Appendix C, showing how the universal quantification in Definition B.1 is a double invariant predictor.

C Proofs

Theorem 2.6. *For all $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, given a double invariant triplet representation Γ_{tri}, we can construct a double equivariant graph representation as $\left(\Gamma_{gra}(\mathbf{A}^{(*)})\right)_{i,k,j} := \Gamma_{tri}((i,k,j), \mathbf{A}^{(*)})$, $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, and vice-versa.*

Proof. ($\Rightarrow$) For any discrete attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, $\Gamma_{\text{tri}} : \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} ([N] \times [R] \times [N]) \times \mathbb{A} \to \mathbb{R}^d, d \geq 1$ is a double invariant triplet representation as in Definition 2.3. Using the double invariant triplet representation, we can define a function $\Gamma_{\text{gra}} : \mathbb{A} \to \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}$ such that $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, $(\Gamma_{\text{gra}}(\mathbf{A}^{(*)}))_{i,k,j,:} = \Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(*)})$. Then $\forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}$, $(\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}^{(*)}))_{\phi \circ i, \tau \circ k, \phi \circ j,:} = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)})$. We know $\Gamma_{\text{tri}}((i,k,j), \mathbf{A}) = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)})$. Thus we conclude, $\forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}, \forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, $(\phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}^{(*)}))_{\phi \circ i, \tau \circ k, \phi \circ j,:} = (\Gamma_{\text{gra}}(\mathbf{A}^{(*)}))_{i,k,j,:} = \Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(*)}) = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)}) = (\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}^{(*)}))_{\phi \circ i, \tau \circ k, \phi \circ j,:}$. 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 2.5.

($\Leftarrow$) For any discrete attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, assume $\Gamma_{\text{gra}} : \mathbb{A} \to \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}$ is a double equivariant representation as Definition 2.5. Since $\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}^{(*)}) = \phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}^{(*)})$, then $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, $(\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}^{(*)}))_{\phi \circ i, \tau \circ k, \phi \circ j} = (\phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}^{(*)}))_{\phi \circ i, \tau \circ k, \phi \circ j} = (\Gamma_{\text{gra}}(\mathbf{A}))_{i,k,j}$. Then we can define $\Gamma_{\text{tri}} : \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} ([N] \times [R] \times [N]) \times \mathbb{A} \to \mathbb{R}^d, d \geq 1$, such that $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, $\Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(*)}) = (\Gamma_{\text{gra}}(\mathbf{A}^{(*)}))_{i,k,j}$. It is clear that $\Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(*)}) = (\Gamma_{\text{gra}}(\mathbf{A}^{(*)}))_{i,k,j} = (\Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}^{(*)}))_{\phi \circ i, \tau \circ k, \phi \circ j} = \Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)})$. Thus, we show Γ_{tri} is a double invariant triplet representation as in Definition 2.3. $\square$

Theorem 2.8 (From distributional double equivariant positional embeddings to double equivariant representations)**.** *For any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$, the average $\mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}]$ is a double equivariant attributed multigraph representation (Definition 2.5) for any distributional double equivariant positional embeddings $\mathbf{Z}|\mathbf{A}^{(*)}$ (Definition 2.7).*

Proof. Based on Definition 2.7, for any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, the distributionally double equivariant positional embeddings of $\mathbf{A}^{(*)}$ are defined as joint samples of random variables $\mathbf{Z}|\mathbf{A}^{(*)} \sim p(\mathbf{Z}|\mathbf{A}^{(*)})$, where the tensor $\mathbf{Z}$ is defined as $\mathbf{Z}_{i,k,j} \in \mathbb{R}^d, d \geq 1, \forall(i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, where we say $p(\mathbf{Z}|\mathbf{A}^{(*)})$ is a double equivariant probability distribution on $\mathbf{A}^{(*)}$ defined as $\forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}, p(\mathbf{Z}|\mathbf{A}^{(*)}) = p(\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)})$.

The tensor $\mathbf{Z}$ is defined as $\mathbf{Z}_{i,k,j} \in \mathbb{R}^d, \forall(i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, thus $\mathbf{Z} \in \mathbb{R}^{N^{(*)} \times R^{(*)} \times N^{(*)} \times d}$. So we can consider $\mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}]$ as a function on $\mathbf{A}^{(*)}$, and output a representation in $\mathbb{R}^{N^{(*)} \times R^{(*)} \times N^{(*)} \times d}$. Since $\forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}, p(\mathbf{Z}|\mathbf{A}^{(*)}) = p(\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)})$, it is clear to have $\forall \phi \in \mathbb{S}_{N^{(*)}}, \forall \tau \in \mathbb{S}_{R^{(*)}}, \phi \circ \tau \circ \mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}] = \phi \circ \tau \circ \int \boldsymbol{z} p(\mathbf{Z} = \boldsymbol{z}|\mathbf{A}^{(*)}) d\boldsymbol{z} = \int \phi \circ \tau \circ \boldsymbol{z} p(\mathbf{Z} = \boldsymbol{z}|\mathbf{A}^{(*)}) d\boldsymbol{z} = \int \phi \circ \tau \circ \boldsymbol{z} p(\phi \circ \tau \circ \mathbf{Z} = \phi \circ \tau \circ \boldsymbol{z}|\phi \circ \tau \circ \mathbf{A}^{(*)}) d(\phi \circ \tau \circ \boldsymbol{z}) = \mathbb{E}_{p(\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)})}[\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)}]$. Since the permutation ϕ, τ only changes the ordering of the output representation element-wise, we can interchange the permutations with the integral.

Finally, for any attributed multigraph $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, we can define $\Gamma_{\text{gra}}(\mathbf{A}^{(*)}) : \mathbb{A} \to \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \mathbb{R}^{N \times R \times N \times d}, d \geq 1$ such that $\Gamma_{\text{gra}}(\mathbf{A}^{(*)}) := \mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}]$. And we can derive $\phi \circ \tau \circ \Gamma_{\text{gra}}(\mathbf{A}^{(*)}) = \phi \circ \tau \circ \mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}] = \mathbb{E}_{p(\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)})}[\phi \circ \tau \circ \mathbf{Z}|\phi \circ \tau \circ \mathbf{A}^{(*)}] = \Gamma_{\text{gra}}(\phi \circ \tau \circ \mathbf{A}^{(*)})$. Thus, $\Gamma_{\text{gra}}(\mathbf{A}^{(*)}) := \mathbb{E}_{p(\mathbf{Z}|\mathbf{A}^{(*)})}[\mathbf{Z}|\mathbf{A}^{(*)}]$ is a double equivariant attributed multigraph representation as per Definition 2.5. $\square$

Lemma 3.1. *$\Gamma_{\textit{ISDEA}}$ in Equation* (2) *is a double invariant triplet representation as per Definition 2.3.*

Proof. From our model architecture (Equation (2)), $\Gamma_{\text{ISDEA}}((i,k,j), \mathbf{A}^{(*)}) = (h_{i,k}^{(T)} \parallel h_{j,k}^{(T)} \parallel d(i,j) \parallel d(j,i))$. Using DSS layers, we can guarantee the node representations $h_{i,k}^{(T)}$ we learn are double invariant under the node and relation permutations, where $h_{i,k}^{(T)}$ in $\mathbf{A}^{(*)}$ is equal to $h_{\phi \circ i, \tau \circ k}^{(T)}$ in $\phi \circ \tau \circ \mathbf{A}^{(*)}$. It is also clear that the distance function is invariant to node and relation permutations, i.e. $\forall i,j \in [N^{(*)}]$, $d(i,j)$ in $\mathbf{A}^{(*)}$ is the same as $d(\phi \circ i, \phi \circ j)$ in $\phi \circ \tau \circ \mathbf{A}^{(*)}$. Thus $\Gamma_{\text{ISDEA}}((i,k,j), \mathbf{A}^{(*)}) = \Gamma_{\text{ISDEA}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)})$ is a double invariant triplet representation as in Definition 2.5. $\square$

Lemma 3.2. *The triplet representations generated by InGram [35] output distributionally double equivariant positional embeddings (Definition 2.7).*

Proof. To solve doubly inductive link prediction, InGram [35] first constructs a *relation graph*, in which the relation types are treated as nodes, and the edges between them are weighted by the affinity scores, a measure of co-occurrence between relation types in the original attributed multigraph. It then employs a variant of the GATv2 [65, 6] on the relation graph to propagate and generate embeddings for the relation types. These relation embeddings, together with another GATv2, are applied to the original attributed multigraph to generate embeddings for the nodes. Finally, a variant of DistMult [76] is used to compute the scores for individual triplets from the embeddings of the head and tail nodes and the embedding of the relation.

If the input node and relation embeddings to the InGram model were to be the same across all nodes and across all relation types respectively (such as vectors of all ones), then InGram would have produced double structural representations for the triplets (definition 2.3). Simply put, this is because the relation graphs proposed byLee et al. (2023) [35] encode only the structural features of the relation types (their mutual structural affinity), which is double equivariant to the permutation of relation type and node indices. Since the same initial embeddings for all nodes and relations are naively double equivariant, and the GATv2 [65, 6] is a message-passing neural network [27] that also produces equivariant representations, the final relation embeddings would be double equivariant. Same analysis will also show the final node embeddings are double equivariant.

However, to improve the expressivity of the model, Lee et al. (2023) [35] chose to randomly re-initialize the input embeddings for all node and relation types using Glorot initialization [28] *for each epoch during training*, a technique inspired by recent studies on the expressive power of GNNs [1, 52, 42]. Unfortunately, random initial features break the double equivariance of the generated representations, making them sensitive to the permutation of node and relation type indices. However, since the initial node $\boldsymbol{V}^{(0)}$ and relation embeddings $\boldsymbol{R}^{(0)}$ are randomly initialized, and by design of InGram architecture, we have $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}], \forall \phi \in \mathbb{S}_{N^{(*)}}, \tau \in \mathbb{S}_{R^{(*)}}, \mathbf{Z}_{\text{InGram}}((i,k,j), \mathbf{A}^{(*)}, \boldsymbol{V}^{(0)}, \boldsymbol{R}^{(0)}) = \mathbf{Z}_{\text{InGram}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \mathbf{A}^{(*)}, \boldsymbol{V}^{(0)}, \boldsymbol{R}^{(0)})$ for any random samples of node and relation embeddings $v^{(0)}, r^{(0)}$. We define $\mathbf{Z}_{\text{InGram}}|\mathbf{A}^{(*)} = [\mathbf{Z}_{\text{InGram}}((i,k,j), \mathbf{A}^{(*)}, \boldsymbol{V}^{(0)}, \boldsymbol{R}^{(0)}))]_{(i,k,j)\in[N^{(*)}]\times[R^{(*)}]\times[N^{(*)}]}$, and $\phi \circ \tau \circ \mathbf{Z}_{\text{InGram}}|\phi \circ \tau \circ \mathbf{A}^{(*)} = [\mathbf{Z}_{\text{InGram}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)}, \boldsymbol{V}^{(0)}, \boldsymbol{R}^{(0)}))]_{(\phi\circ i,\tau\circ k,\phi\circ j)\in[N^{(*)}]\times[R^{(*)}]\times[N^{(*)}]}$. Since $\boldsymbol{V}^{(0)}, \boldsymbol{R}^{(0)}$ random variables that do not change with permutations, we can easily derive $p(\phi \circ \tau \circ \mathbf{Z}_{\text{InGram}}|\phi \circ \tau \circ \mathbf{A}^{(*)}) = p(\mathbf{Z}_{\text{InGram}}|\mathbf{A}^{(*)})$. Thus, InGram is a distributionally double equivariant positional graph embedding of $\mathbf{A}^{(*)}$ as per Definition 2.7. $\square$

Theorem B.2. *For any UQER Horn clause defined by $\mathbf{B} \in \{0,1\}^{M\times K\times M}$ (Definition B.1), there exists a double invariant triplet predictor $\Gamma_{tri} : \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} ([N] \times [R] \times [N]) \times \mathbb{A} \to \{0,1\}$ (Definition 2.3), such that for any set of truth statements $\mathcal{S} \subseteq \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)}$ and their equivalent tensor representation $\mathbf{A}^{(*)} \in \mathbb{A}$ (where $\mathbf{A}^{(*)}_{i,k,j} = 1$ iff $(v_i^{(*)}, r_k^{(*)}, v_j^{(*)}) \in \mathcal{S}$), it satisfies $\Gamma_{tri}((i,k,j), \mathbf{A}^{(*)}) = 1$ iff $(i,k,j) \in \mathcal{S}'$, where $\mathcal{S}' = \Big\{ (i,k,j) \,\Big|\, \forall (i,k,j)\,, \text{such that}\,(E_1, C_1, E_2) = \Big(v_i^{(*)}, r_k^{(*)}, v_j^{(*)}\Big) \in \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)},\ \exists^{M-2} E_3, ..., E_M \in \mathcal{V}^{(*)} \setminus \{E_1, E_2\}\,, \exists^{K-1} C_2, ..., C_K \in \mathcal{R}^{(*)} \setminus \{C_1\}\,, \text{where}\, \forall (u,c,u') \in [M] \times [K] \times [M], \mathbf{B}_{u,c,u'} = 1 \Rightarrow (E_u, C_c, E_{u'}) \in \mathcal{S} \Big\}$ is the set of true statements induced by modus ponens by the truth statements $\mathcal{S}$ and the UQER Horn clause, where the existential quantifier $\exists^k$ means exists at least k distinct values.*

Proof. Recall that we have two different cases $h = 1$ and $h = 2$ for Equation (5) in Definition B.1 of UQER. For the ease of proof, we will focus on the case where $h = 2$ in the following content, and for the case $h = 1$, the proof will be the same.

Given $h = 2$, any UQER is defined by $\mathbf{B} \in \{0,1\}^{M\times K\times M}$ as

$$\forall E_1 \in \mathcal{V}^{(*)}, \Big(\forall E_u \in \mathcal{V}^{(*)} \setminus \{E_1, \ldots, E_{u-1}\}\Big)_{u=2}^{M}, \forall C_1 \in \mathcal{R}^{(*)}, \Big(\forall C_c \in \mathcal{R}^{(*)} \setminus \{C_1, \ldots, C_{c-1}\}\Big)_{c=2}^{K},$$
$$\bigwedge_{\substack{u,u'=1,\ldots,M,c=1,\ldots,K,\\ \mathbf{B}_{u,c,u'}=1}} (E_u, C_c, E_{u'}) \implies (E_1, C_1, E_h), \tag{6}$$

for any node set $\mathcal{V}^{(*)}$ and relation set $\mathcal{R}^{(*)}$ with number of nodes and relations $N^{(*)}, R^{(*)}$ s.t. $N^{(*)} \geq M, R^{(*)} \geq K$, where if $M > 2, \forall u \in \{3, \ldots, M\}, \sum_{u'=1}^{M} \sum_{c=1}^{K} \mathbf{B}_{u,c,u'} + \mathbf{B}_{u',c,u} \geq 1$, and if $K \geq 2, \forall c \in \{2, \ldots, K\}, \sum_{u=1}^{M} \sum_{u'=1}^{M} \mathbf{B}_{u,c,u'} + \mathbf{B}_{u',c,u} \geq 1$ (every variable should appear at least once in the formula).

For all sets of truth statements $\forall \mathcal{S} \subseteq \cup_{N=1}^{\infty} \cup_{R=2}^{\infty} \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)}$, it has an equivalent tensor representation $\mathbf{A}^{(*)} \in \{0,1\}^{N^{(*)}\times R^{(*)}\times N^{(*)}}$ such that $\mathbf{A}_{i,k,j} = 1 \iff (v_i^{(*)}, r_k^{(*)}, v_j^{(*)}) \in \mathcal{S}$. We can then define a triplet representation Γ_{tri} based on the given UQER as, $\forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$,

$$\Gamma_{\text{tri}}((i,k,j), \mathbf{A}^{(*)}) = \begin{cases} 1 & \text{if } (i,k,j) \in \mathcal{S}' \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where we define $\mathcal{S}' = \Big\{ (i,k,j) \,\Big|\, \forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}], \text{such that}\,(E_1, C_1, E_2) = \Big(v_i^{(*)}, r_k^{(*)}, v_j^{(*)}\Big) \in \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)},\ \exists^{M-2} E_3, ..., E_M \in \mathcal{V}^{(*)} \setminus \{E_1, E_2\}\,, \exists^{K-1} C_2, ..., C_K \in \mathcal{R}^{(*)} \setminus \{C_1\}\,, \text{ where } \forall (u,c,u') \in [M] \times [K] \times [M], \mathbf{B}_{u,c,u'} = 1 \Rightarrow (E_u, C_c, E_{u'}) \in \mathcal{S} \Big\}$ is the set of true statements induced by modus ponens from the truth statements $\mathcal{S}$ and the UQER Horn Clause, where the existential quantifier $\exists^k$ means exists at least k distinct values.

All we need to show is that Equation (7) is a double invariant triplet representation. For any node permutation $\phi \in \mathbb{S}_{N^{(*)}}$ and relation permutation $\tau \in \mathbb{S}_{R^{(*)}}$ of $\mathbf{A}^{(*)}$, we define $\phi \circ \tau \circ \mathcal{S} = \{(v_{\phi\circ i}^{(*)}, r_{\tau\circ k}^{(*)}, v_{\phi\circ j}^{(*)}) | (v_i^{(*)}, r_k^{(*)}, v_j^{(*)}) \in \mathcal{S}\}$ which corresponds to their equivalent tensor representation $\phi \circ \tau \circ \mathbf{A}^{(*)}$, where $(\phi \circ \tau \circ \mathbf{A}^{(*)})_{\phi\circ i,\tau\circ k,\phi\circ j} = 1 \iff (v_i^{(*)}, r_k^{(*)}, v_j^{(*)}) \in \mathcal{S}$ otherwise 0. Similarly, we have $\phi \circ \tau \circ \mathcal{S}' = \Big\{ (\phi \circ i, \tau \circ k, \phi \circ j) \,\Big|\, \forall (i,k,j) \in [N^{(*)}] \times [R^{(*)}] \times$

$[N^{(*)}]$, such that $(E_1, C_1, E_2) = \left(v^{(*)}_{\phi \circ i}, r^{(*)}_{\tau \circ k}, v^{(*)}_{\phi \circ j}\right) \in \mathcal{V}^{(*)} \times \mathcal{R}^{(*)} \times \mathcal{V}^{(*)},\ \exists^{M-2} E_3, ..., E_M \in \mathcal{V}^{(*)} \setminus \{E_1, E_2\}, \exists^{K-1} C_2, ..., C_K \in \mathcal{R}^{(*)} \setminus \{C_1\}$, where $\forall (u, c, u') \in [M] \times [K] \times [M], \mathbf{B}_{u,c,u'} = 1 \Rightarrow (\phi \circ E_u, \tau \circ C_c, \phi \circ E_{u'}) \in \phi \circ \tau \circ \mathcal{S}\big\}$.

By definition, we have that for any $(i, k, j) \in \mathcal{S}'$,

$$\Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)}) = \begin{cases} 1 & \text{if } (\phi \circ i, \tau \circ k, \phi \circ j) \in \phi \circ \tau \circ \mathcal{S}' \\ 0 & \text{otherwise,} \end{cases}.$$

Now we show that $(i, k, j) \in \mathcal{S}'$ if and only if $(\phi \circ i, \tau \circ k, \phi \circ j) \in \phi \circ \tau \circ \mathcal{S}'$. If $(i, k, j) \in \mathcal{S}'$, then $E_1 = v^{(*)}_i, E_2 = v^{(*)}_j, C_1 = r^{(*)}_k, \exists^{M-2} E_3, ..., E_M \in \mathcal{V}^{(*)} \setminus \{E_1, E_2\}, \exists^{K-1} C_2, ..., C_K \in \mathcal{R}^{(*)} \setminus \{C_1\}$, such that $\mathbf{B}_{u,c,u'} = 1 \implies (E_u, C_c, E_{u'}) \in \mathcal{S}$. Since $(E_u, C_c, E_{u'}) \in \mathcal{S}$ if and only if $(\phi \circ E_u, \tau \circ C_c, \phi \circ E_{u'}) \in \phi \circ \tau \circ \mathcal{S}$ by definition, we have $(\phi \circ i, \tau \circ k, \phi \circ j) \in \phi \circ \tau \circ \mathcal{S}'$. Similarly we can prove if $(\phi \circ i, \tau \circ k, \phi \circ j) \in \phi \circ \tau \circ \mathcal{S}'$, then $(i, k, j) \in \mathcal{S}'$ with the same reasoning.

In conclusion, for any $\mathbf{A}^{(*)} \in \mathbb{A}$ with number of nodes and relations $N^{(*)}, R^{(*)}$, since $(i, k, j) \in \mathcal{S}'$ if and only if $(\phi \circ i, \tau \circ k, \phi \circ j) \in \phi \circ \tau \circ \mathcal{S}'$, then by definition $\Gamma_{\text{tri}}((\phi \circ i, \tau \circ k, \phi \circ j), \phi \circ \tau \circ \mathbf{A}^{(*)}) = \Gamma_{\text{tri}}((i, k, j), \mathbf{A}^{(*)})$ holds $\forall (i, k, j) \in [N^{(*)}] \times [R^{(*)}] \times [N^{(*)}]$, which proves Γ_{tri} is a double invariant triplet representation (Definition 2.3).

$\square$

D Additional Related Work

Link prediction in discrete attributed multigraphs, which are commonly used to represent relational data in a structured way by indicating different types of relations between pairs of nodes in the graph, involves predicting not only the existence of missing edges but also the associated relation types.

Transductive link prediction. In transductive link prediction, missing links are predicted over a fixed set of nodes and relation types as in training. Traditionally, factorization-based methods [59, 45, 5, 69, 76, 63, 44, 62, 19, 58] have been proposed to obtain latent embedding of nodes and relation types to capture their relative information in the graph. These models try to score all combinations of nodes and relations with embeddings as factors, similar to tensor factorization. Although excellence in transductive tasks, these positional embeddings [54] (a.k.a. permutation-sensitive embeddings) require extensive retraining to perform inductive tasks over new nodes or relations [60]. However, in real-world applications, relational data is often evolving, requiring link prediction over new nodes and new relation types, or even entirely new graphs.

Inductive link prediction over new nodes (but not new relations) with GNN-based model. In recent years, with the advancement of graph neural networks (GNNs) [18, 33, 30, 64, 8, 43], 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 [53], GraIL [60], NodePiece [22], NBFNet [87], ReFactorGNNs [14] etc.. As GNNs are node permutation equivariant [73, 54], these models learn structural node/pairwise representation, which can be used to perform *inductive link prediction over solely new nodes*, while most of the GNN performance are worse than FM-based methods [50, 14]. Specifically, Teru et al. (2020) [60] extends the idea from [83] to use local subgraph representations for discrete attributed multigraph link prediction. Chen et al. (2022b) [14] aims to build the connection between FM and GNNs, where they propose an architecture to cast FMs as GNNs. Galkin et al. (2021) [22] uses anchor-nodes for parameter-efficient architecture for discrete attributed multigraph completion. Zhu et al. (2021) [87] extends the Bellman-Ford algorithm, which learns pairwise representations by all the path representations between nodes. [2] analyzes discrete attributed multigraph-GNNs expressiveness by connecting it with the Weisfeiler-Leman test in discrete attributed multigraph.

Inductive link prediction over new nodes (but not new relations) with logical induction. The relation prediction problem in relational data represented by discrete attributed multigraph can also be considered as the problem of learning first-order logical Horn clauses [76, 77, 51, 60] from the relational data, where one aims to extract logical rules on binary predicates. These methods are inherently node-independent and are able to perform *inductive link prediction over solely new nodes*. Barceló et al. (2020) [3] discusses the connection between the expressiveness of GNNs and first-order logical induction, but only on node GNN representation and logical node classifier. Qiu et al. (2023) [47] further analyzes the logical expressiveness of GNNs for attributed multigraph by showing GNNs are able to capture logical rules from graded modal logic and provides a logical explanation of why pairwise GNNs [84, 87] can achieve SOTA results. In our paper, we try to build the connection between triplet representation and logical Horn clauses. Traditionally, logical rules are learned through statistically enumerating patterns observed in discrete attributed multigraph [34, 21]. Neural LP [77] and DRUM [51] learn logical rules in an end-to-end differentiable manner using the set of logic paths between two nodes with sequence models. Cheng et al. (2022) [15] follows a similar manner, which breaks a big sequential model into small atomic models in a recursive way. Galkin et al. (2022) [23] aims to inductively extract logical rules by devising NodePiece [22] and NBFNet [87]. However, all these methods are not able to deal with new relation types in test.

Inductive link prediction over both new nodes and new relations (with extra context) Few-shot and zero-shot relational reasoning [72, 38, 46, 85, 24, 67, 31, 11, 25] aim to query triplets involving unseen relation types with access to few or zero support triplets of these unseen relation types at test time. Recent methods [46, 85, 31, 25] can even query over unseen nodes. Yet, they often need extra context in the test graph, such as textual descriptions and/or ontological information of the unseen relation types or a shared background graph between the training and test graph, i.e., the test nodes and relation types are connected to the training ones. For instance, zero-shot link prediction methods such as Qin et al. (2020) [46] employ a generative adversarial network [29] to utilize the additional textual information to bridge the semantic gap between seen and unseen relations. Later, Geng et al. (2021) [24] presented an ontology-enhanced zero-shot learning approach that incorporates both ontology structural and textural information. Similarly, TACT [10] aims to model the topological correlations between the target relations and their adjacent relations (assumes there are relations that are seen in train) using a relational correlation network to learn more expressive representations of the target relations. A recent work is RMPI [25] that extracts enclosing subgraphs around the target triplet, which are assumed to contain triplets of some relation types seen in training and uses graph ontology to bridge the unseen relation types to the seen ones. Zhao et al. (2020) [85] uses attention-based GNNs and convolutional transition for link prediction over new nodes and new relations assuming a shared background graph between training and test (i.e., new relations in test are connected with existing nodes and relations in training). MaKEr [12] also uses the local graph structure to handle new nodes and new relation types using a meta-learning framework, assuming the test graph has overlapping relations and entities with the training graph. On the other hand, few-shot relational reasoning methods learn representations of the unseen relation types from the few support triplets, which are generally assumed to connect to existing nodes and relations seen in training [72, 13, 82]. For example, Xiong et al. (2018) [72] was the first to solve the one-shot task by proposing to compute matching scores between the new relation types observed in the support set to those training relation types. Later, Zhang et al. (2020) [82] extends Xiong et al. (2018) [72] by using an attention-based aggregation to take advantage of information from all support triplets. Recently, Huang et al. (2020) [31] proposed a hypothesis testing method that matches the new relation types to the training ones by learning to compare the similarity between the connection subgraph patterns surrounding the target triplets. Another line of research is to solve few-shot relational reasoning via meta-learning. For instance, Chen et al. (2019) [13] updates a meta representation over the relation types, and Lv et al. (2019) [38] adopts MAML [20] to learn meta parameters for frequently occurring relations, which can then be adapted to few-shot relations. All of these few-shot learning methods, however, require that the few-shot triplets are connected to a background graph observed during training in order to learn about the relationship between new relation types and existing ones. Hence, all these methods cannot be directly applied to test graphs that neither contain textual descriptions of the unseen relation types nor triplets involving those relation types seen in training.

Inductive link prediction over both new nodes and new relations (no extra context) In this paper, we focus on the most general task, i.e., inductive link prediction over both new nodes and new relations on entirely new test graphs without textual descriptions, which we call *doubly inductive link prediction*. To the best of our knowledge, InGram [35] is the first and only existing method capable of performing this task. In contrast to Lee et al. (2023) [35] that designed a specific architecture, i.e., InGram, our work proposes a general theoretical framework for designing an entire class of models capable of solving the doubly inductive link prediction task, which encompasses InGram as a specific instantiation. Modeling details of InGram have been substantially discussed in the main paper.

Knowledge graph alignment. Knowledge graph alignment tasks [55, 56, 75, 57] are very common in heterogeneous, cross-lingual, and domain-specific relational data, where the task aims to align nodes among different domains. For example, matching nodes with their counterparts in different languages [70, 74]. It is intrinsically different than our task, where we aim to inductively apply on completely new nodes and relations, possibly with no clear alignments between them.

E Experiments

Our code is available at `https://anonymous.4open.science/r/ISDEA-Fix-B3D7`.

E.1 Doubly inductive link prediction task over both new nodes and new relation types

In this section, we provide more detailed experiment results and analysis for our method on inductively doubly inductive link prediction on both new nodes and new relation types.

Datasets. To the best of our knowledge, there are no existing real-world benchmarks that are specially designed to test a model's extrapolation capability for doubly inductive link prediction task by training the model on one graph and testing it on another completely new graph coming from different domains and distributions. Existing datasets such as NL-100, WK-100, and FB-100 from Lee et al. (2023) [35] are typically created by randomly splitting a larger graph (e.g. NELL-995 [72], Wikidata68K [26], FB15K237 [61]) into disjoint node and relation sets, implying that the test and training graphs still come from the same distribution. In contrast, we purposefully create two doubly inductive link prediction benchmark datasets: PediaTypes and WikiTopics, sampled respectively from the OpenEA library [57] and WikiData-5M [68], where by design the test and training graphs are either from different domains or different topic groups and are likely to possess different characteristics to fully test model's capability for doubly inductive link prediction. We also propose another task with modifications of the NL-k, WK-k, and FB-k datasets from InGram [35] and one synthetic task FD2 to study the expressive power of ISDEA.

E.1.1 Experiment Setup

Baselines. To the best of our knowledge, InGram [35] is the first and only work capable of performing doubly inductive link prediction without needing significant modification to the model. Hence, we chose InGram as one baseline. We also run RMPI [25], which is capable of reasoning over new nodes and new relations but requires extra context at test time (test graphs either contain training relations or ontological information of unseen relations). We simply provide randomized embeddings of unseen relations at test time following Lee et al. (2023) [35]. In addition, we consider the state-of-the-art link prediction model NBFNet [87] capable of generalizing over to new nodes but not new relations and modifying its architecture to work with new relations at test time by providing randomized embeddings of unseen relations at test time following Lee et al. (2023) [35]. We also compare our models with message-passing GNNs including GAT [64], GIN [73], GraphConv [41] which treats the graph as a homogeneous graph by ignoring the relation types. For fair comparisons, we add distance features as in Equation (2) to increase the expressiveness of these GNNs. For training of each single run, we augment each triplet (i, k, j) by its inversion (i, k^{-1}, j), and sample 2 negative (node) triplets (i', k, j') and 2 negative (relation) triplets (i, k', j) per positive in training as Sun et al. (2018) [58] and Zhu et al. (2021) [87]. Training was performed on NVidia A100s, L4s, GeForce RTX 2080 Ti, and TITAN V GPUs.

Evaluation Metrics. We sample 50 negative triplets for each test positive triplet during test evaluation by corrupting either nodes or relation types (Equation (3)), and use Nodes Hits@k and Relation Hits@k separately which counts the ratio of positive triplets ranked at or above the k-th place against the 50 negative samples as evaluation metric over 5 runs. Specifically, for Node prediction evaluation, we sample without replacement 50 negative tail (or head) nodes, and for Relation prediction evaluation, we sample with replacement 50 negative relation types (can also handle cases where the number of test relations is less than 50). We also report other widely used metrics such as MRR.

Hyperparameters and Implementation Details. For homogeneous GNN methods, NBFNet and ISDEA, We follow the same configuration as Teru et al. (2020) [60] such that the hidden layers have 32 neurons. We use Adam optimizer with grid search over learning rate $\alpha \in \{0.01, 0.001, 0.0001\}$, and over weight decay $\beta \in \{0.0005, 0\}$. For all datasets, we train these models for 10 epochs with a mini-batch size of 16. For the GNN kernel (e.g., GraphConv, GIN, GAT) of ISDEA, we choose the best-performing model in validation. For these models, the number of hops and number of layers are 2 on FD-2, and 3 on all other datasets to ensure fair comparison.

Since NBFNet is designed to only perform inductive link prediction with solely new nodes and utilizes trained relation embeddings, we use randomly initialized embeddings for the unseen relation types at test time to enable it for performing doubly inductive link prediction.

To run InGram [35] on PediaTypes and WikiTopics, we conduct hyperparameter search over the configurations of ranking loss margin $\gamma \in \{1.0, 2.0\}$, learning rate $\alpha \in \{0.0005, 0.001\}$, number of entity layers $L \in \{2, 3, 4\}$, and number of entity layers $\hat{L} \in \{2, 3, 4\}$. For other hyperparameters, we use the suggested values from Lee et al. (2023) [35] and their codebase, such as the number of bins $B = 10$ and the number of attention heads $K = 8$. We then use the overall best-performing hyperparameters on PediaTypes and the best-performing hyperparameters on WikiTopics to run InGram on all tasks in PediaTypes and all tasks in WikiTopics respectively. For running on the (modified) NL-k, WK-k, and FB-k datasets from Lee et al. (2023) [35], we use the provided hyperparameters for each task from the authors.

To run DEq-InGram, we use the same trained checkpoints of InGram. The difference is at inference time, where instead of a single forward pass with one sample of randomly initialized entity and relation embeddings for InGram, we draw 10 samples of initial entity and relation embeddings and run 10 forward passes. This yields 10 Monte Carlo samples of the triplet scores, which we then use to compute the DEq-InGram triplet scores according to Equation (4).

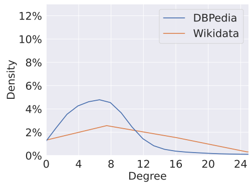
For RMPI [25], we use the provided hyperparameters from the codebase and run the RMPI-NE version of the model with a concatenation-based fusion function, which generally has the best performance reported in Geng et al. (2023) [25]. We note that, since our attributed multigraph does not contain ontological information over the unseen relation types of the test graphs, we instead provide the model with randomly initialized embeddings for the unseen relation types to perform doubly inductive link prediction.

E.1.2 Doubly inductive link prediction over PediaTypes

As discussed in Section 5, we create our own doubly inductive link prediction benchmark dataset PediaTypes. Each graph in PediaTypes is sampled from a graph in the OpenEA library [57] (under GPL-3.0 license). OpenEA [57] library provides multiple pairs of attributed multigraph, each pair of which is a database containing similar topics. Each node of a graph corresponds to the Universal Resource Identifier (URI) of an entity in the database, e.g., "*http://dbpedia.org/resource/E399772*" from English DBPedia. Each relation type of a graph corresponds to the URI of a relation in the database, e.g., "*http://dbpedia.org/ontology/award*" from English DBPedia. Moreover, since each pair of graphs describes similar topics, most entities and relations are highly related, e.g., "*http://dbpedia.org/resource/E678522*" from English and "*http://fr.dbpedia.org/resource/E415873*" from French are indeed the same thing, except that the labeling is different. Thus, we would expect a powerful model that is insensitive to node and relation type labelings to be able to learn on one graph of the pair and perform well on the other graph of the same pair.

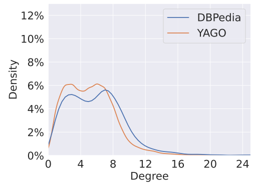
To control the size under a feasible limitation, we use the same subgraph sampling algorithm as GraIL [60], which proposes link prediction benchmarks over solely new nodes. Details are provided in Algorithm 1. For each pair of graphs from the OpenEA library, e.g., English-to-French DBPedia, we first apply the sampling algorithm as in Algorithm 1 on each graph to reduce the size of each graph. Then we randomly split querying triplets given by the Algorithm 1 into 80% training, 10% validation, and 10% test for each graph. Finally, to construct the task where we learn on English DBPedia but test on French DBPedia (denoted as EN-FR), we pick training and validation triplets from the English graph for model tuning, and only use test triplets from the French graph for model evaluation; Similarly, for task from French to English (FR-EN), we pick training and validation triplets from French graph for model tuning, and only use test triplets from English graph for model evaluation. The **dataset statistics** for PediaTypes are summarized in Figure 6.

	DBPedia	Wikidata
#Nodes	4906	4948
#Relations	144	102
#Triplets (Obv.)	17593	23888
#Triplets (Qry.)	1666	2456
#Avg. Deg.	7.85	10.65



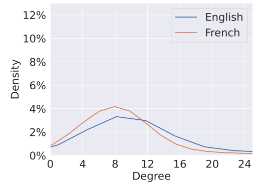
(a) DB⟷WD

	DBPedia	YAGO
#Nodes	4795	4751
#Relations	64	17
#Triplets (Obv.)	13248	11327
#Triplets (Qry.)	1177	973
#Avg. Deg.	6.02	5.18



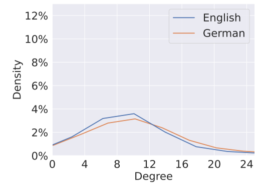
(b) DB⟷YG

	English	French
#Nodes	4962	4933
#Relations	122	101
#Triplets (Obv.)	30876	24165
#Triplets (Qry.)	3326	2485
#Avg. Deg.	13.79	10.81



(c) EN⟷FR

	English	German
#Nodes	4890	4915
#Relations	121	67
#Triplets (Obv.)	25177	29011
#Triplets (Qry.)	2626	3100
#Avg. Deg.	11.37	13.07



(d) EN⟷DE

Figure 6: **Statistics of PediaTypes:** We report graph statistics including the number of nodes, number of relations, observed (obv.) triplets, querying (qry.) triplets, and average degree for each graph pair, e.g., (a) corresponds to DBPedia-and-Wikidata pair, and will be used to construct DB2WD and WD2DB tasks. We also report (in & out) degree distribution on each graph at the bottom. We omit tail distribution larger than 25 since they are too small and almost flat.

Algorithm 1 Sampling Algorithm for PediaTypes. This is a subgraph sampling code for a single graph (either training or test). It will reduce the large original graph into a connected graph of the required size.

Require: Raw graph triplets $\mathcal{S}^{\text{raw}}$, Raw graph node set $\mathcal{V}^{\text{raw}}$, Raw graph relation set $\mathcal{R}^{\text{raw}}$, Maximum number of nodes N, Maximum number of edges M, Maximum node degree D.
Ensure: Subgraph triplets $\mathcal{S}^{\text{sub}}$
$\mathcal{S}^{\text{sub}} \leftarrow \emptyset$
$\mathcal{V}^{\text{sub}} \leftarrow \emptyset$
$\mathcal{R}^{\text{sub}} \leftarrow \emptyset$
Create an empty queue Q.
Get the node v_0 with the highest degree in the raw graph.
Q.add(v_0)
$\mathcal{V}^{\text{sub}} \leftarrow \mathcal{V}^{\text{sub}} \cup \{v_0\}$
while $|Q| > 0$ **do**
  $u \leftarrow Q$.pop()
  if $|\mathcal{V}^{\text{sub}}| \geq N$ or $|\mathcal{V}^{\text{sub}}| \geq M$ **then**
    continue
  end if
  $\mathcal{B} = \{(v, r, u) | (r, v) \in \mathcal{R}^{\text{raw}} \times \mathcal{V}^{\text{raw}}\} \cup \{(u, r, v) | (r, v) \in \mathcal{R}^{\text{raw}} \times \mathcal{V}^{\text{raw}}\}$
  if $|\mathcal{B}| > D$ **then**
    Uniformly select D triplets from $\mathcal{B}$ as $\mathcal{B}'$
  else
    $\mathcal{B}' \leftarrow \mathcal{B}$
  end if
  for $(i, r, j) \in \mathcal{B}'$ **do**
    if $i = u$ **then**
      Q.add(j)
      $\mathcal{V}^{\text{sub}} \leftarrow \mathcal{V}^{\text{sub}} \cup \{j\}$
    else
      Q.add(i)
      $\mathcal{V}^{\text{sub}} \leftarrow \mathcal{V}^{\text{sub}} \cup \{i\}$
    end if
    $\mathcal{S}^{\text{sub}} \leftarrow \mathcal{S}^{\text{sub}} \cup \{(i, r, j)\}$
  end for
end while

Table 2: **Relation & Node MRR performance on Doubly Inductive Link Prediction over PediaTypes.** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. "Rand" column contains unbiased estimations of the performance from a random predictor. **Both ISDEA and DEq-InGram consistently achieve better results than the baselines.** N/A*: Not available due to constant crashes.

(a) **Relation prediction $(i, ?, j)$ performance in %. Higher $\uparrow$ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00
GAT	8.04±00.25	7.93±00.04	8.17±00.08	8.12±00.09	8.06±00.15	7.90±00.12	8.12±00.21	8.17±00.16
GIN	8.07±00.09	8.09±00.05	8.07±00.13	8.07±00.11	8.03±00.20	7.97±00.30	7.82±00.27	7.84±00.14
GraphConv	7.92±00.16	7.97±00.12	8.07±00.15	8.03±00.05	8.14±00.04	7.98±00.18	8.04±00.24	7.84±00.13
NBFNet	10.25±01.24	9.53±00.85	8.15±01.21	4.32±00.26	10.33±02.45	8.97±01.24	9.29±01.38	14.54±04.76
RMPI	12.45±01.90	12.10±02.71	11.69±04.37	10.28±01.28	N/A*	8.54±02.70	17.89±12.22	6.53±02.16
InGram	50.03±05.32	26.31±08.27	21.32±07.84	29.81±14.21	48.70±10.06	38.81±03.10	29.94±13.28	32.26±13.97
DEq-InGram (Ours)	**73.38**±05.77	41.61±10.12	46.86±09.11	40.56±14.80	**80.74**±04.47	66.06±02.91	39.51±16.76	49.10±05.43
ISDEA (Ours)	70.06±02.01	**69.01**±00.57	**78.38**±04.04	**88.82**±00.28	65.89±04.71	**72.57**±00.73	**75.88**±01.58	**74.04**±00.47

(b) **Node prediction $(i, k, ?)$ performance in %. Higher $\uparrow$ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00	8.86±00.00
GAT	51.43±00.25	49.48±01.51	26.22±00.44	25.45±01.23	16.87±00.59	34.66±00.33	37.22±00.29	45.96±00.29
GIN	53.72±03.45	52.03±03.38	34.60±07.43	37.27±09.42	20.75±07.22	40.37±08.20	35.80±01.36	44.77±00.92
GraphConv	63.72±01.76	57.77±01.09	48.18±00.96	45.18±00.15	22.49±00.76	50.30±02.80	38.71±00.55	50.54±00.42
NBFNet	69.22±02.44	**74.01**±01.41	63.49±02.44	38.86±02.55	41.26±02.58	64.02±01.25	38.13±01.11	52.30±02.09
RMPI	63.02±02.94	43.72±05.65	44.82±02.93	46.84±05.36	N/A*	46.33±08.76	**43.00**±03.70	53.72±01.84
InGram	71.23±01.73	55.67±05.65	55.94±02.76	61.15±01.42	34.50±08.47	57.05±03.73	26.36±04.73	**56.23**±01.56
DEq-InGram (Ours)	**78.45**±00.89	68.59±04.30	**66.13**±01.48	**70.32**±01.58	44.71±08.98	**69.23**±02.53	35.67±03.92	48.07±08.76
ISDEA (Ours)	53.92±00.26	57.68±00.68	50.30±02.08	51.33±00.40	**45.75**±00.66	51.64±00.60	41.72±01.64	48.21±01.06

Additional Results We present the Node & Relation Hits@10 performance in the main paper. We provide more results including MRR, Hits@1, Hits@5 in Tables 2 to 4. We can see that our proposed ISDEA and DEq-InGram perform consistently and significantly better than the baselines in the much harder relation prediction task, showing their power to generalize to both new nodes and new relations. The structural double equivariant model ISDEA performs worse on node prediction over some datasets, which might be due to the node GNN implementation of ISDEA. These tasks do not care much about the actual relation type as we can see from the superior performance of homogeneous GNNs on node prediction. So the additional equivariance over relations and the training loss over both negative nodes and negative relations might cause the model to focus more on the relation prediction task, while the double equivariant structural representation might hurt the performance of missing node prediction [54].

But it is important to note that the structural double equivariant ISDEA model excels on relation prediction and achieves much better results on Hits@1 and Hits@5 as shown in Tables 3 and 4. The performance of baseline models that is lower than random is probably because the knowledge they learn from one dataset is not able to correctly transform to another dataset, while our double equivariant model architecture is able to perform this hard doubly inductive link prediction over both new nodes and new relation types. We also note that in the Hits@1 and Hits@5 Tables 3 and 4, there are cases where DEq-InGram has higher variances than the original InGram while achieving much better average performance. This is because due to the random initialization, InGram performs poorly on the much harder Hits@1 and Hits@5 performance compared to Hits@10. In some seeds of the runs, DEq-InGram successfully improves the performance of InGram, but there are still seeds of runs that DEq-InGram still performs similar to InGram. Thus, it results in DEq-InGram having much better average results while also with higher standard deviations.

E.1.3 Doubly inductive link prediction over WikiTopics

As discussed in Section 5.2, the WikiTopics dataset is created from the WikiData-5M [68] (under CC0 1.0 license). Each node in the graphs of this dataset represents an entity described by an existing Wikipedia page, and each relation type corresponds to a particular relation between the entities, such as "director of" or "designed by". The node and relation type indices are codenames that start

Table 3: **Relation & Node Hits@1 performance on Doubly Inductive Link Prediction over PediaTypes.** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. “Rand” column contains unbiased estimations of the performance from a random predictor. **Both ISDEA and DEq-InGram consistently achieve better results than the baselines.** N/A*: Not available due to constant crashes.

(a) **Relation prediction $(i, ?, j)$ performance in %. Higher ↑ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00
GAT	1.07±00.14	1.01±00.01	1.03±00.03	1.11±00.09	1.07±00.14	0.99±00.21	0.96±00.16	1.09±00.25
GIN	1.01±00.03	0.95±00.08	1.03±00.06	1.10±00.06	0.96±00.15	1.00±00.15	0.92±00.15	0.83±00.17
GraphConv	0.91±00.03	0.97±00.06	1.05±00.14	1.01±00.03	1.09±00.07	0.91±00.04	0.94±00.22	0.88±00.20
NBFNet	4.43±01.24	3.62±01.01	2.49±01.23	0.51±00.18	4.18±02.17	2.80±00.83	1.63±00.89	7.30±05.01
RMPI	3.92±02.08	4.04±01.83	3.37±02.20	2.13±00.79	N/A*	2.39±02.35	7.36±09.03	0.91±00.92
InGram	35.19±07.73	12.40±07.55	8.45±06.57	16.46±16.33	33.66±12.09	25.69±03.88	14.24±12.00	15.83±12.59
DEq-InGram (Ours)	**65.26**±10.23	26.90±12.97	36.80±11.16	25.34±18.48	**75.00**±06.42	60.35±02.56	24.28±14.29	30.82±10.43
ISDEA (Ours)	61.46±00.79	**58.18**±00.14	**68.00**±06.41	**84.83**±00.29	57.51±05.40	**62.72**±01.24	**69.12**±02.40	**66.68**±00.81

(b) **Node prediction $(i, k, ?)$ performance in %. Higher ↑ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00	1.96±00.00
GAT	31.80±00.64	30.19±02.30	10.23±00.96	8.68±01.69	7.98±00.89	16.26±00.34	26.09±00.47	33.06±00.29
GIN	34.59±04.64	34.57±05.26	17.69±07.91	20.74±10.01	12.42±06.59	23.10±09.67	23.72±01.62	32.26±01.89
GraphConv	47.48±02.60	40.37±01.52	31.96±01.02	28.46±00.13	12.53±00.34	35.82±03.54	24.12±00.80	37.05±00.51
NBFNet	64.17±02.68	**69.68**±01.63	**57.50**±02.66	32.26±02.81	34.56±02.54	**59.70**±01.38	**33.32**±01.11	**47.47**±02.08
RMPI	48.27±03.74	26.92±04.87	27.38±03.09	29.60±04.77	N/A*	34.81±08.97	33.29±03.20	42.14±02.87
InGram	60.00±02.06	41.59±06.37	39.05±02.99	45.44±01.69	22.06±08.10	42.54±04.50	13.47±03.50	20.09±04.96
DEq-InGram (Ours)	**69.46**±01.12	57.65±05.54	51.93±01.88	**57.06**±01.96	32.12±09.51	57.84±03.28	20.49±03.35	33.01±08.87
ISDEA (Ours)	43.03±00.25	47.38±00.28	35.41±02.25	37.12±00.31	**35.59**±00.73	40.56±01.72	27.70±01.95	35.29±01.67

with the prefix “Q” and “P” respectively, which are devoid of semantic meaning. Nevertheless, WikiData-5M [68] provides aliases for all nodes and relation types that map their indices to textual descriptions, and we use these textual descriptions to group the relation types into 11 different topics (we do not however provide these textual descriptions to the models per the specification of the doubly inductive link prediction task). In total, WikiData-5M [68] contains 822 relation types. We create WikiTopics datasets from all 822 relation types, which comprise graphs with as many as 66 relation types. Each graph has a disjoint set of relation types from all other graphs. Below is a list of all 11 topics:

- T1: Art and Media Representation
- T2: Award Nomination and Achievement
- T3: Education and Academia
- T4: Health, Medicine, and Genetics
- T5: Infrastructure and Transportation
- T6: Location and Administrative Entity
- T7: Organization and Membership
- T8: People and Social Relationship
- T9: Science, Technology, and Language,
- T10: Sport, and Game Competition
- T11: Taxonomy and Biology

To control the overall size of the graphs in WikiTopics, we downsample $10,000$ nodes for each topic from the subgraph consisting of only the triplets with the relation types belonging to that topic. We adopt the Forest Fire sampling procedure with burning probability $p = 0.8$ [37] implemented in the Little Ball of Fur Python package [49]. We then split the downsampled topic graph into 90% observable triplets and 10% querying triplets to be predicted by the models. When splitting, we ensure that the set of nodes in the querying triplets is a subset of those in the observable triplets. This

Table 4: **Relation & Node Hits@5 performance on Doubly Inductive Link Prediction over PediaTypes.** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. "Rand" column contains unbiased estimations of the performance from a random predictor. **Both ISDEA and DEq-InGram consistently achieve better results than the baselines.** N/A*: Not available due to constant crashes.

(a) **Relation prediction $(i, ?, j)$ performance in %. Higher $\uparrow$ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00
GAT	9.08±00.39	8.63±00.25	9.47±00.18	9.20±00.24	8.95±00.36	8.63±00.29	9.58±00.50	9.16±00.23
GIN	9.09±00.16	9.31±00.15	9.18±00.28	9.23±00.34	9.12±00.12	8.85±00.56	8.53±00.66	8.61±00.34
GraphConv	8.97±00.66	8.74±00.26	9.23±00.11	8.82±00.10	9.17±00.29	9.11±00.50	9.01±00.72	8.73±00.15
NBFNet	12.94±01.77	12.46±01.40	8.56±01.67	2.68±00.72	13.44±04.02	11.74±03.02	11.95±03.78	20.37±05.90
RMPI	16.39±04.15	15.76±04.58	15.86±08.05	12.56±02.70	N/A*	8.91±03.51	24.25±19.24	4.98±03.08
InGram	67.15±05.04	37.86±14.41	30.99±11.82	40.00±13.02	65.80±09.59	51.66±03.57	43.27±19.30	51.54±26.09
DEq-InGram (Ours)	**83.23**±05.64	59.83±11.57	54.30±08.25	57.65±15.74	**87.08**±02.55	70.79±03.80	51.45±29.14	75.85±07.26
ISDEA (Ours)	82.11±04.01	**83.19**±01.73	**92.39**±00.83	**93.59**±00.53	75.95±03.89	**86.10**±01.26	**85.80**±01.23	**83.36**±01.55

(b) **Node prediction $(i, k, ?)$ performance in %. Higher $\uparrow$ is better.**

Models	EN-FR	FR-EN	EN-DE	DE-EN	DB-WD	WD-DB	DB-YG	YG-DB
Rand	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00	9.80±00.00
GAT	78.49±00.44	74.70±00.68	42.17±00.91	42.39±00.52	20.96±00.65	57.26±00.89	46.92±00.37	59.20±00.41
GIN	79.96±01.88	74.33±01.16	53.97±07.61	55.89±10.06	25.05±09.23	61.94±06.71	46.56±01.37	57.48±00.35
GraphConv	85.21±00.63	80.67±00.30	67.76±01.19	64.97±00.43	28.37±01.41	67.36±02.37	53.79±00.72	64.13±00.23
NBFNet	81.48±02.24	**85.15**±01.06	77.62±02.41	48.73±02.59	51.52±03.21	72.18±00.90	44.01±01.40	60.34±02.28
RMPI	82.47±02.25	64.88±07.62	67.24±04.38	69.47±06.60	N/A*	60.11±08.77	51.57±05.03	**66.67**±01.28
InGram	85.15±01.74	72.32±05.31	78.84±02.86	81.01±00.97	45.96±11.09	74.88±03.09	37.49±06.84	50.66±06.76
DEq-InGram (Ours)	**89.62**±00.63	81.54±02.82	**84.57**±00.95	**87.16**±01.04	**57.44**±09.14	**83.14**±01.64	51.77±05.14	65.33±09.57
ISDEA (Ours)	64.45±00.24	67.24±01.32	68.80±01.90	68.20±00.53	54.83±00.90	62.60±02.55	**55.21**±01.07	61.87±01.30

way, the model is not tasked with the impossible task of predicting relation types between orphaned nodes previously unseen in the observable part of the graph. This is implemented via an iterative procedure, where we first sample a batch of missing triplets from the downsampled topic graph, then discard those that contain unseen nodes in the rest of the triplets, and repeat this process until the number of sampled triplets reaches 10% of total triplets. Figure 7 shows the **data statistics** of WikiTopics dataset.

Additional experiment results on WikiTopics. In Section 5.2, we only provide heatmaps of Relation Hits@10 Performance WikiTopics due to space limit. We present more detailed results (heatmaps with values) of Node and Relation Hits@10, Hits@1, and MRR for WikiTopics in Figures 8 and 9. Due to the large number of runs ($11 \times 10 = 110$ different train-test scenarios, each with 5 random seeds, resulting in a total of 550 runs) and the time constraints to run all baseline models, we perform the evaluation over only the three models (ISDEA, DEq-InGram, and InGram) that are designed for our doubly inductive link prediction task. Figure 8 shows that for the task of predicting missing relation types $(i, ?, j)$, ISDEA and DEq-InGram are consistently better than InGram across all different metrics. Especially, the structural double equivariant ISDEA model exhibits more consistent results across different train-test scenarios than both DEq-InGram and InGram, and achieves significantly better results in Hits@1 and MRR, showcasing its ability for doubly inductive link prediction in a much harder evaluation scenario. For the task of prediction missing nodes $(i, k, ?)$ as shown in Figure 9, ISDEA, DEq-InGram, and InGram showcase comparable performance, whereas ISDEA exhibits more consistent results across different train-test scenarios than both DEq-InGram and InGram. We also note that similar to the relation prediction task, ISDEA also exhibits the best performance in the Hits@1 metric for the node prediction task.

E.1.4 Doubly Inductive Link Prediction over datasets from InGram (Lee et al., 2023)

Lee et al. (2023) [35] proposed the NL-k, WK-k, and FB-k benchmarks originally used to evaluate InGram's performance of reasoning over new nodes and new relation types at test time, where $k \in \{25, 50, 75, 100\}$ means that, in the test graphs, approximately $k\%$ of triplets have unseen relations. For example, the test graph of WK-100 does not contain any training relations and thus induces a doubly inductive link prediction task. Hence, we run our models (ISDEA and DEq-InGram)

	#Nodes	# Relations	#Triplets (Obv.)	#Triplets (Qry.)	Avg. Deg.
Art	10000	45	28023	3113	6.23
Award	10000	10	25056	2783	5.57
Education	10000	15	14193	1575	3.15
Health	10000	20	15337	1703	3.41
Infrastructure	10000	27	21646	2405	4.81
Location	10000	35	80269	8918	17.84
Organization	10000	18	30214	3357	6.71
People	10000	25	58530	6503	13.01
Science	10000	42	12516	1388	2.78
Sport	10000	20	46717	5190	10.38
Taxonomy	10000	31	19416	2157	4.32

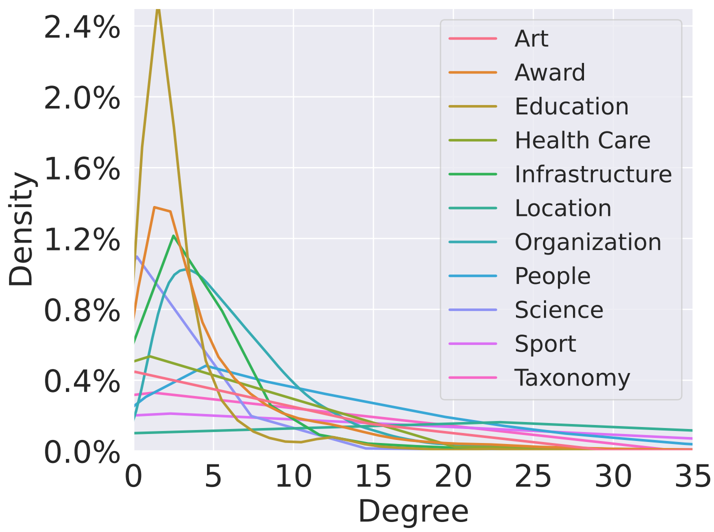
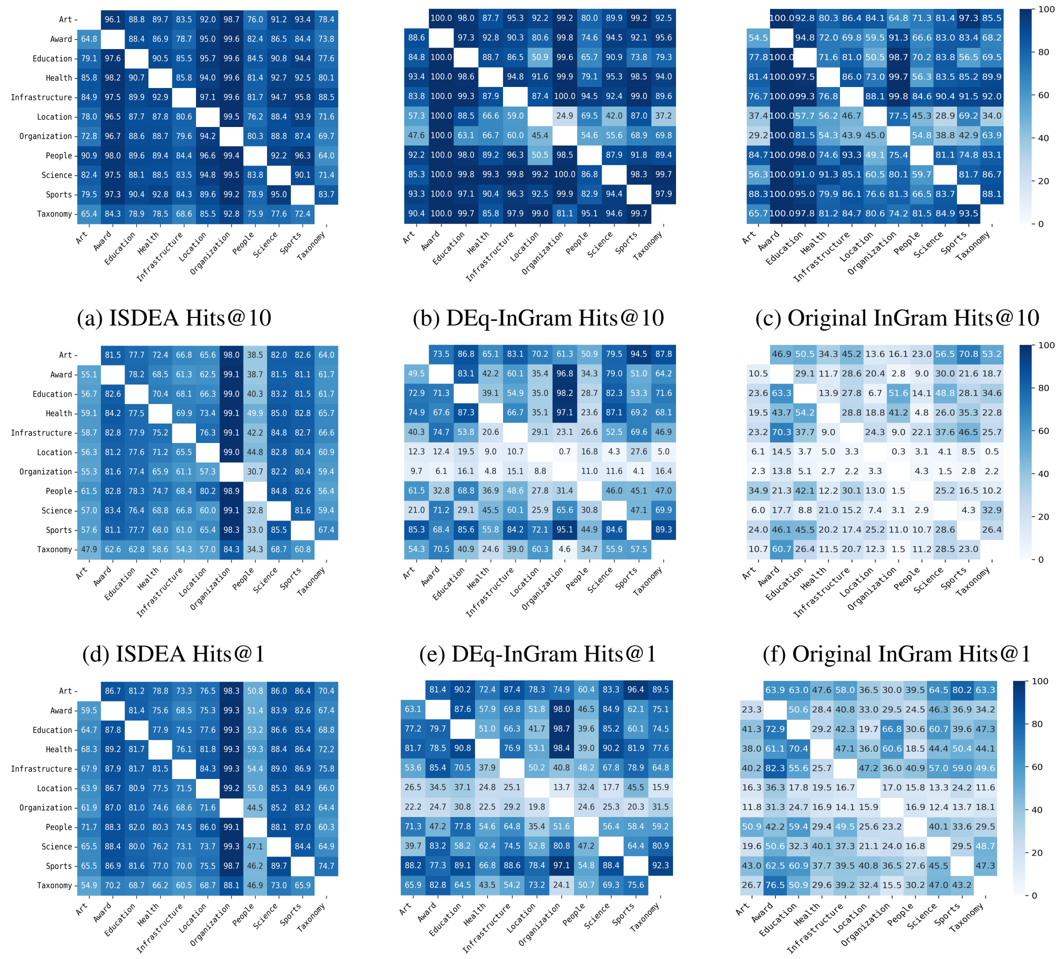


Figure 7: **Statistics of WikiTopics:** We report graph statistics including the number of nodes, number of relations, observed (obv.) triplets, querying (qry.) triplets, and average degree for each graph. We also report (in & out) degree distribution on each graph at the bottom. We omit tail distribution larger than 35 since they are fairly small and almost flat.

against InGram on these benchmarks with results shown in Table 5. We note that, however, due to the different experimental settings (as we discuss next), our results reported in Table 5 are not directly comparable to those reported in Lee et al. (2023) [35], even though they are experimented on essentially the same datasets.

Difference to the original data split and evaluation in InGram [35]: Different from Lee et al. (2023) [35], which uses part of the test graph as the validation set to conduct model hyperparameter search, *our experiments consider a harder setting where the relations in test are not observed in the validation data*. Hence, to modify the NL-k, WK-k, and FB-k datasets to our setting, we discard the original validation set and instead split the original training set into a new set of training and validation triplets with a ratio of 9:1. During training, the models perform self-supervised masking over the training set of triplets to create the training-time observable triplets and training-time target triplets. During validation, the entire set of the new training triplets is taken as the validation-time observable triplets, and the new validation triplets are the target triplets to predict. In addition, Lee et al. (2023) [35] evaluate their model's node prediction performance against *all* nodes in the graph. For efficiency reasons, we evaluate the model performance by sampling without replacement 50 negative nodes for the node prediction task and sampling with replacement 50 negative relation types for the relation prediction task.

Table 5 shows the results, where we can see that ISDEA outperforms InGram on most datasets on the relation prediction task and has smaller standard deviation in general, and DEq-InGram consistently outperforms InGram on all datasets for both relation prediction and node prediction tasks. Importantly, in the dataset FB-100 which follows our doubly inductive link prediction setting with completely new nodes and new relation types in the test with the largest number of training and test relations



(a) ISDEA Hits@10 (b) DEq-InGram Hits@10 (c) Original InGram Hits@10

(d) ISDEA Hits@1 (e) DEq-InGram Hits@1 (f) Original InGram Hits@1

(g) ISDEA MRR (h) DEq-InGram MRR (i) Original InGram MRR

Figure 8: **Relation prediction** $(i, ?, j)$ **performance over WikiTopics** for ISDEA, DEq-InGram, and InGram [35]. Each row within each heatmap corresponds to a training graph, and each column within each heatmap corresponds to a test graph. A darker color means better performance. **Both ISDEA, DEq-InGram perform significantly better than InGram, especially for Hits@1 and MRR, whereas ISDEA exhibits more consistent results across different train-test scenarios than both DEq-InGram and InGram.**

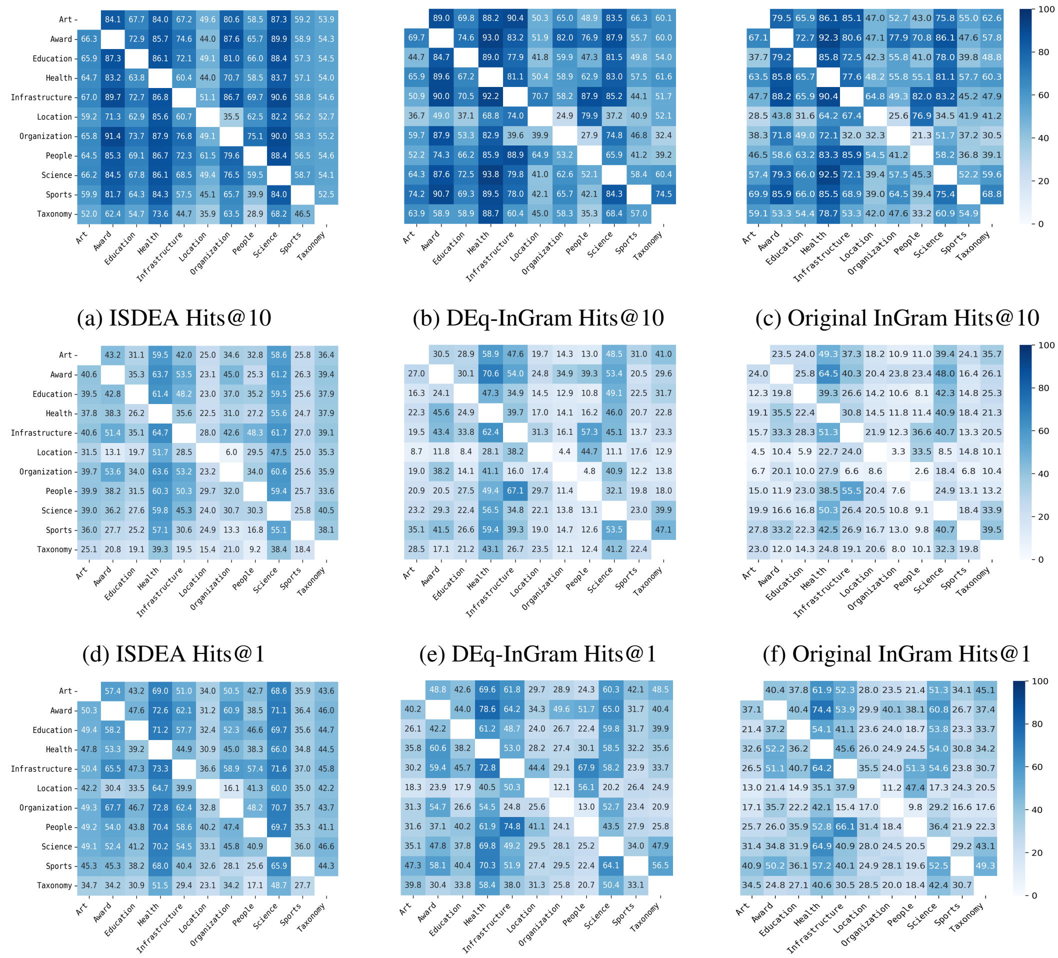
(134 in train and 77 in test) [35], ISDEA achieves significant better results in the relation perdiction task, showcasing its ability for doubly inductive link prediction.

E.1.5 A Synthetic Case Study for ISDEA

To further understand the expressive power and limitations of our proposed sturctural double equivariant model ISDEA, we create **FD-2** to empirically justify the expressivity of our proposal on tasks over both new nodes and new relation types. On FD-2, training has 127 nodes and 2 relations, while test has 254 nodes and 4 relations (more nodes and more relations).

FD-2 is constructed by only a single rule, $(E_1, R_1, E_3) \wedge (E_3, R_2, E_2) \Rightarrow (E_1, R_1, E_3)$ where E_1, E_2, E_3 and R_1, R_2 are all variables. As illustrated in Figure 10, The training data has only two relation types $\{r_1, r_2\}$, while test data has four relation types $\{r_3, r_4, r_5, r_6\}$ which are all different from training relations. For all relation types, only r_1, r_3, r_4 can be used for R_1 assignments, and only r_2, r_5, r_6 can be use for R_2 assignments. Besides, training and test also have distinct node sets.

Each graph (training or test) is consisted by one or more tree-like structures as left side of Figure 10. In each tree-like structure, all solid edges are used as observations, and will form a complete binary



(a) ISDEA Hits@10 (b) DEq-InGram Hits@10 (c) Original InGram Hits@10

(d) ISDEA Hits@1 (e) DEq-InGram Hits@1 (f) Original InGram Hits@1

(g) ISDEA MRR (h) DEq-InGram MRR (i) Original InGram MRR

Figure 9: **Node prediction** $(i, k, ?)$ **performance over WikiTopics** for ISDEA, DEq-InGram, and InGram [35]. Each row within each heatmap corresponds to a training graph, and each column within each heatmap corresponds to a test graph. A darker color means better performance. **ISDEA, DEq-InGram, and InGram showcase comparable performance in general, and ISDEA exhibits the best performance on Hits@1 in particular.**

tree; while all dashed edges are used as training, validation or test samples which are built by applying the only rule over all observed edges. In training, we have only one tree-like structure; while in test, we have two disconnected tree-like structures. A more detailed generation algorithm for a graph given depths of all tree-like structures is provided in Algorithm 2.

Since the structure of FD-2 does not satisfy the requirement of the spanning tree algorithm used in InGram [35], we are not able to apply InGram and DEq-InGram on FD-2. So we provide the results on FD-2 in Table 6 with all remaining baselines and ISDEA. We can see that ISDEA clearly perform better than other baselines, especially in the relation prediction task, and shows capability to perform accurately on the doubly inductive link prediction over both new nodes and new relation types, while methods like NBFNet and RMPI are not able to correctly perdict this task, even for node prediction.

E.1.6 Expressivity Limitation Case Study with FD-2 for ISDEA

We now provide a FD-2 variant where we show that double equivariant representation is not expressive enough to solve a specific task. It is a simple 2-depth tree structure as shown in Figure 11. We denote node representations given by arbitrary double equivariant representation as $H_{v,r}$ where $v \in [1, 7]$ and

Table 5: **Relation & Node Hits@10 performance on Doubly Inductive Link Prediction over NL-k, WK-k, and FB-k of Lee et al. (2023) [35].** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. The best values are shown in bold font, while the second-best values are underlined. **ISDEA outperforms InGram on most datasets on the relation prediction task, and DEq-InGram consistently outperforms InGram on all datasets for both relation prediction and node prediction tasks.**

(a) **Performance in % on NL-k datasets. Higher ↑ is better.**

	Relation prediction $(i, ?, j)$				Node prediction $(i, k, ?)$			
Models	NL-25	NL-50	NL-75	NL-100	NL-25	NL-50	NL-75	NL-100
InGram	64.54±16.86	64.54±12.56	80.16±04.43	70.84±08.52	89.95±02.01	92.74±00.52	95.40±01.38	88.20±01.92
DEq-InGram (Ours)	**83.58**±17.57	**91.32**±05.60	**96.01**±01.23	**87.52**±10.39	**95.03**±00.32	**96.02**±00.34	**97.94**±00.34	**93.80**±01.38
ISDEA (Ours)	69.49±05.71	76.23±06.92	76.03±03.31	80.84±07.35	73.74±03.35	75.76±03.52	77.27±03.80	72.81±04.41

(b) **Performance in % on WK-k datasets. Higher ↑ is better.**

	Relation prediction $(i, ?, j)$				Node prediction $(i, k, ?)$			
Models	WK-25	WK-50	WK-75	WK-100	WK-25	WK-50	WK-75	WK-100
InGram	58.76±13.91	84.01±03.30	80.19±04.19	58.20±11.13	76.99±07.72	70.93±02.38	78.85±04.65	66.29±03.70
DEq-InGram (Ours)	**81.06**±22.31	**94.85**±00.85	**95.84**±01.54	**81.83**±10.10	**87.91**±05.68	**82.58**±01.70	**89.10**±02.15	**79.69**±03.07
ISDEA (Ours)	79.49±06.88	81.25±07.02	84.92±06.86	79.70±07.68	58.28±23.68	73.24±00.57	76.19±01.04	71.76±01.85

(c) **Performance in % on FB-k datasets. Higher ↑ is better.**

	Relation prediction $(i, ?, j)$				Node prediction $(i, k, ?)$			
Models	FB-25	FB-50	FB-75	FB-100	FB-25	FB-50	FB-75	FB-100
InGram	68.26±08.27	50.41±08.79	79.51±02.69	40.46±12.21	86.79±00.70	73.32±06.64	86.57±00.69	71.72±06.93
DEq-InGram (Ours)	82.89±03.58	76.65±04.05	**89.70**±01.14	46.88±15.76	**92.39**±00.30	**81.08**±06.98	**92.14**±00.43	77.54±06.36
ISDEA (Ours)	**83.63**±05.66	**78.70**±05.90	81.27±07.24	**85.41**±04.43	75.93±00.49	69.90±00.81	73.45±01.17	**79.70**±00.81

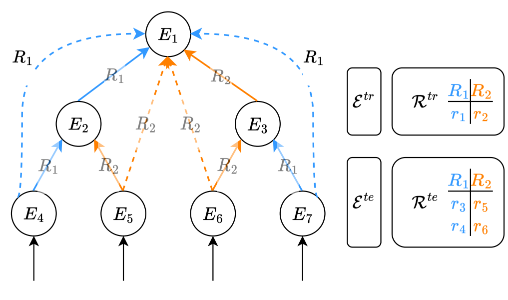


Figure 10: **Synthetic Example of FD-2:** Training and test has their own node and relation type sets: $\mathcal{V}^{(\mathrm{tr})} \cap \mathcal{V}^{(\mathrm{te})} = \emptyset$ and $\mathcal{R}^{(\mathrm{tr})} \cap \mathcal{R}^{(\mathrm{te})} = \emptyset$.

$r \in [1, 4]$. We can easily notice that e_4 and e_7 are symmetric, e_5 and e_6 are symmetric (simply flipping blue and orange colors), thus we will expect $H_{4,1} = H_{7,2}, H_{4,2} = H_{7,1}, H_{5,1} = H_{6,2}, H_{5,2} = H_{6,1}$.

Since there is no r_3 and r_4 in observation, they are freely exchangeable with each other, thus we will also expect

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

Table 6: **Relation & Node performance on Doubly Inductive Link Prediction over FD2.** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. The best values are shown in bold font, while the second-best values are underlined. ISDEA consistently achieve better results than the baselines, especially in the Relation perdiction task. NA* due to the fact that FD2 does not satisfy the spanning tree algorithm used in InGram [35].

	Relation prediction $(i, ?, j)$				Node prediction $(i, k, ?)$			
Models	MRR	Hits@1	Hits@2	Hits@4	MRR	Hits@1	Hits@2	Hits@4
GAT	7.61±00.71	0.77±00.39	2.78±00.80	5.85±00.95	84.62±02.64	71.61±04.94	<u>93.51</u>±01.03	99.72±00.27
GIN	8.44±00.40	1.29±00.37	3.51±00.58	<u>7.18</u>±01.01	73.99±09.60	65.73±06.58	76.69±12.44	81.45±15.80
GraphConv	7.88±00.45	0.81±00.29	2.62±00.61	6.98±01.09	<u>85.95</u>±00.77	<u>74.52</u>±01.81	92.66±01.02	**99.84**±00.15
RMPI	<u>9.09</u>±03.18	<u>1.94</u>±01.88	<u>3.95</u>±04.43	7.10±05.58	21.16±05.85	9.84±05.04	16.74±06.50	27.98±09.96
NBFNet	6.39±02.19	1.50±02.49	1.79±02.39	2.91±02.22	21.95±04.14	14.44±04.34	18.61±04.29	26.47±04.24
InGram	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*
DEq-InGram (Ours)	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*
ISDEA (Ours)	**44.39**±12.17	**32.82**±12.69	**38.71**±13.60	**50.73**±14.04	**90.98**±03.55	**83.59**±06.22	**95.69**±02.34	<u>99.72</u>±00.27

Algorithm 2 Synthesis Algorithm for FD-2. This is triplet generation code for a single graph (either training and test). It will provide observation and query triplets. For training, query triplets are further divided into training and validation triplets; For test, query triplets directly become test triplets.

Require: Tree depth $\{D_1, \dots, D_M\}$, Node Labeling "Names$^{\text{nd}}$", Relation Type Labeling "Names$^{\text{rl}}$".
Ensure: Observation triplets $\mathcal{S}$, Query triplets $\mathcal{Q}$

$\mathcal{S} = \emptyset$
$\mathcal{Q} = \emptyset$
$n \leftarrow 0$
for $m \leftarrow 1, \dots, M$ **do**
  for $d \leftarrow 1, \dots, D_m$ **do**
    for $v \leftarrow 2^d - 1, \dots, 2^{d+1} - 2$ **do**
      $u_1 \leftarrow \lceil (v-2)/2 \rceil$
      $u_2 \leftarrow \lceil (u_1-2)/2 \rceil$
      if $v \bmod 2 = 0$ **then** ▷ For relation type variable R_2.
        if $u_1 \geq 0$ **then**
          $\mathcal{S}$.add$\big((\text{Names}^{\text{nd}}[n+v], \text{Names}^{\text{rl}}[2m-1], \text{Names}^{\text{nd}}[n+u_1]) \big)$
        end if
        if $u_2 \geq 0$ **then**
          $\mathcal{Q}$.add$\big((\text{Names}^{\text{nd}}[n+v], \text{Names}^{\text{rl}}[2m-1], \text{Names}^{\text{nd}}[n+u_2]) \big)$
        end if
      else ▷ For relation type variable R_1.
        if $u_1 \geq 0$ **then**
          $\mathcal{S}$.add$\big((\text{Names}^{\text{nd}}[n+v], \text{Names}^{\text{rl}}[2m-2], \text{Names}^{\text{nd}}[n+u_1]) \big)$
        end if
        if $u_2 \geq 0$ **then**
          $\mathcal{Q}$.add$\big((\text{Names}^{\text{nd}}[n+v], \text{Names}^{\text{rl}}[2m-2], \text{Names}^{\text{nd}}[n+u_2]) \big)$
        end if
      end if
    end for
    $n \leftarrow n + 2^d$
  end for
end for

After getting all those representations, we can now focus on querying triplet representations (dashed green and red) by concatenating head and tail node representations w.r.t. relation types:

$$\begin{aligned}
\Gamma_{\text{tri}}\left((e_1, r_4, e_4), \mathbf{A}\right) &= H_{1,4} \parallel H_{4,4}, \\
\Gamma_{\text{tri}}\left((e_1, r_3, e_5), \mathbf{A}\right) &= H_{1,3} \parallel H_{5,3}, \\
\Gamma_{\text{tri}}\left((e_1, r_3, e_6), \mathbf{A}\right) &= H_{1,3} \parallel H_{6,3}, \\
\Gamma_{\text{tri}}\left((e_1, r_4, e_7), \mathbf{A}\right) &= H_{1,4} \parallel H_{7,4}.
\end{aligned}$$

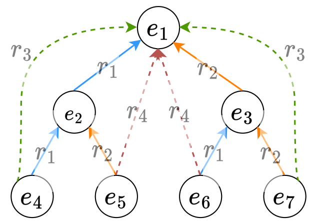


Figure 11: **Expressivity Limitation:** Relation r_1 and r_2 are always observed, while r_3 and r_4 are always querying. r_3 implies that relation types on the path are same, while r_4 implies that relation types on the path are different..

We can notice that

$$\overbrace{H_{1,4} \parallel H_{4,4}}^{\Gamma_{\text{tri}}((e_1,r_4,e_4),\mathbf{A})} = \overbrace{H_{1,4} \parallel H_{7,4}}^{\Gamma_{\text{tri}}((e_1,r_4,e_7),\mathbf{A})} = \overbrace{H_{1,3} \parallel H_{7,3}}^{\Gamma_{\text{tri}}((e_1,r_3,e_7),\mathbf{A})} = \overbrace{H_{1,3} \parallel H_{4,3}}^{\Gamma_{\text{tri}}((e_1,r_3,e_4),\mathbf{A})},$$
$$\overbrace{H_{1,4} \parallel H_{5,4}}^{\Gamma_{\text{tri}}((e_1,r_4,e_5),\mathbf{A})} = \overbrace{H_{1,4} \parallel H_{6,4}}^{\Gamma_{\text{tri}}((e_1,r_4,e_6),\mathbf{A})} = \overbrace{H_{1,3} \parallel H_{6,3}}^{\Gamma_{\text{tri}}((e_1,r_3,e_6),\mathbf{A})} = \overbrace{H_{1,3} \parallel H_{5,3}}^{\Gamma_{\text{tri}}((e_1,r_3,e_5),\mathbf{A})}.$$

Suppose the score of (e_u, r_c, e_v) utilizing such representation is $s_{u,c,v}$, we will have

$$s_{4,4,1} = s_{7,4,1} = s_{7,3,1} = s_{4,3,1},$$
$$s_{5,4,1} = s_{6,4,1} = s_{6,3,1} = s_{5,3,1}.$$

If a model can distinguish r_3 and r_4, it should at least rank node e_7 higher than e_6 given head node e_1 and relation r_3 since this is a positive triplet in training. Then, we will have $s_{7,3,1} > s_{6,3,1}$, since we already knew that $s_{7,3,1} = s_{7,4,1}, s_{6,3,1} = s_{6,4,1}$, we will also have $s_{7,4,1} > s_{6,4,1}$. This means that we rank node e_7 higher than node e_6 given head node e_1 and relation r_4, however, this is incorrect since $(e_7, r_4, e_1))$ is negative while (e_6, r_4, e_1) is positive. In summary, if we use double equivariant representation for triplet scoring in this specific example, there is no way for it to correctly rank r_3 and r_4 in the same time. This shows that double equivariant representation (even the most expressive) can face challenges for doubly inductive link prediction on discrete attributed multigraph.

E.2 Complexity Analysis for ISDEA

For each layer of our method ISDEA, it can be treated as running 2 unattributed GNN $|\mathcal{R}|$ times on the attributed multigraph, thus time cost is roughly $2|\mathcal{R}|$ times of adopted GNN. In our experiment, we use node representation GNNs (e.g., GIN [73], GAT [65], GraphConv [41]) 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 attributed multigraph, and $|\mathcal{S}|$ is number of fact triplets (number of edges) in attributed multigraph.

Besides, for both positive and negative samples (i, k, j), our method requires the shortest distance between any two nodes without considering (i, k, j). Pay attention that this can not be simply achieved from the Dijkstra or Floyd algorithm since the graph changes on computing each node pair, indeed computing such distance needs to traverse the enclosed graph [83, 60] between each node pair once.

E.3 Ablation study for ISDEA

Since a part of negative samplings is drawn by uniformly corrupting objects (without loss of generality), it is very likely that corrupted objects are far way from the subject while the true object is close to the subject. Then, the distance feature can help predict in such cases. However, the shortest distance feature will not provide any additional information if we corrupt the relation type. Under such a scenario, shortest distance itself may provide some features to achieve good ranking performance in inductive link prediction on attributed multigraph, thus we want to know if shortest distance feature augmentation contributes to the performance gain. We perform an ablation study for ISDEA with or without distance on doubly inductive link prediction over PediaTypes.

Table 7: **Relation & Node performance on Doubly Inductive Link Prediction over PediaTypes for ISDEA with/without Shortest Distances.** We report standard deviations over 5 runs. A higher value means better doubly inductive link prediction performance. **Even without the shortest distance as an augmented feature, our proposal still achieves comparable results, especially in the relation prediction task.**

(a) **Relation prediction $(i, ?, j)$ performance in %. Higher ↑ is better.**

Dataset		MRR	Hits@1	Hits@5	Hits@10
EN-FR	ISDEA w/ Distance	70.06±02.01	61.46±00.79	82.11±04.01	84.94±05.00
	ISDEA w/o Distance	68.65±00.41	60.34±00.53	80.17±00.99	82.80±01.73
FR-EN	ISDEA w/ Distance	69.01±00.57	58.18±00.14	83.19±01.73	84.75±02.51
	ISDEA w/o Distance	67.74±01.15	56.35±01.53	83.07±00.75	86.23±00.56
EN-DE	ISDEA w/ Distance	78.38±04.04	68.00±06.41	92.39±00.83	95.26±00.63
	ISDEA w/o Distance	76.52±01.32	67.66±02.37	87.49±00.87	88.47±00.64
DE-EN	ISDEA w/ Distance	88.82±00.28	84.83±00.29	93.59±00.53	94.23±00.71
	ISDEA w/o Distance	88.94±00.92	84.76±00.49	93.98±01.74	94.73±01.98
DB-WD	ISDEA w/ Distance	65.89±04.71	57.51±05.40	75.95±03.89	82.22±02.44
	ISDEA w/o Distance	70.66±07.05	63.36±05.30	79.87±10.19	82.96±11.89
WD-DB	ISDEA w/ Distance	72.57±00.73	62.72±01.24	86.10±01.26	88.87±02.94
	ISDEA w/o Distance	67.98±02.14	60.83±01.55	76.65±03.14	77.90±03.09
DB-YG	ISDEA w/ Distance	75.88±01.58	69.12±02.40	85.80±01.23	91.42±01.79
	ISDEA w/o Distance	75.42±00.35	69.17±01.13	84.86±01.58	88.78±02.36
YG-DB	ISDEA w/ Distance	74.04±00.47	66.68±00.81	83.36±01.55	85.34±01.49
	ISDEA w/o Distance	74.22±01.56	66.97±01.63	83.62±01.85	85.73±02.66

(b) **Node prediction $(i, k, ?)$ performance in %. Higher ↑ is better.**

Dataset		MRR	Hits@1	Hits@5	Hits@10
EN-FR	ISDEA w/ Distance	53.92±00.26	43.03±00.25	64.45±00.24	76.28±00.50
	ISDEA w/o Distance	45.12±00.41	34.04±00.36	56.61±00.48	63.46±00.76
FR-EN	ISDEA w/ Distance	57.68±00.68	47.38±00.28	67.24±01.32	77.51±01.46
	ISDEA w/o Distance	42.52±00.91	30.41±01.17	54.94±00.22	65.29±00.20
EN-DE	ISDEA w/ Distance	50.30±02.08	35.41±02.25	68.80±01.90	82.24±00.94
	ISDEA w/o Distance	45.16±00.76	30.26±00.76	62.59±00.57	76.98±00.63
DE-EN	ISDEA w/ Distance	51.33±00.40	37.12±00.31	68.20±00.53	81.80±00.68
	ISDEA w/o Distance	43.67±00.32	28.97±00.25	60.36±00.54	74.95±00.51
DB-WD	ISDEA w/ Distance	45.75±00.66	35.59±00.73	54.83±00.90	66.69±01.01
	ISDEA w/o Distance	40.26±03.77	30.59±03.89	48.15±03.68	59.43±03.76
WD-DB	ISDEA w/ Distance	51.64±00.60	40.56±01.72	62.60±02.55	75.19±03.12
	ISDEA w/o Distance	45.94±00.14	35.17±00.30	56.89±00.33	66.46±00.55
DB-YG	ISDEA w/ Distance	41.72±01.64	27.70±01.95	55.21±01.07	72.87±01.03
	ISDEA w/o Distance	32.71±00.60	17.69±00.39	47.82±00.60	66.92±01.90
YG-DB	ISDEA w/ Distance	48.21±01.06	35.29±01.67	61.87±01.30	76.41±01.52
	ISDEA w/o Distance	37.52±00.79	23.10±00.76	53.34±00.88	68.43±01.62

As shown in Table 7, even if the shortest distance is excluded from our model, our model still performs quite well and is better than most other baselines in the doubly inductive link prediction on PediaTypes. Especially, as we anticipate, the distance feature is more helpful in the node prediction task than the relation prediction task. Thus, we can say that double equivariant node representation itself is enough to provide good performance on doubly inductive link prediction.

E.4 Limitations and Impacts for ISDEA

ISDEA excels both in synthetic and real-world benchmarks. However, the simplification from pairwise to node embeddings in ISDEA limits its expressivity. In Appendix E.1.5, we give a synthetic counterexample how this could be an issue in some attributed multigraphs. Moreover, ISDEA has the same pre-processing scalability as GraIL. We also do not envision a direct negative social impact of our work.

F Future Work

As addressed in the main paper, our implemented architecture ISDEA has a few limitations, which could be addressed in future work. First, ISDEA has high pre-processing cost. This high time cost is introduced by using shortest distances whose computation is of the same complexity as enclosed subgraph. However, our ablation studies show that shortest distances is not a dominant factor in our model for real-world tasks, thus it is possible that shortest distances can be replaced by other heuristics that can be efficiently extracted.

Second, our specific implementation ISDEA happens to have high training and inference costs, since it relies on repeating GNNs for each relation. Thus, complexity ISDEA 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 which may reduce the cost.

Third, ISDEA 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 ISDEA representations can capture UQER Horn clauses, there is no algorithm to create UQER Horn clauses from ISDEA representations. This topic is known as *explainability* which is important in graph machine learning community. We leave such an algorithm as another future work other than optimization.