An OOD Multi-Task Perspective for Link Prediction with New Relation Types and Nodes

Jincheng Zhou, Beatrice Bevilacqua and Bruno Ribeiro

July 13, 2023

An OOD Multi-Task Perspective for Link Prediction with New Relation Types and Nodes

Jincheng Zhou
Department of Computer Science
Purdue University
West Lafayette, IN 47907
zhou791@purdue.edu

Beatrice Bevilacqua
Department of Computer Science
Purdue University
West Lafayette, IN 47907
bbevilac@purdue.edu

Bruno Ribeiro
Department of Computer Science
Purdue University
West Lafayette, IN 47906
ribeiro@cs.purdue.edu

Abstract

The task of inductive link prediction in (discrete) attributed multigraphs infers missing attributed links (relations) between nodes in new test multigraphs. Traditional relational learning methods face the challenge of limited generalization to OOD test multigraphs containing both novel nodes and novel relation types not seen in training. Recently, under the only assumption that all relation types share the same **structural** predictive patterns (single task), Gao et al. [18] proposed an OOD link prediction method using the theoretical concept of *double exchangeability* (for nodes & relation types), in contrast to the (single) exchangeability (only for nodes) used to design Graph Neural Networks (GNNs). In this work we further extend the double exchangeability concept to *multi-task double exchangeability*, where we define link prediction in attributed multigraphs that can have distinct and potentially conflicting predictive patterns for different sets of relation types (multiple tasks). Our empirical results on real-world datasets demonstrate that our approach can effectively generalize to entirely new relation types in test, without access to additional information, yielding significant performance improvements over existing methods.

1 Introduction

Discrete attributed multigraphs, which we refer as attributed graphs for simplicity, have been widely used for modeling relational data, which can also be expressed as a collection of triplets. Storing factual knowledge in attributed graphs enables their application across a wide variety of tasks, encompassing complex question answering [15, 21] and logical reasoning [8]. Since relational data is often incomplete, predicting missing triplets, or, equivalently, predicting the existence of a relation of a certain type between a pair of nodes, is a key task. However, conventional methods are generally limited to predicting missing links for relation types observed during training. As a consequence, standard attributed link prediction methods are incapable of making predictions that involve completely new relation types over completely new nodes (or new graphs), which is arguably the most difficult and perhaps the most interesting link prediction task in attributed graphs.

In this work we focus on the out-of-distribution (OOD) task of predicting missing triplets in test attributed graphs that contain new nodes and new relation types. Our definition of OOD task follows

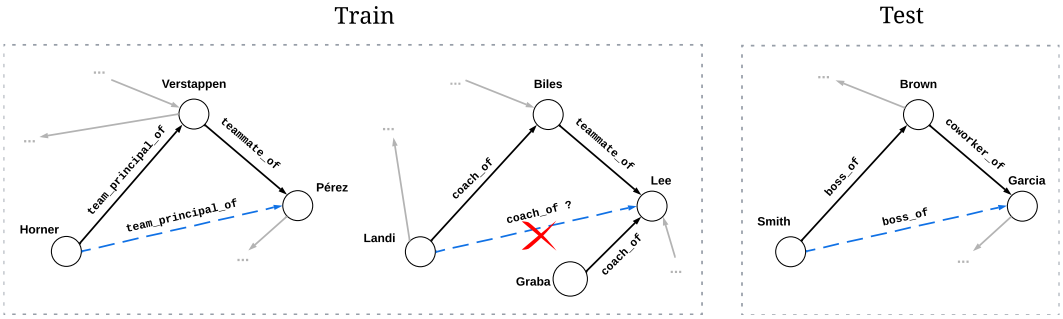


Figure 1: An example of a setting where not all relation types are exchangeable. Train graph contains relation types in the racing and the gymnastic communities. Test graph contains relation types in a business community.

prior work [27], which defines an OOD task as having train and test distribution with distinct support (distinct nodes and new relations types in attributed graphs). We also assume that no extra information is available either at training or test time, apart from the input (obsevable) graph with its nodes and relation types. As a result, existing zero-shot methods [32, 19, 25], which rely on textual descriptions of the relation types, and few-shot learning methods [47, 7, 38, 53, 22], which require examples of triplets to be predicted involving the new relation types, are unable to perform our OOD task. To the best of our knowledge, the only approach that can generalize to completely new nodes and relation types without requiring any extra information is the recent work of Gao et al. [18]. To solve the OOD task, the authors introduce the concept of double exchangeability, which is intuitively the property of all relation types, as well as all nodes, of being interchangeable with one another. Since the new test relation types are likewise assumed to be exchangeable with the training ones, the missing test triplets can be predicted directly using knowledge acquired during training. However, in some real-world attributed graphs, the double-exchangeability assumption might not hold. For instance, consider the scenario in Figure 1 where the training graph comprises two weakly-connected knowledge bases representing different sports communities (racing and gymnastic) with some different relation types. The predictive patterns for relation types in those communities might differ and potentially be contradictory, implying that *not all* relation types are exchangeable. In our example, two racing teammates necessarily have the same team principal in the racing community, and therefore (Horner, `team_principal_of`, Pérez) can be predicted from ((Horner, `team_principal_of`, Verstappen), (Verstappen, `teammate_of`, Pérez)). On the contrary, two gymnastic teammates might have different coaches and (Landi, `coach_of`, Lee) should not be predicted when seeing ((Landi, `coach_of`, Biles), (Biles, `teammate_of`, Lee)). Due to this contradictory patterns, `team_principal_of` and `coach_of` are not exchangeable. Similarly, the new test relation types might be exchangeable only with a subset of the training ones. In our example, `boss_of` is exchangeable with `team_principal_of`, but not with `coach_of`.

Our approach. To predict missing triplets involving new relation types in attributed graphs that have different predictive patterns, we suggest relaxing the double exchangeability assumption [18] and partitioning the set of relations into distinct groups, where each group exclusively contains relation types that are exchangeable among themselves. We demonstrate that these groups of relation types can be understood as distinct tasks in a multi-task setting. Within each task, we assume double exchangeability, so that within a task our method is able to extrapolate OOD. After learning to perform correctly in each task, we can apply the predictive patterns from the relevant tasks directly to predict missing triplets involving the new relation types. Importantly, we do not require prior knowledge of task membership; instead, we can learn the relation type-task assignments during training and adapt to the new test relation types through a test-time adaptation procedure.

Main contributions. Our main contributions are as follows: 1. We develop a method capable of modeling the existence of distinct and contradictory predictive patterns among various sets of relation types by treating them as separate tasks in a multi-task setting; 2. We propose a test-time adaptation procedure that learns task assignments for new relation types, enabling the application of our proposed method to entirely new test relation types; 3. We create new benchmark datasets that fit the multi-task scenario we focus on; 4. We develop a novel evaluation metric to more effectively measure the performance of existing methods in predicting missing triplets.

2 Related work

Link prediction in attributed graphs. Existing link prediction methods in attributed graphs can be categorized into factorization-based approaches [29, 6, 45, 49, 30, 41, 24, 13, 28, 39, 9] and GNN-based models [36, 42, 16, 52, 57]. Although the former exhibit remarkable performances, especially when combined with appropriate training strategies [34, 23], they are typically restricted to transductive settings. Conversely, GNN-based models can also be applied to inductive scenarios involving new nodes in test [40, 58, 1, 56, 17]. All these methods, however, cannot work when presented with new relation types in test, which instead represents the main interest of our paper.

Zero-shot and few-shot learning on new relation types. Recent works, aiming to predict missing triplets involving new relation types, consider the zero-shot or few-shot learning paradigm. To generalize to new relation types, zero-shot methods [32, 19, 25] use additional contextual information, such as the semantic descriptions of the relation types, making them unfit for our scenario. In contrast, few-shot methods primarily adopt a meta-learning paradigm [47, 7, 38, 53] that requires manually curated task labels during training and fine-tune on a small number of test triplet examples during test. To remove the need of task labels, Huang et al. [22] proposed a pre-training and fine-tuning few-shot procedure that uses subgraphs around the nodes, as they should contain the predictive pattern indicating the presence of the missing triplets. Similarly, our method removes the need of task labels, but unlike Huang et al. [22], our test-time adaptation procedure involves only the observable parts of the test graphs and does not require examples of the triplets to be predicted in training nor test.

The most closely related approach is the one presented in Gao et al. [18], which aims at OOD zero-shot generalizing to new test relation types by introducing the concept of exchangeability between relation types, which is intuitively their property of being interchangeable with one another. However, as we will see next, the approach is unable to properly model the difference in predictions between non-exchangeable relation types that have different (and potentially contradictory) predictive patterns. Due to space constrains, we refer the reader to Appendix A for detailed comparisons with prior work.

3 Problem Definition

In this section we introduce the notation used through the remainder of this work. We consider an attributed graph (multigraph) as a finite collection of typed relations between nodes. Formally, let $\mathcal{V}$ be a finite discrete set of nodes and $\mathcal{R}$ a finite discrete set of relation types. A triplet (u, r, v) in the attributed graph indicates that a node $u \in \mathcal{V}$ is linked to another node $v \in \mathcal{V}$ by means of a relation of type $r \in \mathcal{R}$. Without loss of generality, we assume node and relation sets are numbered, that is $\mathcal{V} \coloneqq \{1, 2, \ldots, N\}$ and $\mathcal{R} \coloneqq \{1, 2, \ldots, R\}$, where $N \geq 2$ and $R \geq 2$. Consequently, we can represent an attributed graph in tensor form as $\mathbf{A} \in \mathbb{A}_{N,R}$, with $\mathbb{A}_{N,R} = \{0, 1\}^{N \times R \times N}$, where $\mathbf{A}_{u,r,v} = 1$ if and only if (u, r, v) is a triplet in the attributed graph.

Link prediction in attributed graphs can be cast as a self-supervised learning problem [12, Appendix B], where an input graph $\mathbf{A}$ is assumed to be the result of the application of an unknown mask $M \in \{0, 1\}^{N \times R \times N}$ on an unknown attributed graph $\mathbf{A}^{(\text{full})}$, i.e., $\mathbf{A} = M \odot \mathbf{A}^{(\text{full})}$ with $\odot$ the element-wise product, where the masking process hides the existence of certain triplets. The goal of a model is to predict the existence of the masked (or missing) triplets from $\mathbf{A}$. That is, if $\overline{M}$ is the complement mask defined as $\overline{M} = \mathbf{1} - M$, the model is asked to predict $P(\overline{M} \odot \mathbf{A}^{(\text{full})} \mid \mathbf{A})$.

In this work we focus on predicting missing triplets in new attributed graphs with new relation types. Given a training graph $\mathbf{A}^{(\text{tr})}$, with node set $\mathcal{V}^{(\text{tr})}$ and relation set $\mathcal{R}^{(\text{tr})}$, we aim to learn a model capable of accurately predicting missing triplets in a test graph $\mathbf{A}^{(\text{te})}$, with node set $\mathcal{V}^{(\text{te})}$ and relation set $\mathcal{R}^{(\text{te})}$, involving both new nodes and new relations types: $\mathcal{V}^{(\text{tr})} \not\supseteq \mathcal{V}^{(\text{te})}$ and $\mathcal{R}^{(\text{tr})} \not\supseteq \mathcal{R}^{(\text{te})}$. To accurately predict missing test triplets without any extra information, such as contextual information (as in zero-shot methods [32, 19, 25]) or task labels (few-shot methods [47, 7, 38, 53, 22]), $\mathbf{A}^{(\text{te})}$ must exhibit predictive patterns found in $\mathbf{A}^{(\text{tr})}$, which implies that missing test triplets can be predicted using the knowledge acquired from training, even if the relations convey entirely different meanings.

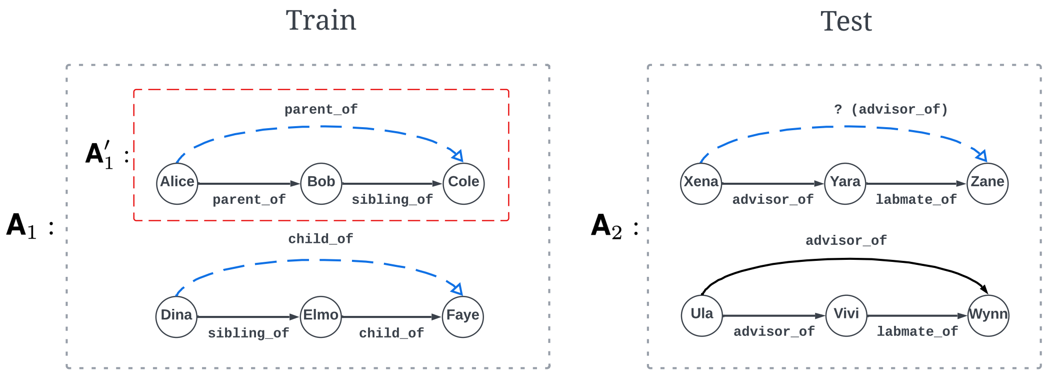


Figure 2: Example of our multi-task scenario. Solid arrows constitute the input (or observable) triplets, whereas the blue, dashed arrows represent the missing triplets to be predicted. The training attributed graph exhibits conflicting predictive patterns for `parent_of` and `child_of`. The new test relation `advisor_of` follows the same predictive pattern as `parent_of`.

To the best of our knowledge, the only approach capable of handling new relation types at test time without the need of extra information is the recent work of Gao et al. [18]. However, as we discuss next, Gao et al. [18] relies on the assumption that all relation types share a single predictive pattern.

Existing gap: Learning to predict new relation types on graphs with conflicting predictive patterns. To the best of our knowledge, no existing method can predict missing triplets involving new relation types without the need of extra information when relations exhibit different and potentially conflicting predictive patterns. Figure 2 illustrates an example of this setting, with solid arrows representing the input graph and dashed arrows denoting the missing triplets to be predicted. In this example, the training graph $\mathbf{A}_1$ has family-tree relationships, while the test graph $\mathbf{A}_2$ has relations in academia. Our goal is to learn predictive patterns from $\mathbf{A}_1$ and generalize to predict the missing triplets in $\mathbf{A}_2$. However, as demonstrated in the figure, the training graph $\mathbf{A}_1$ contains conflicting predictive patterns. The type of relation between Alice and Cole is the *first* relation type in the 2-hop chain ((Alice, `parent_of`, Bob), (Bob, `sibling_of`, Cole)), which is `parent_of`. Conversely, the type of relation between Dina and Faye is the *second* relation type in the 2-hop chain ((Dina, `sibling_of`, Elmo), (Elmo, `child_of`, Faye)), which is `child_of`. Having conflicting predictive patterns in training does not prevent accurate prediction of test triplets, as long as we can correctly identify which pattern, among the learned ones, a test relation follows. In our example, the true test relation type `advisor_of` between Xena and Zane in $\mathbf{A}_2$ shares the same predictive pattern as the relation `parent_of` in $\mathbf{A}_1$, as can be inferred from the observable triplet (Ula, `advisor_of`, Wynn). Since the pioneering work by Gao et al. [18] assumes a single predictive pattern, their proposed approach would be unable to model the difference between the predictions of `parent_of` and `child_of`, and, consequently, between those of `advisor_of` and `labmate_of`.

4 A Multi-Task Perspective on Inductive Learning of New Relation Types

In the previous section we noted that accurate prediction of missing triplets in a test graph with completely new relation types requires the test graph to exhibit the same predictive patterns as the training graph. Expanding upon this concept, we next define a predictive pattern through the notion of exchangeability and describe a task as a set of relation types sharing the same predictive patterns. This allows us to frame the problem of predicting with new relation types as a multi-task problem, where learning the different predictive patterns reduces to learning all tasks. Correctly performing on all tasks will then result in accurate predictions of test triplets having new relation types, as long as we can identify to which task they belong to.

4.1 Re-imagining Inductive Learning as Relational Tasks

We begin by defining the concept introduced in Gao et al. [18] of exchangeability between relation types, which can informally be understood as the property of (certain) relation types to be interchangeable with each other. This property encapsulates our notion of shared predictive patterns, as exchangeable relation types necessarily follow the same patterns.

Definition 4.1 (Exchangeability between relation types [18])**.** Let $\mathbf{A} \in \mathbb{A}_{N,R}$ be a random variable representing an attributed graph with node set $\mathcal{V} = \{1, 2, \ldots, N\}$ and relation set $\mathcal{R} = \{1, 2, \ldots, R\}$ and two relation types $r, r' \in \mathcal{R}$, we say that r and r' are *exchangeable* if there exists some node permutation $\pi \in \mathbb{S}_N$ and relation type permutation $\sigma \in \mathbb{S}_R$ such that the following two conditions are satisfied:

$$\sigma \circ r = r' \qquad \text{and} \qquad P(\mathbf{A}) = P(\sigma \circ \pi \circ \mathbf{A}),$$

where the permutation actions of π and σ are defined on the nodes and relation types respectively. That is, for all $u, v \in \mathcal{V}$ and $r \in \mathcal{R}$, the symmetric group $\mathbb{S}_N$ and the symmetric group $\mathbb{S}_R$ act on a graph via $(\pi \circ \mathbf{A})_{\pi \circ u, r, \pi \circ v} = \mathbf{A}_{u,r,v}$ and $(\sigma \circ \mathbf{A})_{u, \sigma \circ r, v} = \mathbf{A}_{u,r,v}$. We denote the exchangeability between relation types by $\sim_e$.

A similar formalization can be made for nodes. Due to space limit, we omit this definition, but we will assume nodes are exchangeable throughout the paper.

In the following, we introduce our theoretical contributions, which open up an entirely novel perspective to the problem. We start by proving that the exchangeability property between relation types can be regarded as a higher-order relation between the elements in $\mathcal{R}$, or, more precisely, as an equivalence relation on $\mathcal{R}$.

Lemma 4.2. The exchangeability between relation types $\sim_e$ defines an **equivalence relation** on $\mathcal{R}$, since it satisfies the reflexivity, symmetry, and transitivity properties.

The importance of Lemma 4.2 is in that it allows us to partition the set of relations $\mathcal{R}$ into disjoint equivalence classes. Each equivalence class contains relation types that are exchangeable with each other, whereas relation types that are not exchangeable belong to different equivalence classes. Consequently, these partitions of $\mathcal{R}$ can naturally be considered as different tasks, dubbed relational tasks, each containing all and only relation types that share the same predictive patterns.

Definition 4.3 (Relational tasks)**.** The *relational tasks* of an attributed graph $\mathbf{A}$ with node set $\mathcal{V}$ and relation set $\mathcal{R}$ are the equivalence classes $[r]$ of the relation types $r \in \mathcal{R}$ under $\sim_e$, i.e., each

$$\mathcal{T}_r \coloneqq [r] = \{r' \in \mathcal{R} : r' \sim_e r\} \tag{1}$$

is a relational task.

Viewing $\mathcal{R}$ as partitioned into disjoint relational tasks allows us to consider the problem from a multi-task perspective. Our goal then becomes finding a model able to accurately learn all tasks, specializing on the patterns that are unique to each task, which are potentially conflicting among each other. Such model will then be asked to recognize which task a test relation type belongs to, in order to correctly predict missing test triplets by applying what was learned on training for the same task.

4.2 Handling Conflicting Patterns for Different Relation Types as a Multi-task Scenario

Definitions 4.1 and 4.3 allow us to formalize in the following definition the attributed graphs of our interest, such as the one presented in Section 3, which we dub multi-task double-exchangeable attributed graphs. We adopt the term *double-exchangeable* from Gao et al. [18], since it inherently captures the idea of exchangeability both between node ids and between relation types – a shared concept between our work and theirs – but we extend it to our multi-task scenario.

Definition 4.4 (Multi-task double-exchangeable attributed graph)**.** Given an attributed graph $\mathbf{A}$, with node set $\mathcal{V}$ and relation set $\mathcal{R}$, $\mathbf{A}$ is said to be a *multi-task double-exchangeable attributed graph* if it has more than one relational tasks, i.e.,

$$|\{\mathcal{T}_r : r \in \mathcal{R}\}| > 1.$$

Equivalently, $\mathbf{A}$ is a multi-task double-exchangeable attributed graph if there exist two relation types $r, r' \in \mathcal{R}$ such that $r \not\sim_e r'$.

In this work, we assume our data satisfies Definition 4.4. More precisely, the training graph is considered a multi-task double-exchangeable attributed graph, in which each task comprises exchangeable relation types that follow the same predictive patterns, while distinct tasks may have conflicting patterns. We further assume our test graph is a multi-task double-exchangeable attributed graph, with tasks that constitute a subset of the training ones.

4.3 The Special Case of Single-Task Double-Exchangeable Attributed Graphs

Definition 4.4 can be specialized into what we refer to as single-task double-exchangeable attributed graphs if all relations belong to the same equivalence class (Lemma 4.2). An example of this scenario happens when considering only the boxed subgraph $\mathbf{A}_1'$ of the training graph in Figure 2. If we restrict our training graph to $\mathbf{A}_1'$ and maintain $\mathbf{A}_2$ as the test graph, then the true test relation type `advisor_of` between Xena and Zane in $\mathbf{A}_2$ can accurately be predicted by the model from Gao et al. [18]. This is because it follows the *only* predictive pattern present in the data, which is the one of the relation `parent_of`. Nonetheless, as emphasized throughout our work, the single-task configuration represents a particular case of the more general multi-task setting, which accommodates a greater variety of attributed graphs, such as the complete training graph $\mathbf{A}_1$ in Figure 2.

5 Proposed Method

In this section we introduce our framework to learn the different predictive patterns which are specific for each task and a procedure to generalize to new test relation types. Our proposed architecture models exchangeability between relation types belonging to the same task while differentiating them from the learned patterns of other tasks. To adapt to the unseen relations in the test attributed graph, we propose a test-time adaptation procedure to identify the tasks to which test relations belong.

5.1 Multi-Task Double-Equivariant Linear Layer

Suppose we knew that the ground-truth relational tasks $\{\mathcal{T}_r : r \in \mathcal{R}\}$ in Equation (1) given an attributed graph $\mathbf{A}$ with node set $\mathcal{V}$ and relation set $\mathcal{R}$. [1] Without loss of generality, consider an arbitrary ordering of the relational tasks and denote the ordered relational tasks as $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(K)}$, where K is the total number of tasks. We denote by $i : \mathcal{R} \to \{1, 2, \ldots, K\}$ a task index mapping, such that $\mathcal{T}^{(i(r))} = \mathcal{T}_r$. Inspired by the equivariant framework proposed by Maron et al. [26], Bevilacqua et al. [4] we present the following **Multi-Task Double-Equivariant Linear Layer (MTDE linear layer)**, which updates representations at every layer t as

$$\begin{aligned}\mathbf{H}^{(t+1)}_{\cdot,r,\cdot,\cdot} = L_1^{(t)}\big(\mathbf{H}^{(t)}_{\cdot,r,\cdot,\cdot}\big) + L_2^{(t)}\big(\mathbf{1} \otimes p_{i(r)} + \sum_{r' \in \mathcal{T}^{(i(r))} \setminus \{r\}} \mathbf{H}^{(t)}_{\cdot,r',\cdot,\cdot}\big) \\ + \sum_{\substack{k=1,\ldots,K \\ k \neq i(r)}} L_3^{(t)}\big(\mathbf{1} \otimes p_k + \sum_{r'' \in \mathcal{T}^{(k)}} \mathbf{H}^{(t)}_{\cdot,r'',\cdot,\cdot}\big),\end{aligned} \tag{2}$$

where $\mathbf{H}^{(t)} \in \mathbb{R}^{N \times R \times N \times d}$ is the layer input with $\mathbf{H}^{(0)} = \mathbf{A}$, and $L_1^{(t)}, L_2^{(t)}, L_3^{(t)} : \mathbb{R}^{N \times N \times d} \to \mathbb{R}^{N \times N \times d'}$ are GNN layers that output *pairwise* representations with N the number of nodes, R the number of relations and d, d' appropriate dimensions. The vectors $p_k \in \mathbb{R}^d$, $k = 1, 2, \ldots, K$ are learnable positional embeddings, each specific to the task $\mathcal{T}^{(k)}$, which are repeated on the last dimension through the Kronecker product with the matrix of all ones, $\mathbf{1} \in \{1\}^{N \times N \times 1}$. The sums in $\sum_{r' \in \mathcal{T}^{(i(r))} \setminus \{r\}} \mathbf{H}^{(t)}_{\cdot,r',\cdot,\cdot}$ and $\sum_{r'' \in \mathcal{T}^{(k)}} \mathbf{H}^{(t)}_{\cdot,r'',\cdot,\cdot}$ can be replaced by any other set aggregations.

Note that if the total number of relational tasks K is 1, then Equation (2) recovers the single-task double-equivariant layer proposed in Gao et al. [18]. Indeed, if $K = 1$, then $i(r) = 1$ for any $r \in \mathcal{R}$ with $\mathcal{T}^{(1)} = \mathcal{R}$, and Equation (2) can be rewritten as

$$\mathbf{H}^{(t+1)}_{\cdot,r,\cdot,\cdot} = L_1^{(t)}(\mathbf{H}^{(t)}_{\cdot,r,\cdot,\cdot}) + L_2^{(t)}\Big(\sum_{r' \in \mathcal{R} \setminus \{r\}} \mathbf{H}^{(t)}_{\cdot,r',\cdot,\cdot}\Big), \tag{3}$$

where the term $\mathbf{1} \otimes p_{i(r)}$ was absorbed into $L_2^{(t)}$.

The role of positional embeddings. Equation (2) uses the positional embedding vectors $p_j \in \mathbb{R}^d$, $j \in \{1, \ldots, K\}$, to allow representations of relation types belonging to different tasks to be different, even when they have isomorphic observable graphs. In order to understand the role of p_j in our architecture, we refer once again to Figure 2. Without the inclusion of the positional embeddings,

[1] In Section 5.2 we will remove this assumption and show how to learn task memberships.

Equation (2) would give the same representation to the missing triplet involving `parent_of` and the missing triplet involving `child_of`, even if those relations belong to two different relational tasks, because the inputs to $L_1^{(t)}, L_2^{(t)}, L_3^{(t)}$ are the same, starting from $t = 0$.

5.2 Learning Soft Task Membership via Attention Weights

The previous section assumes we know the ground-truth assignment of relation types to tasks. In what follows, we learn such assignments from data only. Intuitively, we need to partition all relations $\mathcal{R}$ into disjoint equivalence classes, where each partition corresponds to a unique relational task. This process is a discrete optimization problem, which we relax into a continuous one by means of an attention matrix $\alpha \in [0,1]^{R \times \hat{K}}$, where $\hat{K}$ is the maximum number of partitions we allow our architecture to model (i.e., the maximum number of relational tasks we expect). The individual attention value $\alpha_{r,k}$ denotes the degree (or probability) that the relation $r \in \mathcal{R}$ belongs to the k-th equivalence class, with the constraint that $\sum_{k=1}^{\hat{K}} \alpha_{r,k} = 1$ for all $r \in \mathcal{R}$. Hence, the MTDE linear layer of Equation (2) can be relaxed into what we called the **soft MTDE linear layer**:

$$\begin{aligned} \mathbf{H}^{(t+1)}_{\cdot,r,\cdot,\cdot} = {} & L_1^{(t)}\big(\mathbf{H}^{(t)}_{\cdot,r,\cdot,\cdot}\big) + L_2^{(t)}\big(\mathbf{1} \otimes p_{\hat{i}(r)} + \sum_{r' \in \mathcal{R} \setminus \{r\}} \alpha_{r',\hat{i}(r)} \mathbf{H}^{(t)}_{\cdot,r',\cdot,\cdot}\big) \\ & + \sum_{\substack{k=1,\ldots,\hat{K} \\ k \neq \hat{i}(r)}} L_3^{(t)}\big(\mathbf{1} \otimes p_k + \sum_{r'' \in \mathcal{R} \setminus \{r\}} \alpha_{r'',k} \mathbf{H}^{(t)}_{\cdot,r'',\cdot,\cdot}\big), \end{aligned} \tag{4}$$

where $\hat{i}(r) = \arg\max_{k=1\ldots\hat{K}} \alpha_{r,k}$, which ideally should give the correct id $i(r)$ of the ground-truth relational task $\mathcal{T}_r$ that the relation r belongs to. The final architecture, which we name the **Multi-Task Double-Equivariant Architecture (MTDEA)**, is obtained by stacking T soft MTDE linear layers to produce a graph representation $\Gamma(\mathbf{A}) \in \mathbb{R}^{N \times R \times N \times d}$ for a given attributed graph $\mathbf{A}$:

$$\Gamma(\mathbf{A}) \coloneqq L^{(T)}\big(f\big(\cdots f\big(L^{(1)}(\mathbf{A})\big)\cdots\big)\big),$$

where f is a non-polynomial activation such as ReLU. The predictions of individual triplets can then be obtained through a triplet score function $\Gamma_{\text{tri}} : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \times \mathbb{A}_{N,R} \to [0,1]$ followed by a sigmoid activation function, i.e., $\Gamma_{\text{tri}}((u,r,v), \mathbf{A}) \coloneqq \sigma(\Gamma(\mathbf{A})_{u,r,v,\cdot})$.

5.3 Dual-Sampling Loss with Task Membership Regularization

Existing literature that tackles link prediction in attributed graphs relies on loss as based on entity-centric negative sampling [49, 36, 58], where for each ground-truth (existing) triplet (u,r,v), the tail node v of (u,r,v) is randomly corrupted to obtain a fixed number of negative samples (u,r,v'). Such entity-based negative sampling is insufficient for our loss because correctly predicting the *relation type* between two nodes is equally important as correctly predicting the *tail node* given the head node and relation type. To this end, we propose the *dual-sampling task loss* $\mathcal{L}_{\text{dual}}$, which given the training attributed graph $\mathbf{A}^{(\text{tr})}$ with node set $\mathcal{V}^{(\text{tr})}$ and relation set $\mathcal{R}^{(\text{tr})}$, makes use of n negative samples obtained by corrupting tail nodes and m negative samples obtained by corrupting the relation types from positive samples, that is

$$\begin{aligned} \mathcal{L}_{\text{dual}} \coloneqq - \sum_{(u,r,v) \in \mathcal{S}} \Bigg(& \log(\Gamma_{\text{tri}}((u,r,v), \mathbf{A}^{(\text{tr})})) - \frac{1}{n} \sum_{i=1}^{n} \log(1 - \Gamma_{\text{tri}}((u,r,v'_i), \mathbf{A}^{(\text{tr})}))) \\ & - \frac{1}{m} \sum_{j=1}^{m} \log(1 - \Gamma_{\text{tri}}((u,r'_j,v), \mathbf{A}^{(\text{tr})}))) \Bigg), \end{aligned} \tag{5}$$

where $\mathcal{S} \coloneqq \{(u,r,v) \in \mathcal{V}^{(\text{tr})} \times \mathcal{R}^{(\text{tr})} \times \mathcal{V}^{(\text{tr})} \mid \mathbf{A}^{(\text{tr})}_{u,r,v} = 1\}$ is the set of positive triplets, (u,r,v'_i) is the i-th entity-based negative sample and (u,r'_j,v) the j-th relation-based negative sample corresponding to the positive triplet (u,r,v).

Equation (5) constitutes only a term of the loss function we optimize, which further contains regularization terms on the the attention matrix α. Intuitively, we want the individual attention values to be either 0 or 1, because each value should represent whether a relation type belongs to certain task (value 1) or not (value 0). Moreover, we aim to have a large concentration of the attention values, in order to have as few partitions as possible. Hence, we propose the following model loss, where $\lambda_1, \lambda_2 \in \mathbb{R}$ are hyper-parameters weighting the terms:

$$\mathcal{L} = \mathcal{L}_{\text{dual}} + \lambda_1 \underbrace{\sum_{r\in\mathcal{R}^{(\text{tr})}} \Big(-\sum_{j=1\dots k} \alpha_{r,j}\log\alpha_{r,j}\Big)}_{\mathcal{L}_{\text{1-hot}}} + \lambda_2 \underbrace{\Big(-\sum_{j=1\dots k}\text{LGamma}\Big(1+\sum_{r\in\mathcal{R}^{(\text{tr})}}\alpha_{r,j}\Big)\Big)}_{\mathcal{L}_{\text{conc}}}. \qquad (6)$$

The first term $\mathcal{L}_{\text{dual}}$ is the dual-sampling loss in Equation (5). The second term $\mathcal{L}_{\text{1-hot}}$ *minimizes* the entropy of the distribution of partition membership probabilities of each relation type r, so that individual attention values are pushed to be either 0 or 1, and attention weights for every relation type to be one-hot encoded. Finally, the third term $\mathcal{L}_{\text{conc}}$ takes advantage of the log-gamma function to encourage the relation set to be split in as few partitions as possible.

5.4 A Test-Time Adaptation Procedure

The attention matrix learned during training encodes task membership of relation types in training, and therefore it cannot be directly ported to the new test relation types in our test graph $\mathbf{A}^{(\text{te})}$ with $N^{(\text{te})}$ nodes and $R^{(\text{te})}$ relation types. We address this issue by adopting a test-time adaptation procedure where we optimize a test-time attention matrix $\alpha^{(\text{te})} \in [0,1]^{R^{(\text{te})}\times\hat{K}}$, while freezing all other parameters of the architecture. During the adaptation, only the observable triplets of the test graph $\mathbf{A}^{(\text{te})}$ are used for training $\alpha^{(\text{te})}$. That is, we follow the standard self-supervised procedure for link prediction (as in Section 3 and Cotta et al. [12, Appendix B]), and create a self-supervised mask $M \in \{0,1\}^{N^{(\text{te})}\times R^{(\text{te})}\times N^{(\text{te})}}$ that tunes $\alpha^{(\text{te})}$ to maximize $P(\overline{M}\odot\mathbf{A}^{(\text{te})} \mid M\odot\mathbf{A}^{(\text{te})})$, with $\overline{M} = \mathbf{1} - M$.

6 Experiments

In this section, we empirically evaluate our model in predicting missing triplets involving new relation types under different settings. We aim to address the following main questions:

Q1 *Does our model outperform the baselines when predicting new relation types in datasets containing multiple tasks?*

Q2 *How effectively does our model handle a general dataset when the presence of a multi-task structure is unknown?*

We present our primary experimental results below, and we defer readers to Appendices D and E for additional experiments and details.

Baselines. We evaluate our model against three baselines: IS-DEA [18], the homogeneous version of NBFNet [58] (NBFNet-homo), and the homogeneous version of IS-DEA (IS-DEA-homo). These homogeneous models are obtained by modifying the corresponding base models to treat all relation types equally. As a result, when predicting a tail node v given a head node u and a relation type r, a homogeneous model returns the node v for which the edge (u, v) is most likely to exist, regardless of the relation type. When predicting the relation type r between given nodes u and v, a homogeneous model returns a uniform prediction over all possible relation types. This modification allows NBFNet to generalize to new test relation types, a task it cannot perform otherwise. To the best of our knowledge, these models are the only ones that are applicable to our scenario.

Dual-sampling metrics. In line with the dual-sampling loss we proposed in Equation (5), we present the *dual-sampling metrics*, which include Hits@k, Mean Rank (MR), and Mean Reciprocal Rank (MRR). For each positive triplet we generate 24 negative samples by corrupting the tail entity and 26 negative samples by corrupting the relation type. These metrics are better suited for measuring the capabilities of the models in our tasks, and we refer to Appendix E.1 for a thorough discussion.

A1: Multi-task datasets. To address **Q1**, we create a novel multi-task scenario, named WikiTopics-MT, in the WikiTopics datasets introduced by Gao et al. [18] and obtained from the WikiData5M datasets [44] by grouping the relation types into different topics, such as Art, Education, and Sports. The WikiTopics dataset was employed by Gao et al. [18] to assess the

Table 1: Model performance on WIKITOPICS-MT, tested on two topics (HEALTH and TAXONOMY) not seen in training. We report mean and std over 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2). **Our models consistently outperform the baselines with significantly smaller standard deviations on both test topics.**

	Test on HEALTH topic				Test on TAXONOMY topic			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	15.843 (0.119)	0.122 (0.001)	0.041 (0.000)	0.339 (0.003)	16.260 (0.068)	0.113 (0.000)	0.034 (0.000)	0.315 (0.001)
IS-DEA-homo	42.276 (0.768)	0.025 (0.001)	0.000 (0.000)	0.000 (0.000)	40.963 (1.276)	0.026 (0.001)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	15.405 (6.030)	0.384 (0.133)	0.323 (0.140)	0.481 (0.138)	19.143 (3.895)	0.323 (0.063)	0.269 (0.063)	0.365 (0.082)
MTDEA ($\hat{K} = 2$)	15.015 (4.947)	0.422 (0.170)	0.358 (0.191)	0.513 (0.112)	**16.840 (4.222)**	0.393 (0.141)	0.330 (0.157)	0.457 (0.121)
MTDEA ($\hat{K} = 4$)	16.576 (2.293)	0.441 (0.115)	0.390 (0.127)	0.496 (0.108)	18.171 (4.242)	0.365 (0.191)	0.307 (0.203)	0.417 (0.175)
MTDEA ($\hat{K} = 6$)	**13.774 (3.224)**	**0.504 (0.013)**	**0.457 (0.012)**	**0.555 (0.010)**	16.902 (3.837)	**0.470 (0.018)**	**0.422 (0.010)**	**0.504 (0.025)**

extrapolation performance of the double-equivariant model, IS-DEA, when trained on one topic and tested on a different one. For our multi-task dataset, we select for training pairs of topics where, as shown in Gao et al. [18], IS-DEA exhibits the lowest transfer-topic performance (the ART and PEOPLE) and test on a third topic (HEALTH or TAXONOMY) on which the IS-DEA trained on one training topic (e.g. ART) performs good but the IS-DEA trained on the other training topic (e.g. PEOPLE) performs poorly. Since the transfer-topic performance of IS-DEA indicates the degree of double-exchangeability between graphs in the two topics, which is related to the definition of tasks (Definition 4.3), this selection strategy likely produces train and test graphs with multiple tasks.

Table 1 presents the model performance on WIKITOPICS-MT when trained on relations from both ART and PEOPLE topics and tested on either HEALTH or TAXONOMY topic. Our model MTDEA, and in particular the one with the largest number of task partitions ($\hat{K} = 6$), outperforms all baselines, while having a significantly smaller standard deviation on MRR, Hits@1, and Hits@10 metrics, suggesting that the WIKITOPICS-MT dataset indeed possesses a complicated multi-task structure, and our model, which has the best multi-task modeling capability, yields most consistent performance.

A2: General dataset. To address **Q2**, we create the FBNELL dataset by combining FB15K-237 [36] and NELL-995 [46]. The training graph consists of the 50 most frequent relation types and the test graph of the 100 most frequent ones for each dataset. We note that it is not clear from this construction whether FBNELL exhibits multi-task structures because the relation types might still be exchangeable despite belonging to different domains.

Table 2: Model performance on FBNELL. We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2). **Our model with $\hat{K} = 2$ is comparable to, if not better than, the baselines even when the dataset does not have a multi-task structure.**

Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	14.116 (0.029)	0.129 (0.001)	0.042 (0.001)	0.379 (0.002)
IS-DEA-homo	31.625 (0.940)	0.033 (0.001)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	10.925 (0.383)	**0.624 (0.010)**	**0.562 (0.014)**	0.697 (0.010)
MTDEA ($\hat{K} = 2$)	**9.106 (0.162)**	0.622 (0.012)	0.553 (0.010)	**0.704 (0.024)**
MTDEA ($\hat{K} = 4$)	10.730 (0.666)	0.606 (0.006)	0.543 (0.007)	0.680 (0.023)
MTDEA ($\hat{K} = 6$)	10.386 (0.683)	0.609 (0.015)	0.547 (0.017)	0.678 (0.008)

Table 2 shows the performance on FBNELL. Our model is on par (and sometimes outperforms) the baselines, even in this scenario where a multi-task structure may not be present. We associate smaller performance gaps to the simplicity of the constructed dataset, which does not seem to exhibit complex multi-task structures (the smallest $\hat{K}$ has the highest performance). Overall, our result suggests that in real-world scenarios it is always advantageous to employ the MTDEA model because, even in the single-task setting (due to its regularization towards fewer relation equivalence classes and patterns), it obtains a similar, if not better, performance than the baselines.

7 Conclusions

In this work we studied the problem of extrapolating to new relation types in link prediction tasks in discrete attributed multigraphs. To overcome the challenge faced by existing work when the graphs contain relation types exhibiting contradictory predictive patterns, we proposed a relaxation of the double exchangeability assumption of Gao et al. [18] and demonstrated that this relaxation can be interpreted within a multi-task framework. We designed an architecture capable of modeling this multi-task double exchangeability, along with a test-time adaptation procedure to learn task assignments for new relation types. To empirically evaluate our method, we introduced new benchmark datasets featuring multi-task structures and presented novel evaluation metrics to measure its benefits.

Acknowledgments and Disclosure of Funding

The authors would like to thank Jianfei Gao and Yangze Zhou for insightful discussions. This work was supported in part by the National Science Foundation (NSF) awards CAREER IIS-1943364, CCF-1918483, and CNS-2212160 and an Amazon Research Award. Any opinions and findings expressed in this manuscript are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Mehdi Ali, Max Berrendorf, Mikhail Galkin, Veronika Thost, Tengfei Ma, Volker Tresp, and Jens Lehmann. Improving inductive link prediction using hyper-relational facts. In *The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20*, pages 74–92. Springer, 2021.

[2] Pablo Barceló, Egor V. Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *International Conference on Learning Representations*, 2020.

[3] Pablo Barceló, Mikhail Galkin, Christopher Morris, and Miguel Romero Orth. Weisfeiler and leman go relational. In *The First Learning on Graphs Conference*, 2022.

[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*, 2022.

[5] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[7] 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 4208–4217, Hong Kong, China, November 2019. Association for Computational Linguistics.

[8] Xiaojun Chen, Shengbin Jia, and Yang Xiang. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948, 2020.

[9] Yihong Chen, Pasquale Minervini, Sebastian Riedel, and Pontus Stenetorp. Relation prediction as an auxiliary training objective for improving multi-relational graph representations. *ArXiv*, abs/2110.02834, 2021.

[10] 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 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[11] 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*, 2022.

[12] Leonardo Cotta, Beatrice Bevilacqua, Nesreen Ahmed, and Bruno Ribeiro. Causal lifting and link prediction. *arXiv preprint arXiv:2302.01198*, 2023.

[13] 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.

[14] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[15] Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*, 2020.

[16] Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. Message passing for hyper-relational knowledge graphs. In *EMNLP*, 2020.

[17] 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*, 2022.

[18] Jianfei Gao, Yangze Zhou, Jincheng Zhou, and Bruno Ribeiro. Inductive link prediction for both new nodes and new relation types via double equivariance. *arXiv preprint arXiv:2302.01313*, 2023.

[19] 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 *Proceedings of the Web Conference 2021*, pages 3325–3336, 2021.

[20] Patrick Hohenecker and Thomas Lukasiewicz. Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research*, 68:503–540, 2020.

[21] Ningyuan Huang, Yash R Deshpande, Yibo Liu, Houda Alberts, Kyunghyun Cho, Clara Vania, and Iacer Calixto. Endowing language models with multimodal knowledge graph representations. *arXiv preprint arXiv:2206.13163*, 2022.

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

[23] Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. Knowledge base completion: Baseline strikes back (again). *ArXiv*, abs/2005.00804, 2020.

[24] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872. PMLR, 2018.

[25] Mingchen Li, Junfan Chen, Samuel Mensah, Nikolaos Aletras, Xiulong Yang, and Yang Ye. A hierarchical n-gram framework for zero-shot link prediction. *arXiv preprint arXiv:2204.10293*, 2022.

[26] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, 2022.

[27] S Chandra Mouli and Bruno Ribeiro. Asymmetry learning for counterfactually-invariant classification in ood tasks. In *International Conference on Learning Representations*, 2022.

[28] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 327–333, 2018.

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

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

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[32] 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.

[33] Haiquan Qiu, Yongqi Zhang, Yong Li, and Quanming Yao. Logical expressiveness of graph neural network for knowledge graph reasoning, 2023.

[34] 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.

[35] 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.

[36] 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.

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

[38] Jian Sun, Yu Zhou, and Chengqing Zong. One-shot relation learning for knowledge graphs via neighborhood aggregation and paths encoding. *Transactions on Asian and Low-Resource Language Information Processing*, 21(3):1–19, 2021.

[39] 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.

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

[41] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA, 20–22 Jun 2016. PMLR.

[42] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, 2020.

[43] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014. ISSN 0001-0782.

[44] 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.

[45] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28 (1), Jun. 2014.

[46] Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. *CoRR*, abs/1707.06690, 2017. URL `http://arxiv.org/abs/1707.06690`.

[47] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. One-shot relational learning for knowledge graphs. *arXiv preprint arXiv:1808.09040*, 2018.

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

[49] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, 2015.

[50] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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

[52] Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. Generalized multi-relational graph convolution network. *arXiv*, page 07331, 2020.

[53] 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.

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

[55] 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.

[56] Yongqi Zhang and Quanming Yao. Knowledge graph reasoning with relational digraph. In *Proceedings of the ACM Web Conference 2022*, pages 912–924, 2022.

[57] Zhanqiu Zhang, Jie Wang, Jieping Ye, and Feng Wu. Rethinking graph convolutional networks in knowledge graph completion. In *The Web Conference 2022*, 2022.

[58] 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, 2021.

Supplementary Material for

An OOD Multi-Task Perspective for Link Prediction with New Relation Types and Nodes

A Expanded Related Work

GNNs for attributed graph completion. Due to their recent success in diverse graph-learning tasks, Graph Neural Networks (GNNs) have been widely used to predict missing attributed links between nodes in attributed graphs. One of the first adaptation of standard GNNs to multi-relational data was proposed in Schlichtkrull et al. [36], while an alternative formulation has been considered in Vashishth et al. [42]. These two models have then inspired several improved versions for both transductive [16, 52, 57] and inductive [40, 58, 1, 56] link prediction tasks on attributed graphs, even in the context of large graphs [17]. Recently, their limitations and their relationships have been studied from a theoretical viewpoint, by relating their capabilities in distinguishing different attributed graphs to the Weisfeiler-Leman algorithm [3]. These methods, however, cannot work when presented with new relation types in test, which instead represents the main interest of our work.

Tensor Factorization. Factorization-based methods [29, 6, 45, 49, 30, 41, 24, 13, 28, 39, 9] are classical graph representation learning methods for attributed graphs. Despite their superior empirical performance on transductive tasks, especially when coupled with specific training strategies [34, 23], these models cannot be applied to inductive tasks featuring new nodes in test. To overcome this limitation, Chen et al. [10] propose a new architecture that borrows principles from GNNs and bridges the gap between these two approaches. All these methods, however, are not applicable to the tasks of our interest, where test graphs contain both new nodes and relation types.

Logical reasoning. Predicting missing attributed links in attributed graph can also be performed by learning logical rules that are then used to infer the missing links. Yang et al. [49, 50], Sadeghian et al. [35], Cheng et al. [11] focus on learning Horn clauses from the graph. To understand the expressive power of standard GNNs in learning logical rules, Barceló et al. [2] characterize the fragment of FOC_2 formulas, a well-studied fragment of first order logic, that can be expressed as GNNs. Recently, Qiu et al. [33] extended the analysis to heterogeneous graphs.

Zero-shot learning for link prediction in attributed graphs. To predict links involving completely new relation types at test time, zero-shot methods require additional information encoding the semantic of the relation types. Qin et al. [32] rely on semantic features obtained from the text descriptions of the relation types. Geng et al. [19] enrich the relation features using information from the ontological schema. Finally, Li et al. [25] use the character n-gram information from the relation name to generate more expressive representations of the relations. As we do not assume access to any extra information apart from the input graphs, not even the relation textual names[2], these methods are inapplicable to our scenario.

Few-shot learning for link prediction in attributed graphs. To the best of our knowledge, most few-shot methods predict novel relation types in test following a meta-learning paradigm [47, 7, 38, 53]. This meta-learning paradigm requires learning with a set of meta-training tasks and then adapts to a new task during meta-testing. Consequently, these methods require access to many few-shot tasks for training, which can be challenging to obtain in real-world datasets. To remove the dependency on manually created training tasks, Huang et al. [22] develop a pretraining procedure followed by finetuning, which predicts triplets for new relation types through the existence of a hypothesis in the form of a subgraph, which should represent the predictive pattern indicating the existence of the triplet. Differently from our approach, however, Huang et al. [22] require access to support triplets (examples) of the new relation types, which are used to learn the hypothesis proposal module. Our approach instead only relies on seeing the observable test graph to learn the task-membership of the new relation types, whose observable triplets are not necessarily similar to the triplets we aim to predict, and therefore further removes the need of examples of test triplets to be predicted.

[2] We always consider relation types as numbers, $\mathcal{R} \coloneqq \{1, \ldots, R\}$, $R \in \mathbb{N}$.

B Theoretical Analysis

Lemma 4.2. The exchangeability between relation types $\sim_e$ defines an **equivalence relation** on $\mathcal{R}$, since it satisfies the reflexivity, symmetry, and transitivity properties.

Proof. We prove each property separately.

The exchangeability between relation types $\sim_e$ is reflexive. This can be trivially shown by considering the identity node permutation $\text{Id}_N \in \mathbb{S}_N$ and identity relation type permutation $\text{Id}_R \in \mathbb{S}_R$. Namely, let $\mathbf{A} \in \mathbb{A}_{N,R}$ be a random variable representing an attributed graph sampled from some data distribution. For any relation $r \in \mathcal{R}$, naturally $\text{Id}_R \circ r = r$ and $\text{Id}_R \circ \text{Id}_N \circ \mathbf{A} = \mathbf{A}$, and consequently $P(\text{Id}_R \circ \text{Id}_N \circ \mathbf{A}) = P(\mathbf{A})$. Hence, r is exchangeable with r.

The exchangeability between relation types $\sim_e$ is symmetric. We first note that the node permutation and relation type permutation are commutative [18]. That is, given any attributed graph $\mathbf{A} \in \mathbb{A}_{N,R}$, $\pi \in \mathbb{S}_N$, and $\sigma \in \mathbb{S}_R$, we have $\sigma \circ \pi \circ \mathbf{A} = \pi \circ \sigma \circ \mathbf{A}$. In other words, it makes no difference whether we permute the nodes first or we permute the relation types first. Now, for any two relations $r, r' \in \mathcal{R}$, if r is exchangeable with r', then we know there exists some $\pi \in \mathbb{S}_N$ and $\sigma \in \mathbb{S}_R$ such that $\sigma \circ r = r'$ and $P(\mathbf{A}) = P(\sigma \circ \pi \circ \mathbf{A})$ for any $\mathbf{A}$ sampled from the data distribution. Since $\mathbb{S}_N$ and $\mathbb{S}_R$ are groups, π and σ have unique inverses $\pi' \in \mathbb{S}_N$ and $\sigma' \in \mathbb{S}_R$ satisfying $\pi' \circ \pi = \text{Id}_N$ and $\sigma' \circ \sigma = \text{Id}_R$ respectively. Hence,

$$\sigma' \circ r' = \sigma' \circ (\sigma \circ r) = (\sigma' \circ \sigma) \circ r = \text{Id}_N \circ r = r$$
$$\sigma' \circ \pi' \circ (\sigma \circ \pi \circ \mathbf{A}) = \sigma' \circ (\pi' \circ \pi) \circ \sigma \circ \mathbf{A} = \sigma' \circ \sigma \circ \mathbf{A} = \mathbf{A}.$$

Consequently, we have $P(\sigma \circ \pi \circ \mathbf{A}) = P(\mathbf{A}) = P(\sigma' \circ \pi' \circ (\sigma \circ \pi \circ \mathbf{A}))$. Moreover, since permutations are bijective mappings, we know that for any $\mathbf{A}' \in \mathbb{A}_{N,R}$ there exists some $\mathbf{A}$ such that $\mathbf{A}' = \sigma \circ \pi \circ \mathbf{A}$. Hence, $P(\mathbf{A}') = P(\sigma \circ \pi \circ \mathbf{A}) = P(\sigma' \circ \pi' \circ (\sigma \circ \pi \circ \mathbf{A})) = P(\sigma' \circ \pi' \circ \mathbf{A}')$ for any $\mathbf{A}'$ sampled from the data distribution. Therefore, r' is also exchangeable with r.

The exchangeability between relation types $\sim_e$ is transitive. Let $r_1, r_2, r_3 \in \mathcal{R}$ be three relation types such that r_1 is exchangeable with r_2, and r_2 is exchangeable with r_3. Then, there exists some $\pi_1, \pi_2 \in \mathbb{S}_N$ and $\sigma_1, \sigma_2 \in \mathbb{S}_R$ such that

$$\sigma_1 \circ r_1 = r_2 \qquad \text{and} \qquad P(\mathbf{A}) = P(\sigma_1 \circ \pi_1 \circ \mathbf{A})$$
$$\sigma_2 \circ r_2 = r_3 \qquad \text{and} \qquad P(\mathbf{A}') = P(\sigma_2 \circ \pi_2 \circ \mathbf{A}'),$$

for any $\mathbf{A}$ and $\mathbf{A}'$ sampled from the data distribution. Hence, take any $\mathbf{A} \in \mathbb{A}_{N,R}$,

$$\begin{aligned} P(\mathbf{A}) &= P(\sigma_1 \circ \pi_1 \circ \mathbf{A}) = P(\sigma_2 \circ \pi_2 \circ (\sigma_1 \circ \pi_1 \circ \mathbf{A})) \\ &= P((\sigma_2 \circ \sigma_1) \circ (\pi_2 \circ \pi_1) \circ \mathbf{A}), \end{aligned}$$

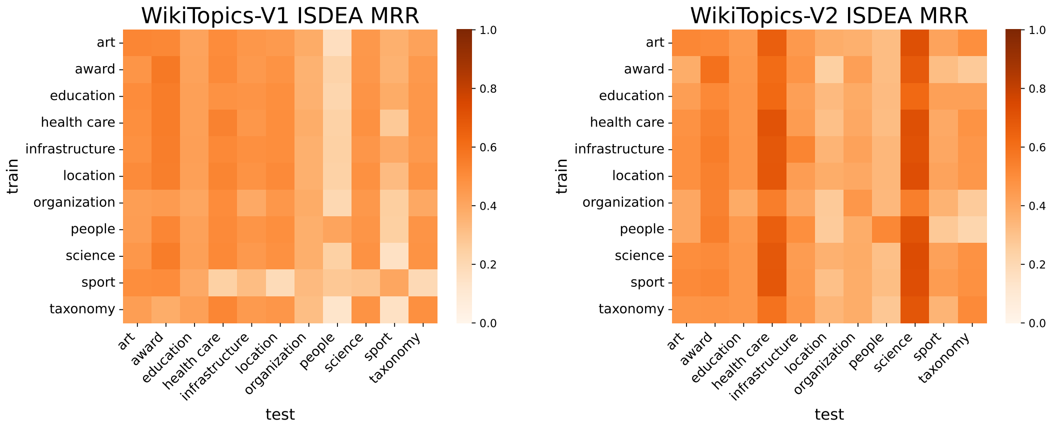
where we also have $(\sigma_2 \circ \sigma_1) \circ r_1 = r_3$, showing that r_1 is exchangeable with r_3.

Since the exchangeability between relation types is reflexive, symmetric, and transitive, it is an equivalence relation on $\mathcal{R}$. □

C Datasets Construction

C.1 WikiTopics-MT

The WikiTopics-MT scenarios are derived from the WikiTopics dataset previously introduced by Gao et al. [18], which comprises 11 attributed graphs with relation types in each attributed graph corresponding to a specific topic. Our goal is to construct training graphs exhibiting multi-task structures while ensuring the test graph possesses a task also present in training. In Gao et al. [18], this dataset was leveraged to assess the ISDEA model's zero-shot generalization capabilities on pairs of attributed graphs that may have relation types not exchangeable with each other. We observe that the ISDEA model's performance can be viewed from an alternative perspective: poor test performance on a certain topic when trained on a different topic indicates that the topics contain relation types that are likely not exchangeable. Consequently, the corresponding attributed graphs likely contains distinct tasks (Definition 4.3), implying that combining the attributed graphs of these two topics would yield an aggregated attributed graph containing multiple tasks.



(a) WIKITOPICS-V1.

(b) WIKITOPICS-V2.

Figure 3: ISDEA [18] transfer-topics performance on both versions of WIKITOPICS without the shortest-distance heuristic embeddings. Color indicates the value of the dual-sampling MRR metric.

To identify which pairs of topics are likely to contain distinct tasks, we rerun the ISDEA model from Gao et al. [18] on both versions of the WIKITOPICS dataset. To reduce the memory footprint and the computational time, we run the experiments without the shortest-distance heuristic embeddings that are used to augment the triplet representations in Gao et al. [18] (as explained in Appendix D). Figures 3a and 3b show the heatmaps representing the transfer-topic performance of the ISDEA model on the 121 pairs of topics for each version of the dataset. Each row in Figures 3a and 3b corresponds to one training topic, each column corresponds to a test topic, and the color represents the performance evaluated using the dual-sampling Mean Reciprocal Ranks (MRR).

Based on the heatmaps (Figures 3a and 3b), we devise multi-task scenarios using the outlined strategy. For the training graph, we pick two topics such that when trained on one and evaluated on the other the performance is low (e.g. HEALTH and SPORT in Figure 3a). This indicates that the ISDEA model, trained on one topic (HEALTH), demonstrates relatively poor zero-shot generalization performance on the other topic (SPORT), suggesting that the attributed graphs associated with these topics likely contain distinct tasks. We then combine the attributed graphs of the two topics to create an aggregated graph (HEALTH + SPORT), which is expected to exhibit multi-task structures. Next, we select one test topic for each training topic, and evaluate separately on the two test topics. A test topic (e.g., LOCATION) is determined such that the ISDEA model trained on one of the training topics (in this case, SCIENCE) performs well on the selected test topic (LOCATION), but the good performance on this test topic may not necessarily be observed when the ISDEA model is trained on the other training topic (SPORT).

Using this data creation strategy, we construct the following four multi-task scenarios, two for each version of WIKITOPICS:

1. **WIKITOPICS-MT**: created from WIKITOPICS-V1. The training topic is a combination of ART and PEOPLE, and the 2 test topics are HEALTH and TAXONOMY.
2. **WIKITOPICS-MT2**: created from WIKITOPICS-V1. The training topic is a combination of SPORT and HEALTH, and the 2 test topics are LOCATION and SCIENCE.
3. **WIKITOPICS-MT3**: created from WIKITOPICS-V2. The training topic is a combination of PEOPLE and TAXONOMY, and the 2 test topics are ART and INFRASTRUCTURE.
4. **WIKITOPICS-MT4**: created from WIKITOPICS-V2. The training topic is a combination of LOCATION and ORGANIZATION, and the 2 test topics are HEALTH and SCIENCE.

Table 3 shows the datasets statistics of the 4 multi-task scenarios. The experiment results of the WIKITOPICS-MT scenario are shown in Section 6 in the main paper, and the additional experiment results of the WIKITOPICS-MT2, WIKITOPICS-MT3, and WIKITOPICS-MT4 scenarios are shown in Appendix E.3.

Table 3: Dataset statistics of the WikiTopics-MT scenarios.

Scenarios	Train/Test Topics	# Entities	# Relation Types	# Observable Triplets	# Missing Triplets
WikiTopics-MT	Art + People (Train)	10000	35	32145	3571
	Health (Test)	10000	8	14110	1566
	Taxonomy (Test)	10000	10	16526	1834
WikiTopics-MT2	Sport + Health (Train)	10000	19	43528	4836
	Location (Test)	10000	11	22971	2552
	Science (Test)	10000	17	14852	1650
WikiTopics-MT3	People + Taxonomy (Train)	10000	99	57815	6243
	Art (Test)	10000	65	28023	3113
	Infrastructure (Test)	10000	37	21646	2405
WikiTopics-MT4	People + Taxonomy (Train)	10000	94	54140	6015
	Location (Test)	10000	62	80269	8918
	Organization (Test)	10000	34	30214	3357

Table 4: Dataset statistics of the FBNELL dataset.

Train/Test Splits	# Entities	# Relation Types	# Observable Triplets	# Missing Triplets
Train	4797	100	10275	1224
Test	4725	200	10685	597

C.2 FBNELL

We create the FBNELL dataset by combining the FB15K-237 [36] and the NELL-995 [46] datasets. The training graph is obtained by first choosing the top 50 most frequent relation types in each FB15K-237 and NELL-995, yielding a total of 100 relation types, and then extracting the triplets corresponding to these 100 relation types. For the test graph, we pick the top 100 most frequent relation types from each dataset, extract the triplets corresponding to the resulting 200 relation types, and predict only those triplets that involve new relation types, while using the remaining triplets as the observable (test) graph. Consequently, the test graph's set of relation types forms a strict superset of those in the training graph, but the evaluation is performed only on the unseen ones. Table 4 shows the statistics of the dataset.

We emphasize that, unlike in the data construction strategy employed for the WikiTopics-MT scenarios, we do not actively identify the presence of multiple tasks within either FB15K-237 or NELL-995, nor verify whether their combination exhibit a multi-task structure. Therefore, even though the two datasets come from distinct domains, they might still share the same relational task (single-task).

C.3 MetaFam

We construct a synthetic dataset, dubbed MetaFam, that explicitly exhibits conflicting predictive patterns, or equivalently, a multi-task structure. In particular, we recreate the conflicting predictive patterns shown in Figure 2. We generate the dataset by first creating the family trees using the ontology and the code provided in Hohenecker and Lukasiewicz [20]. Each family tree is generated by starting from a single person and incrementally adding a new child to an existing node until the tree reaches the maximum size of 26 nodes or a maximum depth of 5. The parent of the node to be added is chosen uniformly at random, with the only constraint that the maximum branching factor of each tree is 5. Each family tree contains triplets involving 29 different relation types representing different kinds of relationships, such as `mother_of`, `daughter_of`, `uncle_of`.

We generate the training split by randomly selecting 50 non-isomorphic family trees. In each training family tree we mask out either some of the triplets with relation types `mother_of` and `father_of`, or some of the triplets involving relation types `son_of` and `daughter_of`, and we use those as the triplets we aim to predict during training time. Doing so ensures that the model is challenged with the two conflicting patterns illustrated in Figure 2 when learning to predict these triplets. Specifically, the former two relation types, `mother_of` and `father_of`, obey to the first kind of predictive pattern (e.g. subgraph $\mathbf{A}_1'$ as illustrated in Figure 2), and the latter two relation types, `son_of` and `daughter_of`, follow the second kind of predictive pattern (the rest of the training graph $\mathbf{A}_1$ as illustrated in Figure 2). As the test split, we create 25 additional non-isomorphic family trees having the same relation types

Table 5: Dataset statistics of the MetaFam dataset.

Train/Test Splits	# Entities	# Relation Types	# Observable Triplets	# Missing Triplets
Train	1316	29	13630	781
Test	656	29	7257	184

of the training attributed graph but permuted. In test, we only mask out triplets corresponding to the (permuted) relation types `mother_of` and `father_of`, so that only one predictive pattern is required to accurately predict these missing triplets at test time.

Table 5 shows the statistics of the MetaFam dataset. The experiment results are described in Appendix E.2.

D Implementation and Experiment Details

D.1 Licenses, Computational Resources and Experimental Setup

We implemented our MTDEA model using PyTorch [31] and PyTorch Geometric [14], which are available under the BSD and MIT license respectively. The Wikidata knowledge base [43], which the WikiTopics dataset is based on, is available under the CC0 1.0 license. We ran our experiments on NVIDIA V100, A100, GeForce RTX 2080Ti, GeForce RTX 4090, and Titan V GPUs. We use Weights & Biases [5] to perform hyperparameter tuning. We train all models (baselines and MTDEA) in all experiments for a maximum of 10 epochs, with an early stop patience of 5 epochs based on the dual-sampling MRR value on the validation set. At test time we adapt our MTDEA models to learn the task assignments for the test relation types (as described in Section 5.4) for a maximum of 10 epochs. For our MTDEA models, we train with a number of maximum task partitions $\hat{K} = 2, 4, 6$ in all experiments. The time spent on each experiment depends mainly on the size of the dataset and on $\hat{K}$. For example, training MTDEA with $\hat{K} = 4$ on WikiTopics-MT3 takes around 10 hours to complete, while training MTDEA with $\hat{K} = 2$ on WikiTopics-MT takes around 2 hours and 30 minutes. Our code and datasets are available. [3]

D.2 Details of the Neural Architecture

Attention matrix. In our implementation, the attention matrix α (Section 5.2) is obtained from a real-valued learnable weight matrix $w \in \mathbb{R}^{R \times \hat{K}}$, where R is the number of relations and $\hat{K}$ the number of maximum partitions we allow. We apply a Softmax activation over the task partition dimension for every relation type $r \in \mathcal{R}$, i.e., $\alpha_{r,k} = \frac{\exp(w_{r,k})}{\sum_{k'=1}^{\hat{K}} \exp(w_{r,k'})}$, for $k \in \{1, \dots \hat{K}\}$. At training time we refer to w as $w^{(\text{tr})}$, since R is $R^{(\text{tr})}$, the number of training relation types. During the test-time adaptation, we freeze all parameters of the model, we discard $w^{(\text{tr})}$ and initialize a new matrix $w^{(\text{adapt})} \in \mathbb{R}^{R^{(\text{te})} \times \hat{K}}$, where $R^{(\text{te})}$ is the number of test relation types. Then, $w^{(\text{adapt})}$ is optimized via gradient descent with the same training loss used in training (Equation (6)), with the only difference that the positive and negative triplets are now sampled from the observable test graph $\mathbf{A}^{(\text{te})}$.

Structural node representation for link prediction tasks. Structural node representations, which are obtained from GNNs, are known to have limited capabilities for link prediction tasks in homogeneous graphs [37, 51], an issue that also arises in attributed graphs. Theoretically, Equation (4) overcomes this limitation by employing GNNs that output pairwise-representations as $L_1^{(t)}, L_2^{(t)}, L_3^{(t)}$. However, most-expressive pairwise representations [54, 58, 55] are computationally expensive. In Gao et al. [18], the authors sought a middle ground for their ISDEA model by employing structural node representations enhanced with heuristic embeddings, such as the shortest distances between the two nodes in the pair to be predicted. Specifically, the representation for a triplet (u, r, v), with $u, v \in \mathcal{V}, r \in \mathcal{R}$, before the final MLP layers is obtained as $h_{u,r}^{(T)} \| h_{v,r}^{(T)} \| d(u, v) \| d(v, u)$, where $h_{u,r}^{(T)}$

[3] `https://anonymous.4open.science/r/MTDEA.`

and $h_{u,r}^{(T)}$ are the structural node representations for, respectively, nodes u and v specific to the relation type r obtained after T ISDEA layers; $d(u,v)$ and $d(v,u)$ are the shortest distances from node u to v and from node v to u in the directed attributed graph, and $\|$ denotes the vector concatenation operation.

Nevertheless, computing the shortest distance, $d(u,v)$, for all pairs (u,v), $u,v \in \mathcal{V}$ in the attributed graph is time- and space-demanding. Due to this limitation, we opt *not* to compute the shortest distances, and instead use only the structural node representations as the representation of a triplet (u,r,v), that is $h_{u,r}^{(T)} \| h_{v,r}^{(T)}$, in all our experiments, except for the synthetic METAFAM dataset. In practice, this means that we implement $L_1^{(t)}, L_2^{(t)}, L_3^{(t)}$ in Equation (4) as GNNs outputting node representations. We empirically observe no performance degradation when removing the shortest distance heuristics on real-world attributed graphs.

Layers. In all our experiments, excluding those on the synthetic METAFAM dataset, our MTDEA model employs two GNN-based soft MTDE linear layers (Equation (4)). Conversely, for the synthetic METAFAM dataset, our model consists of only one GNN-based MTDE linear layer, in order to learn exactly the conflicting predictive patterns depicted in Figure 2. We select GIN [48] with $\epsilon = 0$ as our GNN layer, which implements the L_1, L_2, and L_3 components of a MTDE linear layer.

After these GNN-based MTDE linear layers, we employ two soft MTDE linear layers with MLPs L_1, L_2, and L_3 components. The representation of each triplet (u,r,v) with $u,v \in \mathcal{V}$, $r \in \mathcal{R}$ is then obtained using the node representations after these layers as $h_{u,r}^{(T)} \| h_{v,r}^{(T)}$, and it is then passed to a two-layers MLP to obtain the final prediction.

D.3 Hyper-parameters

In all experiments, involving either our MTDEA models or the baseline ISDEA-homo, we train with mini-batches comprising 256 positive triplets. We use a training negative sample rate of 2 for both the tail-based negative samples and the relation-based negative samples. Hence, for each positive triplet in a minibatch, we construct four negative samples, thus resulting in mini-batches containing 1280 triplets in total. Additional hyper-parameters values for MTDEA and ISDEA-homo include hidden layer dimension of 32, ReLU activations, mean set aggregation (i.e. the set aggregation operation within the parentheses of L_2 and L_3 in Equation (2)), a gradient clipping norm of 1.0, a learning rate of 0.001, and a weight decay rate of 5×10^{-4} in our Adam optimizer.

Apart from the aforementioned hyper-parameters, our MTDEA features additional hyper-parameters associated with its regularization losses. In all our experiments, we set the initial regularization coefficient values to $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$ (Equation (6)), with a per-epoch multiplicative annealing factor of 1.1. This approach ensures that the regularization gains more significance in later epochs (closer to the end). That is, after each epoch, the values of λ_1 and λ_2 are updated to $1.1 \times \lambda_1$ and $1.1 \times \lambda_2$ respectively.

For the NBFNet-homo baseline, we use the default hyper-parameters provided by Zhu et al. [58]. Specifically, we choose a hidden dimension of 32 for all layers, the distance multiplier message function, the pna aggregate function, and we employ layer norm. We further use Adam optimizer with a learning rate of 0.005 and a batch size of 64.

E Additional Experiments

E.1 Dual-sampling versus Entity-centric Metrics

In both our training loss (Equation (5)) and evaluation metrics (Section 6), we adopt a dual-sampling scheme wherein for each positive triplet we draw two different types of negative samples: those with the tail node randomly corrupted (entity-centric) and those with the relation type randomly corrupted (relation-type centric). In contrast, existing literature focuses solely on entity-centric negative samples [49, 36, 58]. We argue that correctly predicting the tail node, given a head node and a relation type, is as important as determining the type of relation connecting a head and a tail node of interest, and therefore the two negative sampling schemes should be used in conjunction. This combination results in the dual-sampling metrics we propose.

Table 6: Model performances on WIKITOPICS-MT under the **dual-sampling metrics**, tested on two topics (HEALTH and TAXONOMY) not seen in training (ART + PEOPLE). We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2).

	Test on HEALTH topic				Test on TAXONOMY topic			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	15.843 (0.119)	0.122 (0.001)	0.041 (0.000)	0.339 (0.003)	16.260 (0.068)	0.113 (0.000)	0.034 (0.000)	0.315 (0.001)
ISDEA-homo	42.276 (0.768)	0.025 (0.001)	0.000 (0.000)	0.000 (0.000)	40.963 (1.276)	0.026 (0.001)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	15.405 (6.030)	0.384 (0.133)	0.323 (0.140)	0.481 (0.138)	19.143 (3.895)	0.323 (0/063)	0.269 (0.063)	0.365 (0.082)
MTDEA ($\hat{K}=2$)	15.015 (4.947)	0.422 (0.170)	0.358 (0.191)	0.513 (0.112)	16.840 (4.222)	0.393 (0.141)	0.330 (0.157)	0.457 (0.121)
MTDEA ($\hat{K}=4$)	16.576 (2.293)	0.441 (0.115)	0.390 (0.127)	0.496 (0.108)	18.171 (4.242)	0.365 (0.191)	0.307 (0.203)	0.417 (0.175)
MTDEA ($\hat{K}=6$)	**13.774 (3.224)**	**0.504 (0.013)**	**0.457 (0.012)**	**0.555 (0.010)**	**16.902 (3.837)**	**0.470 (0.018)**	**0.422 (0.010)**	**0.504 (0.025)**

Table 7: Model performances on WIKITOPICS-MT under the **entity-centric metrics**, tested on two topics (HEALTH and TAXONOMY) not seen in training (ART + PEOPLE). We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2).

	Test on HEALTH topic				Test on TAXONOMY topic			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	**5.767 (0.238)**	**0.628 (0.004)**	**0.568 (0.004)**	**0.858 (0.017)**	**5.980 (0.149)**	**0.572 (0.001)**	**0.498 (0.002)**	**0.855 (0.003)**
ISDEA-homo	32.793 (1.683)	0.112 (0.035)	0.054 (0.025)	0.172 (0.065)	30.065 (2.717)	0.108 (0.051)	0.039 (0.028)	0.203 (0.124)
IS-DEA [18]	19.538 (8.350)	0.364 (0.136)	0.314 (0.142)	0.444 (0.140)	24.138 (4.299)	0.301 (0.059)	0.252 (0.063)	0.336 (0.071)
MTDEA ($\hat{K}=2$)	19.707 (6.651)	0.422 (0.166)	0.360 (0.185)	0.486 (0.138)	23.146 (7.251)	0.368 (0.149)	0.300 (0.158)	0.457 (0.119)
MTDEA ($\hat{K}=4$)	20.316 (5.700)	0.442 (0.122)	0.386 (0.131)	0.494 (0.115)	22.408 (6.274)	0.375 (0.132)	0.310 (0.141)	0.449 (0.124)
MTDEA ($\hat{K}=6$)	14.920 (1.523)	0.513 (0.013)	0.455 (0.023)	0.570 (0.007)	19.405 (0.518)	0.453 (0.002)	0.392 (0.002)	0.524 (0.002)

Table 8: Model performances on FBNELL under the **dual-sampling metrics** (left) and the **entity-centric metrics** (right). We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2).

	Dual-Sampling Metrics				**Entity-Centric Metrics**			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	9.410 (0.029)	0.129 (0.001)	0.042 (0.001)	0.379 (0.002)	**2.501 (0.041)**	**0.818 (0.008)**	**0.782 (0.010)**	**0.950 (0.001)**
ISDEA-homo	31.625 (0.940)	0.033 (0.001)	0.000 (0.000)	0.000 (0.000)	9.424 (1.915)	0.373 (0.028)	0.235 (0.025)	0.700 (0.067)
IS-DEA [18]	10.925 (0.383)	**0.624 (0.010)**	**0.562 (0.014)**	0.697 (0.010)	10.924 (0.893)	0.611 (0.013)	0.540 (0.009)	0.712 (0.030)
MTDEA ($\hat{K}=2$)	**9.106 (0.162)**	0.622 (0.012)	0.553 (0.010)	**0.704 (0.024)**	10.797 (0.920)	0.602 (0.011)	0.524 (0.010)	0.707 (0.028)
MTDEA ($\hat{K}=4$)	10.730 (0.666)	0.606 (0.006)	0.543 (0.007)	0.680 (0.023)	10.464 (0.092)	0.613 (0.011)	0.540 (0.015)	0.720 (0.007)
MTDEA ($\hat{K}=6$)	10.386 (0.683)	0.609 (0.015)	0.547 (0.017)	0.678 (0.008)	10.753 (0.828)	0.612 (0.005)	0.538 (0.008)	0.719 (0.021)

Tables 6 to 8 compare the performances of various models under the dual-sampling metrics with their performances under the traditional entity-centric metrics, commonly used in the literature. As can be seen from the tables, the baseline NBFNet-homo demonstrates strong performances under the entity-centric metrics, but not under the dual-sampling metrics. In particular, it achieves 95% entity-centric Hits@10 on the FBNELL dataset. Therefore *a model that disregards the information contained in the relation types can achieve near-perfect accuracy under the entity-centric metrics* (simply by predicting whether u is connected to v, irrespective of the relation type). These results suggest that the entity-centric metrics are insufficient for assessing model performance, as homogeneous link prediction methods can easily solve a task when evaluated based on these metrics. In contrast, the homogeneous models achieve at most 38% Hits@10 accuracy under the dual-sampling metrics, illustrating that the dual-sampling scheme is a more suitable, comprehensive, and challenging evaluation scheme, where homogeneous link prediction models cannot unreasonably obtain near-perfect performances.

E.2 Synthetic Experiments

We conduct additional experiments using the dataset METAFAM, which was explicitly constructed to exhibit conflicting predictive patterns, or multiple tasks, in the attributed graphs (Appendix C.3). Table 9 shows the results under the dual-sampling metrics. As we can see from the table, our MTDEA model with two task partitions $\hat{K} = 2$ consistently obtains the best performance under all the metrics. This observation conforms to our expectation, since the METAFAM was constructed to include exactly

Table 9: Model performances on MetaFam under the **dual-sampling metrics**. We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2). *Our MTDEA is the only model that can correctly represent the existing conflicting patterns.*

Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@5 ↑	Hits@10 ↑
NBFNet-homo	13.785 (0.045)	0.153 (0.001)	0.068 (0.001)	0.145 (0.002)	0.216 (0.001)	0.400 (0.001)
ISDEA-homo	27.369 (0.027)	0.037 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	10.112 (1.120)	0.292 (0.029)	0.145 (0.025)	0.323 (0.038)	0.440 (0.044)	0.609 (0.050)
MTDEA ($\hat{K} = 2$)	**9.763 (4.246)**	**0.344 (0.067)**	**0.178 (0.063)**	**0.407 (0.068)**	**0.518 (0.083)**	**0.704 (0.072)**
MTDEA ($\hat{K} = 4$)	13.795 (3.966)	0.172 (0.134)	0.070 (0.114)	0.150 (0.173)	0.213 (0.174)	0.358 (0.199)
MTDEA ($\hat{K} = 6$)	11.332 (0.894)	0.169 (0.057)	0.031 (0.029)	0.148 (0.110)	0.272 (0.161)	0.520 (0.117)

Table 10: Model performances on MetaFam under the **entity-centric metrics**. We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2).

Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@5 ↑	Hits@10 ↑
NBFNet-homo	**1.114 (0.071)**	**0.952 (0.026)**	**0.939 (0.032)**	**0.987 (0.010)**	**0.998 (0.002)**	**1.000 (0.000)**
ISDEA-homo	1.742 (0.056)	0.741 (0.031)	0.558 (0.056)	0.936 (0.004)	0.991 (0.002)	0.998 (0.003)
IS-DEA [18]	1.663 (0.069)	0.757 (0.015)	0.580 (0.021)	0.941 (0.020)	0.996 (0.006)	**1.000 (0.000)**
MTDEA ($\hat{K} = 2$)	4.226 (4.363)	0.697 (0.082)	0.527 (0.052)	0.870 (0.118)	0.913 (0.139)	0.919 (0.140)
MTDEA ($\hat{K} = 4$)	1.812 (0.032)	0.710 (0.021)	0.505 (0.046)	0.935 (0.014)	0.989 (0.005)	**1.000 (0.000)**
MTDEA ($\hat{K} = 6$)	1.685 (0.123)	0.748 (0.039)	0.564 (0.063)	0.950 (0.010)	0.995 (0.004)	**1.000 (0.000)**

Table 11: Model performances on MetaFam under the **relation-type centric metrics**. We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denote the maximum number of tasks the architecture can model (Section 5.2).

Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@5 ↑	Hits@10 ↑
NBFNet-homo	51.000 (0.000)	0.020 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
ISDEA-homo	51.000 (0.000)	0.020 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	18.043 (1.957)	0.224 (0.006)	0.117 (0.009)	0.224 (0.012)	0.330 (0.044)	0.473 (0.037)
MTDEA ($\hat{K} = 2$)	**15.001 (4.420)**	**0.279 (0.062)**	**0.163 (0.057)**	**0.294 (0.076)**	**0.387 (0.082)**	**0.545 (0.069)**
MTDEA ($\hat{K} = 4$)	25.124 (7.784)	0.126 (0.126)	0.059 (0.102)	0.101 (0.150)	0.140 (0.169)	0.233 (0.179)
MTDEA ($\hat{K} = 6$)	20.160 (1.464)	0.110 (0.041)	0.023 (0.017)	0.067 (0.056)	0.128 (0.096)	0.315 (0.127)

two conflicting predictive patterns and therefore a model capable of modeling two distinct tasks is expected to obtain the best predictions in this dataset.

We further investigate the performances of the models under different metrics. We consider the entity-centric metrics, where we generate 50 negative samples for each positive triplet by corrupting its tail node, and we additionally compare to what we refer as the relation-type centric metrics, obtained by constructing 50 negative samples for each positive triplet by corrupting its relation type. These results are summarized in Tables 10 and 11. We observe that, although our best-performing model (MTDEA with $\hat{K} = 2$) is slightly worse than the baseline ISDEA under the entity-centric metrics, it outperforms all the models under the relation-based metrics.

E.3 More WikiTopics-MT Scenarios

Tables 12 to 14 show the additional experiment results on multi-task scenarios WikiTopics-MT2, WikiTopics-MT3, and WikiTopics-MT4. In most cases, our MTDEA model outperforms the baseline models on the MRR and Hits@1, while being comparable in other metrics.

Table 12: Model performance on WikiTopics-MT2 under the **dual-sampling metrics**, tested on two topics (Location and Science) not seen in training (Sport + Health). We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denotes the maximum number of tasks the architecture can model (Section 5.2).

	Test on Location topic				Test on Science topic			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	18.231 (1.027)	0.109 (0.004)	0.035 (0.002)	0.292 (0.016)	19.525 (0.908)	0.091 (0.002)	0.024 (0.001)	0.235 (0.008)
IS-DEA-homo	39.716 (1.022)	0.027 (0.001)	0.000 (0.000)	0.000 (0.000)	37.581 (2.115)	0.028 (0.001)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	**14.873 (1.270)**	0.462 (0.031)	0.393 (0.038)	**0.569 (0.005)**	**12.991 (1.067)**	0.473 (0.024)	0.390 (0.034)	**0.617 (0.023)**
MTDEA ($\hat{K} = 2$)	15.619 (1.262)	0.480 (0.178)	0.417 (0.023)	0.557 (0.014)	13.795 (3.410)	0.482 (0.007)	0.406 (0.008)	0.595 (0.027)
MTDEA ($\hat{K} = 4$)	16.864 (0.704)	0.470 (0.042)	0.405 (0.049)	0.547 (0.037)	13.302 (1.518)	**0.483 (0.002)**	**0.409 (0.004)**	0.590 (0.010)
MTDEA ($\hat{K} = 6$)	16.362 (0.933)	**0.490 (0.001)**	**0.431 (0.004)**	0.558 (0.014)	14.531 (1.121)	0.476 (0.007)	0.405 (0.012)	0.580 (0.024)

Table 13: Model performance on WikiTopics-MT3 under the **dual-sampling metrics**, tested on two topics (Art and Infrastructure) not seen in training (People + Taxonomy). We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denotes the maximum number of tasks the architecture can model (Section 5.2).

	Test on Art topic				Test on Infrastructure topic			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	19.495 (0.447)	0.099 (0.002)	0.030 (0.001)	0.257 (0.006)	15.622 (0.174)	0.137 (0.002)	0.056 (0.001)	0.357 (0.004)
IS-DEA-homo	37.31 (0.5739)	0.028 (0.000)	0.000 (0.000)	0.000 (0.000)	35.456 (1.634)	0.029 (0.001)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	**12.981 (0.753)**	0.475 (0.049)	0.385 (0.062)	**0.638 (0.035)**	12.438 (0.904)	0.461 (0.011)	0.304 (0.008)	**0.696 (0.011)**
MTDEA ($\hat{K} = 2$)	13.388 (1.928)	0.464 (0.021)	0.376 (0.030)	0.614 (0.041)	11.887 (1.314)	0.456 (0.013)	0.351 (0.010)	0.664 (0.008)
MTDEA ($\hat{K} = 4$)	14.185 (1.181)	**0.488 (0.002)**	**0.405 (0.010)**	0.598 (0.004)	**10.224 (0.917)**	**0.466 (0.007)**	**0.353 (0.006)**	0.685 (0.012)
MTDEA ($\hat{K} = 6$)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM

Table 14: Model performance on WikiTopics-MT4 under the **dual-sampling metrics**, tested on two topics (Health and Science) not seen in training (Location + Organization). We report mean and std across 3 random seeds. For our MTDEA, $\hat{K}$ denotes the maximum number of tasks the architecture can model (Section 5.2).

	Test on Health topic				Test on Science topic			
Models	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑	MR ↓	MRR ↑	Hits@1 ↑	Hits@10 ↑
NBFNet-homo	17.083 (0.070)	0.124 (0.001)	0.046 (0.001)	0.328 (0.002)	19.998 (0.081)	0.094 (0.001)	0.026 (0.001)	0.246 (0.002)
IS-DEA-homo	32.286 (0.731)	0.032 (0.001)	0.000 (0.000)	0.000 (0.000)	36.882 (4.891)	0.029 (0.004)	0.000 (0.000)	0.000 (0.000)
IS-DEA [18]	7.199 (1.047)	**0.681 (0.002)**	**0.581 (0.003)**	0.819 (0.009)	6.696 (1.160)	0.707 (0.012)	0.612 (0.019)	**0.847 (0.012)**
MTDEA ($\hat{K} = 2$)	6.990 (0.686)	0.678 (0.015)	**0.581 (0.015)**	0.803 (0.029)	6.477 (0.464)	0.704 (0.004)	0.615 (0.005)	0.827 (0.005)
MTDEA ($\hat{K} = 4$)	**6.387 (1.531)**	0.680 (0.011)	0.575 (0.013)	**0.827 (0.019)**	**5.994 (0.053)**	**0.715 (0.011)**	**0.634 (0.011)**	0.827 (0.021)
MTDEA ($\hat{K} = 6$)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM

F Time and Space Complexity

In this section we analyze the complexity of our model, focusing on Equation (4). We assume $L_1^{(t)}, L_2^{(t)}, L_3^{(t)} : \mathbb{R}^{N \times N \times d} \to \mathbb{R}^{N \times N \times d'}$ to be GNNs that output *node representations* instead of pairwise representations, which is the setup we adopt in our experimental evaluation, as described in Appendix D. We consider the feature dimension to be a constant.

For input graph with N nodes and R relation types, denote by Δ the maximum node degree, and let $\hat{K}$ be the maximum number of tasks our architecture can model. The time complexity of the $L_1^{(t)}$ and $L_2^{(t)}$ components in Equation (4) is $\mathcal{O}(RN\Delta_{\max})$, as each of the R relation types is processed using a standard GNN, which has time complexity $\mathcal{O}(N\Delta_{\max})$. The time complexity of the $L_3^{(t)}$ component in Equation (4) is $\mathcal{O}(R\hat{K}N\Delta_{\max})$, since each of the R relation types iterates over the $\hat{K}$ tasks and for each of them aggregates all other relation types and processes the aggregation using a standard GNN, which has time complexity $\mathcal{O}(N\Delta_{\max})$. Therefore, our method, as described in Equation (4), has an overall time complexity $\mathcal{O}(R\hat{K}N\Delta_{\max})$. In practice, $\hat{K}$ is small compared to R and N (the maximum value we consider in our experiments is $\hat{K} = 6$).

We note that the complexity of our method can be reduced if we replace the set aggregation inside $L_3^{(t)}$ to avoid excluding the current relation type. That is, if we substitute $\sum_{r'' \in \mathcal{R} \setminus \{r\}} \alpha_{r'',k} \mathbf{H}^{(t)}_{\cdot,r'',\cdot,\cdot}$.

with $\sum_{r''\in\mathcal{R}} \alpha_{r'',k}\mathbf{H}^{(t)}_{\cdot,r'',\cdot,\cdot}$, then for each of the $\hat{K}$ tasks, the output of $L_3^{(t)}$ can be computed only once, instead of computing it for each relation type. Therefore, the overall time complexity of our method can be improved to $\mathcal{O}(RN\Delta_{\max})$ by a simple change in the set aggregation function.

The space complexity of our method is $\mathcal{O}(R(N + N\Delta) + R\hat{K})$, as for each relation type we need to store N node features and its connectivity, as well as the attention weights $\alpha \in [0, 1]^{R\times\hat{K}}$.

G Limitations

Despite the contributions and advancements made in this work, there are aspects that can be further refined and explored in future works:

- **Scalability:** The proposed model may face challenges when scaling up to extremely large graphs, as memory demands might become prohibitively high. Further research is necessary to develop approximation techniques to handle such large-scale applications.
- **Model complexity:** The proposed model introduces additional complexity compared to some baseline methods. Efforts to simplify the models while preserving their performance benefits are worth exploring in future research.
- **Non-exchangeable relations:** There may exist cases where no relations are exchangeable, rendering it necessary to have a number of task partitions equal to the number of relations. In such situations, the benefits of our proposed method may be reduced. Investigating these cases remains an important avenue for future research.