INFORMATION-BASED INVARIANCE, EQUIVARIANCE, AND EXPRESSIVITY FOR GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNNs) leverage the topology of graphs to learn informative representations. It is known that the same information is often represented in nonisomorphic forms. In this paper, we postulate that GNNs should be invariant/equivariant and expressive with respect to these variations. We introduce the notions of invariance/equivariance of GNNs with respect to these variations and prove that current GNN architectures are not invariant/equivariant to these variations. We propose a general approach to modify a GNN to make it invariant/equivariant to these variations. We also generalize the current notion of expressivity to consider these variations and investigate the expressivity of the invariant/equivariant GNNs. Our empirical studies using multiple GNN architectures over real-world graphs indicate that our approach achieves perfect invariance/equivariance and improves the accuracy of current GNNs on average without imposing significant time overhead.

1 INTRODUCTION

Graph data from a wide variety of fields such as chemistry, biology, sociology, transportation, and computer science are now readily available. With this abundance of graph data, there has been great interest in analyzing it for useful patterns, leading to a flurry of activity in the area of graph learning and in particular Graph Neural Networks (GNNs) (Hamilton, 2020). Learning representations for graphs is closely connected to the problem of graph isomorphism. Naturally, we want the learned representations to be invariant (or equivariant) under graph isomorphisms (Masci et al., 2016; Hamilton, 2020; Huang et al., 2023a) and, ideally, to be fully expressive—assigning distinct representations to non-isomorphic graphs (Morris et al., 2019; Xu et al., 2019). Consequently, many graph representation algorithms, such as GNNs, can be viewed as heuristics for graph isomorphism. In particular, message passing neural networks (MPNNs) are closely related to the well-known Weisfeiler-Leman (WL) test for graph isomorphism.

However, in several types of graphs, such as knowledge graphs and biological networks, we need to work with a more general notion of equivalence than graph isomorphism—one that renders much of the theory above not immediately applicable (Skavantzos & Link, 2023; Egger et al., 2026; Fan & Bohannon, 2008; Arenas, 2006; Gutierrez et al., 2004; Picado et al., 2017; Silva et al., 2007; Trivodaliev et al., 2014; Doyle & Snell, 1984; Truemper, 1989; Wood et al., 2021). Here, two graphs are considered equivalent if they represent the same information, even if they are far from being isomorphic. Figure 1 shows two non-isomorphic knowledge graphs that represent essentially the same information, i.e., relationships between a set of objects, in non-isomorphic structures. The prevalence of non-isomorphic but informationally equivalent graphs is well established in knowledge and data management. For example, graph data can be represented in equivalent but non-isomorphic variations using a process called normalization/denormalization (Skavantzos & Link, 2023; Egger et al., 2026; Skavantzos & Link, 2025; Arenas, 2006; Yu & Jagadish, 2008). Due to their prevalence,



(a) Fragments of a knowledge graph **(b)** Mappings of the graph fragments in (a)

Figure 1: Example of two knowledge graphs whose node labels are *year*, *paper*, and *venue*.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

these mappings are taught in undergraduate/graduate knowledge management courses. As another example, organic compounds, e.g., ADP and ATP, are sometimes represented as vertices and other times as edges in metabolic networks (Silva et al., 2007), resulting in non-isomorphic graphs with the same information. Similar problems appear in encode protein annotations (Trivodaliev et al., 2014). In the field of circuit design where circuits are represented as graphs, certain transformations of circuit graphs, such as Δ-Y or generally star-mesh, result in non-isomorphic but equivalent circuit graphs (Doyle & Snell, 1984; Truemper, 1989). Generally, the AI and knowledge representation communities have long recognized that the same information can appear in non-isomorphic structures (Jaeger, 1996; Halpern & Koller, 2004; Codd, 1971).

In this paper, we introduce the notions of invariance/equivariance and expressivity for GNNs with respect to these information-based equivalences. We show that the current popular GNN architectures are not invariance/equivariance to this type of equivalence. Thus, GNNs may deliver accurate results in some styles of graph representation and perform poorly in others. As GNNs applications grow rapidly, they will face more variations in graph representations, increasing the impact of this shortcoming on the effectiveness of GNNs. Because graph learning spans diverse settings and types of architectures, designing invariant/equivariant architectures for each is impractical. Instead, we build theoretically grounded plug-in tools that make current GNNs invariant/equivariant with respect to these equivalences by relatively small changes to their architectures. We prove that these changes are the minimal possible changes to a GNN to achieve this goal. We study the expressivity of the resulting architectures and prove that our changes do not reduce the expressivity of GNNs and strictly increase their expressivity in some cases. Our extensive empirical studies using multiple architectures over real-world graphs indicate that our approach achieves perfect invariance/equivariance and improves the accuracy of current GNNs on average without imposing significant time overhead.

2 Information Equivalence

Basics. Let $\mathcal{V} \subset \mathbf{N}$ be countably finite sets of node ids. A *directed graph* G on $\mathcal{V}$ is a tuple (V, E) where $V \subset \mathcal{V}$ ($V \neq \emptyset$) and $E \subseteq V \times V$. By abuse of notation, we show the set of nodes and edges of G as $V(G)$ and $E(G)$, respectively. Each node in a graph is associated with a single member in a finite set of *labels* $\mathcal{L}$. Such labels are commonly available in our applications of interest, particularly in knowledge and biological graphs. For the rest of the paper, we assume that the nodes have always labels (labeled graphs). Our results also apply to graphs without labels by assuming all nodes have the same label. We denote the edge from node u with the label L_1 to node v with the label L_2 as $(L_1(u), L_2(v))$. We also associate each node with a vector in $\mathbb{R}^d$, i.e., its features. The graph $G = (V, E)$ is undirected provided that if $(u, v) \in E$, then $(v, u) \in E$. In this paper, we assume that the graphs are directed. Our results extend to undirected graphs. Two graphs are **isomorphic** if there is a bijection between their nodes that preserves the features and labels of each node and the edges between nodes. Given the graphs P and G, a **homomorphism** from P to G is a mapping from nodes in P to the nodes G that preserves node labels and adjacency (edges). Each *occurrence* of P in G is the homomorphism image of P in G. We use $hom(P, G)$ to denote the set of occurrences of P in G.

Mappings. Let $\mathcal{G}$ denote the family of graphs and let $\tau : \mathcal{G} \to \mathcal{G}$ be a bijection. A graph $G \in \mathcal{G}$ is called **τ-equivalent** to a graph $H \in \mathcal{G}$ if $G = \tau(H)$. An example of bijection mapping is *permutation*: let $\mathcal{G}^n$ be the set of graphs with n nodes. Let $[n]$ denote the set $\{1 \dots n\}$. Without loss of generality, we assume that for graph $G \in \mathcal{G}^n$, we have $V = [n]$. The permutation $\pi : \mathcal{G}^n \to \mathcal{G}^n$ establishes a bijection from $[n]$ to $[n]$ on the node ids, which preserves the node labels, features, and edges between the nodes, i.e., G and $\pi(G)$ are isomorphic. The τ-equivalence captures equivalence beyond permutation. For example, let $\mathcal{G}$ be the set of all (finite) graphs with a single label l. Consider the mapping δ that is a bijection on nodes and preserves the node ids,labels, and features and for every two nodes $v, u \in V(G)$ if $(l(v), l(u)) \in E(G)$, we have $(l(u), l(v)) \in E(\delta(G))$, i.e., δ reverses the directions of the edges of G. δ is bijective since for each G there is one and only one $\tau(G)$ so G and $\delta(G)$ are δ-equivalent. Similarly, the mapping δ' in the set of all graphs with labels $\{l_1, l_2\}$ which is a bijection on the nodes and preserves their id/label/features and reverses the edges from the nodes with the label l_1 to the nodes with the label l_2 and retains all other edges and their directions is bijective.

Example 1. *Figure 1 shows fragments of two graphs on the relationships between papers, venues, and their years. The graph in Figure 1(a) connects the year of a venue to the papers published in the venue, while the graph in Figure 1(b) connects these years directly to the venues. Intuitively, the information represented in each graph can be recovered from the other one. More precisely, let $\mathcal{G}$*

be the set of all connected graphs with labels $\{paper, venue, year\}$ with at least one node from each label. Let $\beta : \mathcal{G} \to \mathcal{G}$ map each node $v \in V(G)$ to one and only one node $\beta(v) \in V(\beta(G))$ such that v and $\beta(v)$ have the same node id/label/features. For every $u, v, w \in V(G)$ if we have edges $(year(u), paper(v))$ and $(paper(v), venue(w))$ in G, β maps them to $(year(\beta(u)), venue(\beta(w)))$ and $(paper(\beta(v)), venue(\beta(w)))$. β preserves all other edges between $v \in V(G)$ and $\beta(v)$ that do not satisfy the preceding condition in G. It is easy to see that β is bijective.

Representing Mappings. Since mappings often modify the structure of the graph, we are interested in the case where mappings are defined via a collection of **switchings** specified by pairs of labeled (pattern) graphs, $\tau = \{(P_i, P_i')\}_{i=1}^{t}$, where there is a bijection between the nodes in P_i and P_i' that preserves the node id and label. Then $\tau(G)$ is obtained by replacing all occurrences of P_i in G with P_i' for every $i \in [t]$. More precisely, graph G is mapped to graph H by switching (P_i, P_i') if and only if there is bijection b_i between the sets of subgraphs $hom(P_i, G)$ and $hom(P_i', H)$ such that for every $g \in hom(P_i, G)$, there is a bijection between nodes of g and $b_i(g)$ that preserve node ids, labels, and features. Given mapping $\tau = \{(P_i, P_i')\}_{i=1}^{t}$, the graph G is mapped to graph H using τ iff G is mapped to H using every (P_i, P_i') and each node in G (H) appears in at least one occurrence of a graph P_i (P_i'). For example, the mapping δ explained above, which reverses the directions of the edges of a graph, is represented by a single switching (P_1, P_1') where P_1 is $(l(v), l(u))$ and P_1' is $(l(u), l(v))$. We can also define a set of switchings for the mapping δ' described above in the set of graphs with the node labels l_1 and l_2 by the set of switchings $\{(P_i, P_i')\}_{i=1}^{4}$ where P_1 is $(l_1(v), l_2(u))$, P_1' is $(l_2(u), l_1(v))$ and other switchings retain all other edges, e.g., $(l_1(v), l_1(u))$ to $(l_1(v), l_1(u))$.

Example 2. *Consider the graphs in Figure 1 and the mapping β in Example 1. A switching in β is (P_1, P_1'), where P_1 is the graph with edges $(year(u), paper(v))$ and $(paper(v), venue(w))$ and P_1' is the graph with edges $(year(u), venue(w))$ and $(paper(v), venue(w))$.*

Proposition 1. *Every set of switchings over a family of graphs $\mathcal{G}$ defines a unique bijection on $\mathcal{G}$.*

Intuitively, the condition of mappings under switching ensures that the nodes in the occurrences of two distinct patterns P_i and P_j of τ in $G \in \mathcal{G}$ are mapped to same nodes in $\tau(G)$ and avoids creating non-recoverable duplicates. Switchings arise frequently in graph applications and theory: Δ–Y transforms (Doyle & Snell, 1984; Truemper, 1989), Whitney twisting (Whitney, 1933a;b; Moffatt & Oh, 2021; Smith et al., 2019), Seidel switching (Seidel, 1976; VAN LINT & SEIDEL, 1991), Godsil–McKay switching (yielding cospectral graphs) (Godsil & McKay, 1982; Haemers, 1995), and edge flips in triangulations. Also, normalization/denormalization mappings can be expressed using switchings (Arenas, 2006; Yu & Jagadish, 2008). They also appear in biological knowledge bases (Wood et al., 2021). We have the following for the first components of distinct switchings in τ.

Proposition 2. *Given bijective mapping $\tau = \{(P_i, P_i')\}_{i=1}^{t}$, for all $i \neq j$, if P_i is isomorphic to P_j, then P_i' is isomorphic to P_j'.*

Switchings $\tau = \{(P_i, P_i')\}_{i=1}^{t}$ is *minimal* if there is no switching $i \neq j$ where there are isomorphisms between P_i (P_i') and P_j (P_j') that preserve labels. In this paper, we consider only minimal switchings. We refer to bijective mappings as mappings and consider only mappings represented by switchings.

Composition of Mappings. Since compositions of mappings may generate new mappings, we are interested in invariance/equivaraince under the composition of mappings. Also, as permutations and (non-permutation) mappings preserve the information in the graph, we would like to design architectures that are invariant/equivariant to both permutations and sets of switchings.

Proposition 3. *Given mapping $\tau = \{(P_i, P_i')\}_{i=1}^{t}$ and $\sigma = \{(S_i, S_i')\}_{i=1}^{u}$ over a family of graphs $\mathcal{G}$, τ^{-1} and the composition of τ and σ ($\tau \circ \sigma$) can be expressed using sets of switchings.*

The proof for Proposition 3 also provides the methods to compute composition and the inverse. Given a set of mappings $\mathcal{T}$ that is closed under inverse, let $\Gamma^{\mathcal{T}}$ be a set of all mappings constructed from mappings in $\mathcal{T}$ using (finite) compositions. Let $\mathcal{G}^n$ denote the set of graphs with n nodes, and Π^n be the set of all permutations on $\mathcal{G}^n$. $\mathcal{G} \subseteq \mathcal{G}^n$ is *closed under isomorphism* if for every $G \in \mathcal{G}$ and for all H isomorphic to G, $H \in \mathcal{G}$.

Theorem 1. *Given an isomorphic-closed $\mathcal{G} \subseteq \mathcal{G}^n$ and a set of mappings $\mathcal{T}$ on $\mathcal{G}$, $\Gamma^{\mathcal{T}}$ and $\Phi_n^{\mathcal{T}} = \Gamma^{\mathcal{T}} \cup \Pi^n$ are groups with composition on $\mathcal{G}$.*

The action of group $\Phi_n^{\mathcal{T}}$ on the set $\mathcal{G}$ is a function from $\Phi_n^{\mathcal{T}}$ and $\mathcal{G}$ to $\mathcal{G}$. The orbit of a graph $G \in \mathcal{G}$ is the set of graphs obtained by the action of every member of $\Phi_n^{\mathcal{T}}$ to G. Graphs H and J are $\Phi_n^{\mathcal{T}}$*-equivalent* (equivalent) if they belong to the orbit of the same graph. Since $\Phi_n^{\mathcal{T}}$ is a group, $\Phi_n^{\mathcal{T}}$-equivalence is an equivalence relation in $\mathcal{G}$ and partitions $\mathcal{G}$ into equivalence classes.

Invariant DNNs are designed for a priorly known group of transformations in their domains, e.g., GNN for permutation or CNN for translation. Similarly, the mappings in our groups are acquired from the domain, e.g., the fact that the relationships between nodes of some types are symmetric will introduce the mapping that reverses the direction of the edges between those nodes in the graph. There are also methods to infer some mappings from the data (Skavantzos & Link, 2023; Egger et al., 2026; Skavantzos & Link, 2025; Arenas, 2006; Yu & Jagadish, 2008). In this paper, we introduce a modular and principled method to inject these domain invariances and knowledge in graph learning.

3 Invariance & Equivariance

Representation Learning. GNNs learn low-dimensional representations of graphs that should exhibit properties expressed in the graph. The learned representation is often expressed as $m > 0$-dimension feature vectors on the nodes in the graph. Given the graph $G = (V, E)$ of n nodes with a matrix representation that encodes its label and feature information $A \in \mathbb{R}^{n \times n \times (d+1)}$, we denote the learning of the representation by the function f from $\mathbb{R}^{n \times n \times (d+1)}$ to $\mathbb{R}^{n \times m}$. For each node $v \in V$ with id i, the row i in $\mathbb{R}^{n \times m}$ contains the learned representation for v. Sometimes, GNNs learn a scalar value for the input, e.g., graph classification so GNN is a function from $\mathbb{R}^{n \times n \times (d+1)}$ to $\mathbb{R}$.

τ-Invariance/Equivariance. The definition of mapping over graphs naturally extends to their matrix representations. Given the mapping τ, a function $f : \mathbb{R}^{n \times n \times (d+1)} \rightarrow \mathbb{R}^{n \times m}$ is **τ-invariant** if $f(\tau(A)) = f(A)$ for every $A \in \mathcal{R}^{n \times n \times d+1}$, i.e., the transformation of the input does not affect the learned representation. The definition naturally extends to the functions that map the input matrix to $\mathbb{R}$. Each permutation on the family of graphs with n nodes can be represented by a $1 \times n$ matrix (Maron et al., 2019a). Let matrix P represent the permutation mapping π. Let mapping $\tau = \pi \circ \rho$ where π and ρ are a permutation and a non-permutation mapping, respectively. The function $f : \mathbb{R}^{n \times n \times (d+1)} \rightarrow \mathbb{R}^{n \times m}$ is **τ-equivariant** if $f(\tau * A) = \pi * f(A)$ for every $A \in \mathcal{R}^{n \times n \times d}$, i.e., the output is transformed when the input is transformed. If f is both τ-invariant and τ-equivariant, we call f τ-invariant/equivariant. To simplify our exposition, unless otherwise noted, we state our definitions and results for invariance. They naturally extend to equivariance. Following the definition of mappings, if f is τ-invariant, it will be τ^{-1}-invariant. Since mappings are bijective, τ-invariance is preserved under the composition of mappings. Let τ be a mapping on an isomorphism-closed set $\mathcal{G} \subseteq \mathcal{G}^n$. Since permutations are bijective, if f is invariant with respect to permutation $\pi \in \Pi^n$ and τ-invariant, it is invariant over $\tau \circ \pi$. Generally, given group $\Phi_n^{\mathcal{T}}$ on $\mathcal{G}$, function f is $\Phi_n^{\mathcal{T}}$-invariant ($\Phi_n^{\mathcal{T}}$-equivariant) if it is τ-invariant (τ-equivariant) for all τ in $\Phi_n^{\mathcal{T}}$. Following the properties of invariance with respect to inverse and composition, f is $\Phi_n^{\mathcal{T}}$-invariant if it is τ-invariant for every τ in $\mathcal{T}$ and Π^n. Since prominent GNNs are invariant to permutations (Π^n) (Hamilton, 2020) and due to the properties of invariance under composition, to show that they are τ-invariant for $\tau \in \Phi_n^{\mathcal{T}}$, it is sufficient to show that they are τ-invariant to (non-permutation) mappings in $\mathcal{T}$.

τ-Invariance/Equivariance of Current Architectures. There are various neural architectures for learning graph representations (Ju et al., 2025). *Message passing neural networks* (MPNN) are a prominent architecture for GNN's that use the graph structure to learn representations (Hamilton, 2020). Let $h_v^{(l)}$ represent the feature of node v at layer l. Given GNN f, graph G, and $v \in V(G)$, let $\mathcal{M}^f(G, v)$ be a permutation equivariant function that returns a non-empty and proper subset of $V(G)$. Many node-centric architectures f extend Equation 3 by sharing information between each node $v \in V(G)$ and $\mathcal{M}^f(G, v)$ in each layer. Each layer of an MPNN is composed of two learnable functions, the permutation invariant aggregate function and the update function:

$$h_v^{(l+1)} = \text{UPDATE}\left(h_v^{(l)}, \text{AGGREGATE}\left(\{h_u^{(l)} : u \in \mathcal{M}^f(G, v)\}\right)\right). \tag{1}$$

$\mathcal{M}^f(G, v)$ is often defined using the structural properties of the connections between v and $u \in \mathcal{M}^f(G, v)$, e.g., nodes on all simple or shortest paths from v whose lengths are at most $K > 0$ (Michel et al., 2023; Graziani et al., 2024; Sun et al., 2022) or $k \leq K$-hop neighbors of v where $K = 1$ recovers *basic MPNN* (Nikolentzos et al., 2020; Abu-El-Haija et al., 2019; Feng et al., 2022).

Given architecture f, mapping τ, graph G, and node $v \in V(G)$, let UPDATE and AGGREGATE return different results for $\mathcal{M}^f(G, v)$ and $\mathcal{M}^f(\tau(G), v)$ if $\mathcal{M}^f(G, v) \cap \mathcal{M}^f(\tau(G), v) = \emptyset$ and $|\mathcal{M}^f(\tau(G), v)| \neq |\mathcal{M}^f(\tau(G), v)|$. These are mild conditions and are often satisfied by typical AGGREGATE and UPDATE functions. In this case, we have the following result.

Proposition 4. *Consider the architecture f with $\mathcal{M}^f(G, v)$ in Equation 1. There is a mapping τ such that f is not τ-invariant/equivariant.*

This result extends to MPNN architectures that learn and aggregate representations of subgraphs (*Subgraph MPNN*) (Papp et al., 2021) and those that learn representations for tuples of nodes *Higher-order MPNN* (Morris et al., 2020) (Details in Appendix D). We use GNN and MPNN interchangeably.

4 τ-INVARIANT/EQUIVARIANT GNNS

Given mapping τ, to create a τ-invariance GNN, one approach is to produce $\tau(G)$ and average the output of f on G and $\tau(G)$ (Wang et al., 2025). This method quickly becomes costly, especially for rather large graphs and/or a set of mappings. An alternative is canonicalization: represent all equivalent graphs in a common form (Ma et al., 2024). It demands heavy (manual) pre-processing and high storage overhead for large graphs. It is also unclear whether there are natural forms that preserve accuracy. Next, we explain our approach to achieve invariance.

4.1 STRUCTURAL BIAS

We first present a general approach to defining hypothesis spaces based on the structure of the graph that generalizes how many (node-centric) GNNs define their hypothesis spaces. This enables us to instill invariance for a large set of GNNs. A *rooted graph* $G^v = (V, E)$ is a graph with a single distinguished node $v \in V$. The definitions of isomorphism and homomorphism for rooted graphs G^v and G^u extend their definitions for graphs by always mapping v to u. The **structural bias** of GNN f is a finite and non-empty set of rooted and connected graphs $\mathcal{S}^f$ such that there are no two graphs $S, S' \in \mathcal{S}^f$ where S and S' are isomorphic. Intuitively, f shares information between nodes in the occurrences of graphs in $\mathcal{S}^f$ in G (Maehara & NT, 2025; Nguyen & Maehara, 2020; Jin et al., 2024).

Example 3. *Consider the MPNN in Equation 1 that shares the information of each node with its 1-hop neighbors (Kipf & Welling, 2017; Xu et al., 2019; Wang et al., 2019). The structural bias of the MPNN architecture (Equation 1) over directed graphs where all nodes share the same label l has a single rooted graph $(l(\mathbf{u}), l(w))$ with root $\mathbf{u}$. As another example, GNNs based on meta-paths (Fu et al., 2020; Yang et al., 2023) learn the representation of a node by sharing and aggregating information over all nodes in a set of paths that start from the node. The structural bias for these methods is a set of rooted paths. For example, a structural bias for these methods in Figure 1(a) is $\{(year(\mathbf{u}), paper(w), venue(z)))\}$ with root $\mathbf{u}$.*

The graphs in the structural bias of an algorithm are significantly smaller than G (Maehara & NT, 2025; Nguyen & Maehara, 2020). Given graph G, let G^v be a rooted version of G for a $v \in V(G)$. Each layer of a GNN with structural bias $\mathcal{S}^f$ has three learnable permutation-invariant functions of pattern-aggregation (PAGG), total-aggregation (TAGG), and update such that for every $S^u \in \mathcal{S}^f$:

$$h_v^{(l+1)} = \text{UPDATE}\left(h_v^{(l)}, \text{TAGG}\left(\{\{(\text{PAGG}(hom(S^u, G^v), S^u), S^u)\}\}\right)\right). \tag{2}$$

The PAGG function generalizes the aggregation function in MPNNs (Equation 1): it maps the information of the set of occurrences of graph $S^u \in \mathcal{S}^f$ in G^v for each node $v \in V(G)$ and S^u to a vector in $\mathbb{R}^m$. The TAGG function maps the results of PAGG for all patterns in $\mathcal{S}^f$ (a multi-set of vectors in $\mathbb{R}^m$) to a vector in $\mathbb{R}^m$. The UPDATE function is similar to that in Equation 1. The exact instantiations of $\mathcal{S}^f$, PAGG, TAGG, and UPDATE depend on the specific GNN.

The definition of structural bias captures the inductive bias of using the structure of the graph to share information in many GNNs (Maehara & NT, 2025; Nguyen & Maehara, 2020; Bouritsas et al., 2023; Jin et al., 2024; Barcelo et al., 2021). Consider again the MPNN in Example 3 that shares information of a node with its 1-hop neighbors. For each node $v \in G$, the set of subgraphs $hom(S^u, G^v)$ contains every path of one edge from v to all its neighbors, which enables PAGG to access the information of 1-hop neighbors of v. The PAGG function is the AGGREGATE function in Equation 1. Since

$\mathcal{S}^f$ has only one graph, TAGG maps a set with a single vector $\mathbf{v}$ to $\mathbf{v}$. The UPDATE function is the same as the one in Equation 3. Details on how Equation 2 generalizes other types of GNNs are in Appendix F. We do not use this abstraction to implement GNNs as more efficient implementations for a fixed structural bias are often possible e.g., MPNNs in Example 3.

Because mapping τ modifies the structure of the graph G, the pattern of information propagation of a GNN f may vary between G and $\tau(G)$, therefore, f may not be τ-invariant. Intuitively, we would like f to have a structural bias that for every pattern of information propagation in G there is an equivalent pattern in $\tau(G)$. Formally, let the mapping τ for rooted graphs naturally extend its definition for graphs by mapping the roots of the graphs, i.e., $\tau(G^v) = \tau(G)^{\tau(v)}$. Given mapping τ, bijection $b_\tau : \mathcal{S}^f \to \mathcal{S}^f$ is **isomorphism-preserving** such that for all G and $v \in V(G)$, there is a bijection between $hom(S^u, G^v)$ and $hom(b_\tau(S^u), \tau(G^v))$ that maps every member of $hom(S^u, G^v)$ to an isomorphic member in $hom(b_\tau(S^u), \tau(G^v))$ where the mapping preserves node ids/labels/features.

Lemma 1. *Given mapping τ, a GNN f with pattern aggregation function PAGG is τ-invariant if there is an isomorphism-preserving mapping b_τ such that PAGG$(hom(S^u, G^v), S^u)$ = PAGG$(hom(b_\tau(S^u), \tau(G^v)), b_\tau(S^u))$ for every G^v.*

If we have an isomorphism-preserving mapping on $\mathcal{S}^f$, following Lemma 1, the PAGG function will get essentially the same information. Thus, without significantly restricting possible instantiations of PAGG, we can achieve invariance. However, since τ modifies the topology of the graph, it is not possible to achieve τ-invariance with the current approaches to structural bias in GNNs without using PAGG functions that return the same results for non-equivalent patterns, which is often not desirable.

Proposition 5. *For every GNN f with structural bias $\mathcal{S}^f$, there exists a mapping τ that admits no isomorphism-preserving mapping ($b_\tau : \mathcal{S}^f \to \mathcal{S}^f$).*

4.2 Extending Structural Bias with Conditional Graphs

Following Proposition 5, we should modify the current approach to structural bias to achieve invariance/equivariance. Ideally, we should change the structural bias in such a way that preserves its original one so as not to miss an accurate hypothesis. We should also not expand its structural bias significantly to keep the time overhead limited. To do so, we take into account how a mapping modifies a graph and reflect that information in the structural bias. That is, for each $S^u \in \mathcal{S}^f$, for each $h \in hom(S^u, G^v)$, we find a h' in $\tau(G^v)$ such that h and h' are isomorphic to have exactly the same pattern of information propagation. We then find a structure R^w where $hom(R^w, G^v)$ represents exactly the set of all such h'. If we add R^w to $\mathcal{S}^f$ for every S^u we obtain an isomorphism-preserving mapping. There are two challenges to address in this approach: first, we should identify R^w and then ensure that this algorithm converges without adding too many members to $\mathcal{S}^f$. Since our results apply to both invariance and equivariance properties, for brevity, we explain them to obtain invariance.

Conditional Graphs. To address the first challenge, we generalize the idea of structural bias based on exact pattern homomorphism. More precisely, for each $S^u : (V^S, E^S) \in \mathcal{S}^f$, let E^S have two mutually exclusive and collectively exhaustive subsets: *context edges* (E^S_C) and *propagation edges* (E^S_P). We call the rooted subgraphs of S^u whose sets of edges are E^C_S and E^P_S the *context subgraph* (S^u_C) and the *propagation subgraph* (S^u_P), respectively. S^u_P and S^u_C share the same root, and $V(S^u_P) \subseteq V(S^u_C)$. In this case, we call S^u a *conditional graph* (CG). For the rest of this section, we assume that a structural bias is a set of CGs.

Information Propagation Using CG. Given GNN f, we modify Equation 2 so that for the rooted graph G^v and CG S^u, f checks the existence of the context subgraph S^u_C in G^v and uses the propagation subgraph S^u_P to share information between v and other nodes in G^v. More precisely, given the rooted graph G^v and the CG S^u, let $g_C \in hom(S^u_C, G^v)$ be an occurrence of S^u_C in G^v using the homomorphism mapping $\mathbf{h}$. Let g_P be a graph homomorphic to S^u_P created by $\mathbf{h}$. Since $V(S^u_P) \subseteq S^u_C$), we can always generate such a g_P for every $\mathbf{h}$. We denote the set of generated graphs using all $\mathbf{h}$ from S^u_C to G^v as $pom(S^u, G^v)$. We use $pom(S^u, G^v)$ instead of $hom(S^u, G^v)$ as the first argument of PAGG in Equation 2. In current structural biases, we have $S^u = S^u_C = S^u_P$ for $S^u \in \mathcal{S}^f$, therefore, structural biases with CGs generalize current structural biases.

Example 4. *A CG for a GNN that shares information between 2-hop neighbors in Figure 1(b) is $R^u_C : (year(\mathbf{u}), venue(v))$, $(paper(w), venue(v))$ and $R^u_P : (year(\mathbf{u}), paper(w))$. For every homomorphism of R^u_C in Figure 1(b), the GNN propagates information between the nodes in R^u_P.*

4.3 MINIMALLY MODIFYING STRUCTURAL BIAS

Given the mapping τ, we propose an algorithm that modifies the structural bias of a GNN f to a set of CGs $\mathcal{S}^f_\tau$. For each switching $(P_i, P'_i) \in \tau$, $1 \leq i \leq t$, and each $S^u \in \mathcal{S}^f$, it follows three main steps.

1. Mapping Structural Bias Using Switchings. Assume that there is an isomorphism mapping from a switching P_i to a connected subgraph of $S^u \in \mathcal{S}^f$ that preserves the node labels. In this case, we can capture how (P_i, P'_i) modifies S^u by treating S^u as an input graph and applying (P_i, P'_i) to S^u to generate a new graph R^u_i. The algorithm adds S^u and R^u_i to $\mathcal{S}^f_\tau$, where all edges in S^u and R^u_i are both context and propagation edges. For example, consider a switching that reverses the direction of all edges in an input graph (ignoring labels for simplicity): $P_1 = (x, y)$ and $P'_1 = (y, x)$ and the structural bias for the MPNN that shares the information of each node with its 1-hop neighbors: $S^u = (\mathbf{u}, w)$. Applying the switching in this case, we will add $S^u = (\mathbf{u}, w)$ and $R^u_1 = (w, \mathbf{u})$ to $\mathcal{S}^f_\tau$.

2. Using Switchings as Context Subgraphs. If there is no isomorphism mapping from P_i to any subgraph of S^u, the above step cannot accurately represent how a mapping modifies structural bias. For example, consider the switching in Figure 1 and the structural bias explained in the preceding paragraph. Because there is no isomorphism mapping from the switching to a subgraph of the member in the structural bias, we cannot use the previous step to represent how the input structural bias must be modified based on the switching. Hence, we use the switchings as context subgraphs and represent the input structural biases as propagation subgraphs in these contexts. More precisely, we construct the set of structural biases $\mathbf{R}^f_i = \bigcup_{v \in V(P_i)} \{P^v_i, P'^v_i\}$. The new structural bias must preserve the original method of propagating information represented by S^u. For every $R^w \in \mathbf{R}^f_i$, let b be an isomorphism mapping from S^u to R^w that preserves the node labels. We mark the edge (y, z) in R^w as propagation edges if $(b^{-1}(y), b^{-1}(z))$ exists in S^u. All edges in S^u and members of $\mathbf{R}^f_i$ are marked as context edges. After this step, each $\mathbf{R}^f_i$ contains the original method of propagating information in S^u. We add $\{S^u\} \cup \mathbf{R}^f_i$ to $\mathcal{S}^f_\tau$ and mark all edges in S^u as context and propagation.

3. Mapping Information Propagation Patterns. In the preceding step, $\mathbf{R}^f_i$ has not yet captured how information propagation should be modified according to switching (P_i, P'_i). Thus, we add new rooted graphs to $\mathbf{R}^f_i$ as follows. Let p_i be the node id and label preserving mapping from P_i to P'_i (explained in Section 2). For every $R^u \in \mathbf{R}^f_i$ with the context graph P'^u_i, we add a rooted graph to $\mathcal{S}^f_\tau$ with the context graph P^u_i and propagation edges (y, z) iff there is a propagation edge $(p_i(y), p_i(z))$ in R^u. This step ensures that the modified structural bias contains the propagation patterns represented by P'^u_i for graphs that contain P^u_i. Similarly, for each $R^u \in \mathbf{R}^f_i$ with the context graph P^u_i, we add a rooted graph to $\mathcal{S}^f_\tau$ whose context graph is P'^u_i and has a propagation edge (y, z) iff there is a propagation edge $(p_i^{-1}(y), p_i^{-1}(z))$ in R^u.

The definition of isomorphic CGs extends the definition of isomorphism for rooted graphs that preserves context and propagation types of edges.

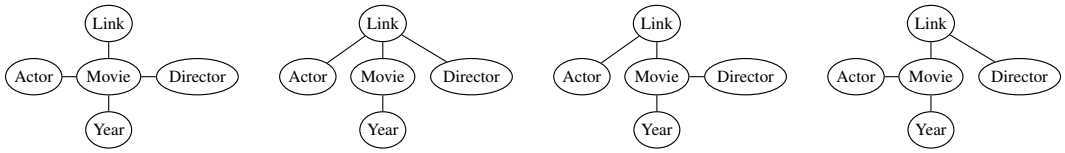
Theorem 2. *$\mathcal{S}^f_\tau$ does not have isomorphic CGs and GNN f with structural bias $\mathcal{S}^f_\tau$ is τ-invariant.*

$\mathcal{S}^f_\tau$ is *minimal*, ensuring the scalability of training. For structural biases $\mathcal{S}$ and $\mathcal{S}'$, we define $\mathcal{S} \preceq \mathcal{S}'$ if for every $S^u \in \mathcal{S}$ there is a $S'^w \in \mathcal{S}'$ such that there is an isomorphism mapping from S^u to a subgraph of S'^w that preserves the node labels and the type of edges (context/propagation). We have $\mathcal{S} \prec \mathcal{S}'$ if at least one isomorphism mapping is from a $S^u \in \mathcal{S}$ to a proper subgraph of a $S'^w \in \mathcal{S}'$.

Proposition 6. *Given GNN f and mapping τ, there is no structural bias $\mathcal{S}'$ such that f is τ-invariant using $\mathcal{S}'$ and $\mathcal{S}' \prec \mathcal{S}^f_\tau$.*

The resulting structural bias is also *complete*: if the original GNN f learns representation of every node in graph$G$, f using $\mathcal{S}^f_\tau$ also learns the representation of every node in G (Details in Appendix G).

The algorithm requires to compute sub-graph isomorphism between graphs used in the structural bias and switchings. These graphs usually have a very few nodes and often in rather simple forms, e.g., paths. Hence, these isomorphisms can be computed efficiently. Ignoring those steps, the time complexity of the algorithm for GNN f and mapping τ is quadratic in the cardinality of $\mathcal{S}^f$ and the number of switchings in τ. These are also often quite small. For example, the cardinality of structural bias (ignoring labels) in MPNN that shares information between 1-hop neighbors is one.



(a) IMDb1 **(b)** IMDb2 **(c)** IMDb3 **(d)** IMDb4

Figure 2: Mappings for IMDb.

5 τ-EXPRESSIVITY

Another important desirable property with respect to an equivalence relation between graphs is for GNN f to learn distinct representations over non-equivalent graphs, i.e., *expressivity* (Xu et al., 2019; Morris et al., 2019). Current works define this property by considering the equivalence under permutations (isomorphism). Since both permutation and switchings preserve the information in graphs, we define expressivity based on both. Formally, consider the isomorphism-closed set of graphs $\mathcal{G} \subseteq \mathcal{G}^n$ and the set of mappings $\mathcal{T}$ on $\mathcal{G}$ that is closed under inverse and composition. f is **more τ-expressive** than f' ($f' \preceq_\tau f$) if $f'(G) \neq f'(H)$, then $f(G) \neq f(H)$ for all graphs $G \neq \rho(H)$ for all $\rho \in \Phi_n^{\mathcal{T}}$. f is **strictly more τ-expressive** than f' ($f' \prec_\tau f$) if f is more τ-expressive than f' and there are G and $H \neq \rho(G)$ for all $\rho \in \Phi_n^{\mathcal{T}}$ such that $f(G) \neq f(H)$ and $f'(G) = f'(H)$.

Maintaining Expressivity. It might reduce the τ-expressivity of a GNN to modify its structural bias. For example, if we modify the MPNN in Equation 2 to ignore the structure of the graph and aggregate the representation of all nodes in the input graph with v in each layer, the resulting GNN is invariant to all mappings τ. Nevertheless, this GNN will have poor τ-expressivity as it learns the same representation for all non-equivalent graphs. Hence, it is important to ensure that the modifications to the structural bias in the algorithm of Section 4.3 of a given GNN do not reduce its expressivity. Given a GNN f with structural bias $\mathcal{S}^f$, we would like to know whether the GNN f_τ (f with structural bias $\mathcal{S}_\tau^f$) is (at least) as expressive as f for every mapping τ.

Theorem 3. *For all mappings τ and GNN f, we have $f \preceq_\tau f_\tau$. Also, there is a mapping τ and GNN f such that $f \prec_\tau f_\tau$.*

Another question is if τ-invariance maintains the relative expressivity of f. Let $f \preceq f'$ denote that f' is more expressive than f only with respect to isomorphism.

Proposition 7. *Given mapping τ and GNNs f and f' such that $f \preceq f'$, we have $f_\tau \preceq_\tau f'_\tau$.*

6 EMPIRICAL STUDY

Datasets & Variants. We use DBLP (bibliography) and IMDb (movies) from (Fu et al., 2020), WN18RR (standard link prediction dataset derived from WordNet) (Dettmers et al., 2018), and Freebase (knowledge graph) (Bollacker et al., 2008). We use the properties of the datasets to infer their mappings (Skavantzos & Link, 2023; Egger et al., 2026). Figure 2 shows the mappings for IMDb; details for the other datasets are in Appendix H.1.

Models & Methods. We evaluate seven GNN architectures spanning diverse structural biases: relation-typed message passing (RGCN (Schlichtkrull et al., 2018)), source-conditioned path propagation (CMPNN (Huang et al., 2023b)), node-type-aware attention (SlotGAT (Zhou et al., 2023)), metapath message passing (MAGNN (Fu et al., 2020)), pre-computed metapath channels (SeHGNN (Yang et al., 2023)), rooted paths (PAIN (Graziani et al., 2024)), and sampled walks (PathNet (Sun et al., 2022)). This follows recent evidence (Yang et al., 2025) that performance gains are better explained by structural information than by architectural complexity alone. Following Section 4.3, we construct invariant structural biases for these algorithms. For each model we compare four arms differing only in their structural input (Appendix H.2): the published GNN, unmodified (Orig); data augmentation (Aug), trained jointly on all variants (Gerken et al., 2022; Chen et al., 2020; Zhao et al., 2021b); the universal graph (Univ), trained on their union, i.e. canonicalization; and our invariant version (Inv), the only arm trained on a single variant.

Metrics. We report degree of invariance, effectiveness, and efficiency. Effectiveness is Micro-F1 for node classification and MRR for link prediction, with Δ its max–min spread across variants; degree of invariance is Kendall-τ between the output-score rankings of every pair of variants, averaged over pairs, runs, and test instances (1 means identical rankings); efficiency is training time. Every number is a mean over variants and three runs. Other metrics and full definitions are in Appendix H.4.

Task	Model	Micro-F1 ↑				ΔMicro-F1 ↓				Kendall-τ ↑				Time (min) ↓			
		Orig	Aug	Univ	Inv	Orig	Aug	Univ	Inv	Orig	Aug	Univ	Inv	Orig	Aug	Univ	Inv
IMDb-NC	RGCN	$63.05_{\pm 1.12}$	$60.47_{\pm 1.61}$	$\mathbf{65.35}_{\pm \mathbf{0.89}}$	$65.31_{\pm 1.07}$	4.82	1.55	0.00	**0.00**	0.784	0.945	1.000	**1.000**	0.74	**0.23**	0.70	0.70
IMDb-NC	SlotGAT	$65.52_{\pm 0.70}$	$64.71_{\pm 1.61}$	$65.19_{\pm 1.02}$	$\mathbf{66.11}_{\pm \mathbf{1.45}}$	0.68	0.96	0.00	**0.00**	0.798	0.931	1.000	**1.000**	0.66	**0.62**	0.90	0.89
IMDb-NC	MAGNN	$62.20_{\pm 0.82}$	$61.92_{\pm 1.21}$	$\mathbf{63.20}_{\pm \mathbf{0.60}}$	$63.08_{\pm 1.17}$	1.04	1.48	0.00	**0.00**	0.720	0.941	1.000	**1.000**	**2.68**	4.04	4.04	4.34
IMDb-NC	SeHGNN	$63.84_{\pm 1.43}$	$62.76_{\pm 1.29}$	$64.31_{\pm 0.39}$	$\mathbf{65.35}_{\pm \mathbf{0.64}}$	1.19	0.52	0.00	**0.00**	0.646	0.937	1.000	**1.000**	1.41	3.36	3.50	**1.07**
IMDb-NC	PAIN	$\mathbf{57.38}_{\pm \mathbf{1.08}}$	$43.71_{\pm 1.56}$	$56.70_{\pm 0.54}$	$56.70_{\pm 0.54}$	3.63	10.68	0.00	**0.00**	0.498	0.405	1.000	**1.000**	**52.22**	120.14	199.88	199.57
IMDb-NC	PathNet	$59.36_{\pm 1.47}$	$54.11_{\pm 6.13}$	$53.81_{\pm 6.12}$	$\mathbf{59.85}_{\pm \mathbf{1.14}}$	0.52	0.16	0.00	**0.00**	0.765	0.944	1.000	**1.000**	0.31	**0.20**	0.53	0.43
Freebase-NC	RGCN	$68.55_{\pm 0.62}$	$\mathbf{69.06}_{\pm \mathbf{0.43}}$	$68.64_{\pm 0.31}$	$68.95_{\pm 0.47}$	0.61	0.19	0.00	**0.00**	0.723	0.982	1.000	**1.000**	**11.65**	12.45	20.42	25.35
Freebase-NC	SlotGAT	$\mathbf{65.50}_{\pm \mathbf{0.95}}$	$63.27_{\pm 0.61}$	$63.96_{\pm 0.61}$	$63.96_{\pm 0.61}$	1.24	5.13	0.00	**0.00**	0.551	0.768	1.000	**1.000**	**266.41**	325.17	606.09	606.82
Freebase-NC	MAGNN	$60.77_{\pm 0.49}$	$61.03_{\pm 0.26}$	$60.21_{\pm 0.62}$	$\mathbf{61.07}_{\pm \mathbf{0.93}}$	0.99	0.08	0.00	**0.00**	0.706	0.995	1.000	**1.000**	**20.69**	43.62	28.95	86.25
Freebase-NC	SeHGNN	$\mathbf{67.04}_{\pm \mathbf{0.55}}$	$65.38_{\pm 1.46}$	$65.87_{\pm 0.41}$	$66.37_{\pm 0.80}$	0.88	0.04	0.00	**0.00**	0.524	0.768	1.000	**1.000**	**3.23**	4.57	6.17	5.21
Freebase-NC	PAIN	$59.22_{\pm 1.18}$	$\mathbf{59.30}_{\pm \mathbf{1.16}}$	$59.29_{\pm 1.31}$	$59.29_{\pm 1.31}$	7.22	4.96	0.00	**0.00**	0.635	0.624	1.000	**1.000**	**743.31**	995.05	813.00	809.08
Freebase-NC	PathNet	$\mathbf{52.02}_{\pm \mathbf{1.68}}$	$48.12_{\pm 1.03}$	$47.68_{\pm 2.69}$	$50.01_{\pm 1.15}$	8.25	0.59	0.00	**0.00**	0.803	0.963	1.000	**1.000**	0.18	0.32	**0.16**	0.28
Task	Model	MRR ↑				ΔMRR ↓				Kendall-τ ↑				Time (min) ↓			
		Orig	Aug	Univ	Inv	Orig	Aug	Univ	Inv	Orig	Aug	Univ	Inv	Orig	Aug	Univ	Inv
IMDb-LP	RGCN	$19.54_{\pm 5.38}$	$\mathbf{21.36}_{\pm \mathbf{2.64}}$	$17.66_{\pm 2.72}$	$20.01_{\pm 0.47}$	2.18	1.17	0.00	**0.00**	0.536	0.747	1.000	**1.000**	3.69	**1.90**	3.49	3.72
IMDb-LP	SlotGAT	$25.34_{\pm 1.07}$	$25.35_{\pm 1.78}$	$\mathbf{26.10}_{\pm \mathbf{0.66}}$	$25.14_{\pm 0.88}$	2.06	0.17	0.00	**0.00**	0.436	0.886	1.000	**1.000**	**0.62**	0.93	1.08	0.91
IMDb-LP	MAGNN	$28.49_{\pm 0.46}$	$28.48_{\pm 0.43}$	$\mathbf{29.22}_{\pm \mathbf{0.55}}$	$28.45_{\pm 0.45}$	0.06	0.03	0.00	**0.00**	0.765	0.997	1.000	**1.000**	14.91	**13.62**	22.53	16.54
IMDb-LP	CMPNN	$37.91_{\pm 0.90}$	$38.09_{\pm 0.67}$	$41.71_{\pm 0.61}$	$\mathbf{42.28}_{\pm \mathbf{0.66}}$	7.22	6.62	0.00	**0.00**	0.782	0.800	1.000	**1.000**	12.19	**11.65**	13.54	12.72
DBLP-LP	RGCN	$34.37_{\pm 1.70}$	$33.19_{\pm 2.58}$	$\mathbf{40.24}_{\pm \mathbf{6.06}}$	$\mathbf{40.24}_{\pm \mathbf{6.06}}$	5.13	4.71	0.00	**0.00**	0.469	0.870	1.000	**1.000**	**0.79**	4.21	0.92	0.93
DBLP-LP	SlotGAT	$55.12_{\pm 0.75}$	$53.43_{\pm 1.51}$	$\mathbf{55.65}_{\pm \mathbf{1.01}}$	$\mathbf{55.65}_{\pm \mathbf{1.01}}$	1.43	0.37	0.00	**0.00**	0.664	0.963	1.000	**1.000**	2.79	6.67	**2.49**	2.62
DBLP-LP	MAGNN	$35.19_{\pm 3.49}$	$28.68_{\pm 2.65}$	$\mathbf{40.14}_{\pm \mathbf{6.31}}$	$\mathbf{40.14}_{\pm \mathbf{6.31}}$	10.99	0.01	0.00	**0.00**	0.486	0.999	1.000	**1.000**	**90.57**	95.82	110.65	111.48
DBLP-LP	CMPNN	$61.59_{\pm 0.39}$	$61.68_{\pm 0.56}$	$\mathbf{61.94}_{\pm \mathbf{0.12}}$	$\mathbf{61.94}_{\pm \mathbf{0.12}}$	0.63	0.17	0.00	**0.00**	0.742	0.839	1.000	**1.000**	**258.32**	292.00	285.81	286.25
WN18RR-LP	RGCN	$\mathbf{38.20}_{\pm \mathbf{0.41}}$	$35.40_{\pm 1.25}$	$37.61_{\pm 0.25}$	$38.02_{\pm 0.19}$	3.14	4.74	0.00	**0.00**	0.239	0.635	1.000	**1.000**	39.79	322.77	**38.35**	42.11
WN18RR-LP	SlotGAT	$36.57_{\pm 0.08}$	$36.32_{\pm 0.16}$	$36.35_{\pm 0.01}$	$\mathbf{36.62}_{\pm \mathbf{0.17}}$	0.08	0.14	0.00	**0.00**	0.265	0.820	1.000	**1.000**	21.84	**20.04**	36.52	33.35
WN18RR-LP	CMPNN	$34.13_{\pm 3.20}$	$31.70_{\pm 3.43}$	$35.24_{\pm 0.36}$	$\mathbf{42.38}_{\pm \mathbf{0.95}}$	13.11	8.77	0.00	**0.00**	0.029	0.336	1.000	**1.000**	403.39	344.49	990.71	**162.72**

Table 1: The published GNN (Orig), data augmentation (Aug), the universal graph (Univ), and our invariant version (Inv) of the same model, averaged over dataset variants, $\pm$ the mean seed-to-seed standard deviation. Δ is the max–min spread across variants and τ the mean pairwise Kendall-τ; the IMDb LP task is Movie–Director. **Bold** is the best of the four, ties included. Univ is excluded for Δ and τ: it trains one model on one graph and is evaluated once, so both are trivial rather than measured, whereas Inv attains them across models trained separately per variant. Other metrics and per-variant scores are in Appendix H.5.

Results & Analysis. Table 1 reports invariance, effectiveness, and efficiency across seven structural biases: relation-typed message passing, node-type-aware attention, source-conditioned paths, metapaths, pre-computed channels, rooted paths, and sampled walks. Our method consistently removes sensitivity to the non-isomorphic dataset variants under the reported aggregate. The original GNNs are far from invariant: their pairwise Kendall-τ ranges from 0.50 to 0.80 on IMDb-NC and 0.47 to 0.74 on DBLP-LP, and falls to 0.03–0.27 on WN18RR-LP, where the mined mappings change the outputs most. The effectiveness spread agrees: the original CMPNN moves by 13.1 MRR points across the WN18RR variants. Aug raises the agreement but never removes it, reaching at most $\tau = 0.999$, and for PAIN it falls below the original, whose augmentation arm is the least stable we trained, running for 461 ± 270 super-epochs across seeds without settling. For Inv, $\tau = 1$ follows from Theorem 2 rather than being discovered: the compiled bias is identical on every variant, so the per-variant models have identical parameters. The measured $\tau = 1.000 \pm 0.000$ with zero spread, on every pair of variants of every dataset, checks that the implementation realizes the theory; what it leaves open, and the rest of this section tests, is whether such a bias can be compiled for architectures this different and what it costs. Univ is invariant too, but trivially, since it trains one model on one graph and needs every variant.

Invariance does not cost effectiveness and usually improves it: Inv improves on the published model in 14 of the 23 model–task cells and matches it within 0.2 points in 5 more, gaining 8.3 MRR points for CMPNN on WN18RR-LP, 5.9 for RGCN and 5.0 for MAGNN on DBLP-LP, and 4.4 for CMPNN on IMDb-LP; the smaller differences elsewhere are within the seed-to-seed variation reported in the table. Inv matches Univ wherever the two compute the same function and exceeds it elsewhere, by 7.1 points for CMPNN on WN18RR-LP and 6.0 for PathNet on IMDb-NC. The four shortfalls trace to the architecture rather than the mapping. Three are on Freebase-NC, where SlotGAT's single softmax, SeHGNN's fixed channel set, and PathNet's fixed walk budget are each diluted by the dense shortcut compositions that the invariant bias must include, and where Univ, which carries the same information, does no better; the fourth is PAIN on IMDb-NC, at 0.7 points.

The cost of invariance is the size of the compiled bias, so where that bias is small the cost is negligible. The largest single effect is in fact a saving: CMPNN on WN18RR-LP trains $2.5\times$ faster, four hours less, propagating over the base relations alone, and RGCN and SeHGNN on IMDb-NC and SlotGAT on DBLP-LP are also faster than the published model. Most remaining cells cost 1.1 to $1.6\times$, seconds at their scale. The real cost is where the mined mappings materialize hundreds of millions of shortcut edges: $2.2\times$ for RGCN and SlotGAT and $4.2\times$ for MAGNN with the Union metapaths on Freebase-NC, and $3.8\times$ for PAIN on IMDb-NC, whose exact path enumeration dominates. Inv is at or below Univ's cost in most cells, and unlike Univ it never needs the other variants at all. Per-variant scores and the full analysis are in Appendix H.5.

AI USE STATEMENT

We have used generative AI tools to improve the presentation of the submission and to help with writing and debugging code.

ETHICS STATEMENT

All datasets used in our study are publicly available. The authors declare that they do not have any conflicts of interest, which may affect the reported results. We do not expect any harmful result from the methods and algorithms proposed by this work. Our proposed methods require full access to the underlying dataset and cannot be used to recover information on private data.

REPRODUCIBILITY STATEMENT

We have reported the details of the datasets in Appendix H.1, details of the models and their hyper-parameters in Appendix H.2, details on the implementation of our approach for each mode in Appendix H.3, and the proofs to all formal results in Appendix E. The codes for all methods and models are available at `https://anonymous.4open.science/r/INV-GNN-12D9/README.md`.

REFERENCES

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 21–29. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/abu-el-haija19a.html.

Marcelo Arenas. Normalization theory for xml. *SIGMOD Rec.*, 35(4):57–64, December 2006. ISSN 0163-5808. doi: 10.1145/1228268.1228284. URL http://doi.acm.org/10.1145/1228268.1228284.

Pablo Barcelo, Jorge Perez, and Juan Reutter. Schema mappings and data exchange for graph databases. In *ICDT*, 2013.

Pablo Barcelo, Floris Geerts, Juan L Reutter, and Maksimilian Ryschkov. Graph neural networks with local graph parameters. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=yGKklt8wyV.

Maciej Besta, Florian Scheidl, Lukas Gianinazzi, Grzegorz Kwasniewski, Shachar Klaiman, Jurgen Muller, and Torsten Hoefler. Demystifying Higher-Order Graph Neural Networks . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 48(03):2544–2565, March 2026. ISSN 1939-3539. doi: 10.1109/TPAMI.2025.3637114. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2025.3637114.

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, G. Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. *ArXiv*, abs/2110.02910, 2021. URL https://api.semanticscholar.org/CorpusID:238407774.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250, 2008. doi: 10.1145/1376616.1376746.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, January 2023. ISSN 1939-3539. doi: 10.1109/tpami.2022.3154319. URL http://dx.doi.org/10.1109/TPAMI.2022.3154319.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Comput. Surv.*, 56(6), March 2024. ISSN 0360-0300. doi: 10.1145/3643806. URL https://doi.org/10.1145/3643806.

Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020. URL https://jmlr.org/papers/v21/20-163.html.

E. F. Codd. Normalized data base structure: a brief tutorial. In *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, SIGFIDET '71, pp. 1–17, New York, NY, USA, 1971. Association for Computing Machinery. ISBN 9781450373005. doi: 10.1145/1734714.1734716. URL https://doi.org/10.1145/1734714.1734716.

Yuanfei Dai, Shiping Wang, Neal N Xiong, and Wenzhong Guo. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics*, 9(5):750, 2020. URL https://api.semanticscholar.org/CorpusID:219053633.

Pim de Haan, Taco Cohen, and Max Welling. Natural graph networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984.

Maximilian K. Egger, Mehdi Allali, Matteo Lissandrini, Davide Mottin, and Panagiotis Karras. Structural normalization of property graphs. *Proc. VLDB Endow.*, 19(6):1400–1412, May 2026. ISSN 2150-8097. doi: 10.14778/3797919.3797943. URL https://doi.org/10.14778/3797919.3797943.

Ronald Fagin. Inverting schema mappings. *ACM Trans. Database Syst.*, 32(2), 2007.

Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005. doi: 10.1016/j.tcs.2004.10.033.

Wenfei Fan and Philip Bohannon. Information preserving xml schema embedding. *TODS*, 33(1), 2008.

Jiarui Feng, Yixin Chen, Fuhai Li, Anindya Sarkar, and Muhan Zhang. How powerful are k-hop message passing graph neural networks? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23537–23550. Curran Associates, Inc., 2022. URL https://papers.neurips.cc/paper_files/paper/2022/file/1ece70d2259b8e9510e2d4ca8754cecf-Paper-Conference.pdf.

Fabrizio Frasca, Beatrice Bevilacqua, Michael M. Bronstein, and Haggai Maron. Understanding and extending subgraph gnns by rethinking their symmetries. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, pp. 2331–2341, 2020.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1eWbxStPH.

Floris Geerts and Juan L Reutter. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=wIzUeM3TAU.

Jan E. Gerken, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Equivariance versus augmentation for spherical images. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7404–7421. PMLR, 2022. URL https://proceedings.mlr.press/v162/gerken22a.html.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1263–1272, 2017. URL https://proceedings.mlr.press/v70/gilmer17a/gilmer17a.pdf.

C. D. Godsil and B. D. McKay. Constructing cospectral graphs. *Aequationes Mathematicae*, 25: 257–268, 1982.

Caterina Graziani, Tamara Drucks, Fabian Jogl, Monica Bianchini, Franco Scarselli, and Thomas Gärtner. The expressive power of path-based graph neural networks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16226–16249. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/graziani24a.html.

Claudio Gutierrez, Carlos Hurtado, and Alberto O. Mendelzon. Foundations of semantic web databases. PODS '04, pp. 95–106, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 158113858X. doi: 10.1145/1055558.1055573. URL https://doi.org/10.1145/1055558.1055573.

Willem H Haemers. Interlacing eigenvalues and graphs. *Linear Algebra and its applications*, 226: 593–616, 1995.

Joseph Y. Halpern and Daphne Koller. Representation dependence in probabilistic inference. *J. Artif. Int. Res.*, 21(1):319–356, March 2004. ISSN 1076-9757.

William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

Aidan Hogan, Marcelo Arenas, Alejandro Mallea, and Axel Polleres. Everything you always wanted to know about blank nodes. *Journal of Web Semantics*, 27–28:42–69, 2014. doi: 10.1016/j.websem.2014.06.004.

Ningyuan (Teresa) Huang, Ron Levie, and Soledad Villar. Approximately equivariant graph networks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, Red Hook, NY, USA, 2023a. Curran Associates Inc.

Xingyue Huang, Miguel Romero, Ismail Ilkan Ceylan, and Pablo Barceló. A theory of link prediction via relational weisfeiler-leman on knowledge graphs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023b. Curran Associates Inc.

Xingyue Huang, Miguel Romero Orth, Pablo Barceló, Michael M. Bronstein, and İsmail İlkan Ceylan. Link prediction with relational hypergraphs, 2024. URL https://arxiv.org/abs/2402.04062.

Truong Son Hy, Shubhendu Trivedi, Horace Pan, Brandon M Anderson, and Risi Kondor. Predicting molecular properties with covariant compositional networks. *The Journal of Chemical Physics*, 148(24):241745, 2018. doi: 10.1063/1.5024797. URL https://doi.org/10.1063/1.5024797.

Yeonjun In, Kanghoon Yoon, Sukwon Yun, Kibum Kim, Sungchul Kim, and Chanyoung Park. Training robust graph neural networks by modeling noise dependencies. In *Advances in Neural Information Processing Systems*, 2025.

Manfred Jaeger. Representation independence of nonmonotonic inference relations. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pp. 461–474, 1996.

Emily Jin, Michael Bronstein, Ismail Ilkan Ceylan, and Matthias Lanzinger. Homomorphism counts for graph neural networks: all about that basis. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Liò. On the expressive power of geometric graph neural networks. In *International Conference on Machine Learning (ICML)*, pp. 15809–15830. PMLR, 2023.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, Xinwang Liu, Philip S. Yu, and Ming Zhang. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges, 2025. URL https://arxiv.org/abs/2403.04468.

Nicolas Keriven and Gabriel Peyré. *Universal invariant and equivariant graph neural networks*. NeurIPS '19. Curran Associates Inc., Red Hook, NY, USA, 2019.

Jinwoo Kim, Saeyoon Oh, Sungjun Cho, and Seunghoon Hong. Equivariant hypergraph neural networks. In *Computer Vision – ECCV 2022*, volume 13681 of *Lecture Notes in Computer Science*, pp. 86–103. Springer, 2022. doi: 10.1007/978-3-031-19803-8_6.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *arXiv preprint arXiv:1611.07308*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Matthias Lanzinger and Pablo Barcelo. On the power of the weisfeiler-leman test for graph motif parameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FddFxi08J3.

Ming Li, Yongchun Gu, Yi Wang, Yujie Fang, Lu Bai, Xiaosheng Zhuang, and Pietro Lio. When hypergraph meets heterophily: New benchmark datasets and baseline. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 18377–18384, 2025.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1150–1160, 2021. doi: 10.1145/3447548.3467350.

Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks, 2020. URL https://arxiv.org/abs/2005.00178.

George Ma, Yifei Wang, Derek Lim, Stefanie Jegelka, and Yisen Wang. A canonicalization perspective on invariant and equivariant learning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Takanori Maehara and Hoang NT. Deep homomorphism networks. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=Syx72jC9tm.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International conference on machine learning*, pp. 4363–4371. PMLR, 2019b.

Jonathan Masci, Emanuele Rodolà, Davide Boscaini, Michael M Bronstein, and Hao Li. Geometric deep learning. In *SIGGRAPH ASIA 2016 Courses*, pp. 1–50. 2016.

Gaspard Michel, Giannis Nikolentzos, Johannes Lutzeyer, and Michalis Vazirgiannis. Path neural networks: Expressive and accurate graph neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24737–24755. PMLR, 2023. URL https://proceedings.mlr.press/v202/michel23a.html.

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

Iain Moffatt and Jaeseong Oh. A 2-isomorphism theorem for delta-matroids. *Advances in Applied Mathematics*, 126:102133, 2021. ISSN 0196-8858. doi: https://doi.org/10.1016/j.aam.2020.102133. URL https://www.sciencedirect.com/science/article/pii/S0196885820301366. Special Issue: Dedicated to Joseph Kung.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 21824–21840, 2020.

Artem Moskalev, Anna Sepliarskaia, Erik J. Bekkers, and Arnold W. M. Smeulders. On genuine invariance learning without weight-tying. In *Proceedings of the 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pp. 218–227. PMLR, 2023. URL https://proceedings.mlr.press/v221/moskalev23a.html.

Hoang Nguyen and Takanori Maehara. Graph homomorphism convolution. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7306–7316. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/nguyen20c.html.

Giannis Nikolentzos, George Dasoulas, and Michalis Vazirgiannis. k-hop graph neural networks. *Neural Networks*, 130:319–330, 2020.

Oskar Nordenfors, Fredrik Ohlsson, and Axel Flinth. Optimization dynamics of equivariant and augmented neural networks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=PTTa3U29NR.

Theodore Papamarkou, Tolga Birdal, Michael M. Bronstein, Gunnar E. Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Lio, Paolo Di Lorenzo, Vasileios Maroulas, Nina Miolane, Farzana Nasrin, Karthikeyan Natesan Ramamurthy, Bastian Rieck, Simone Scardapane, Michael T Schaub, Petar Veličković, Bei Wang, Yusu Wang, Guowei Wei, and Ghada Zamzmi. Position: Topological deep learning is the new frontier for relational learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39529–39555. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/papamarkou24a.html.

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey of message-passing topological neural networks, 2024. URL https://arxiv.org/abs/2304.10031.

Pal Andras Papp and Roger Wattenhofer. A theoretical comparison of graph neural network extensions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17323–17345. PMLR, 2022. URL https://proceedings.mlr.press/v162/papp22a.html.

Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. Dropgnn: Random dropouts increase the expressiveness of graph neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21997–22009. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b8b2926bd27d4307569ad119b6025f94-Paper.pdf.

Jose Picado, Arash Termehchy, Alan Fern, and Parisa Ataei. Schema independent relational learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pp. 929–944, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450341974. doi: 10.1145/3035918.3035923. URL https://doi.org/10.1145/3035918.3035923.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pp. 593–607. Springer, 2018.

Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems 30*, pp. 991–1001. Curran Associates, Inc., 2017.

J. J. Seidel. A survey of two-graphs. *Colloquio Internazionale sulle Teorie Combinatorie*, pp. 481–511, 1976.

Márcio Rosa Da Silva, Jibin Sun, Hongwu Ma, Feng He, and An Ping Zeng. *Metabolic Networks*, pp. 233–253. John Wiley & Sons, Inc., aug 2007. ISBN 9780470041444. doi: 10.1002/9780470253489.ch10.

Philipp Skavantzos and Sebastian Link. Normalizing property graphs. *Proc. VLDB Endow.*, 16 (11):3031–3043, July 2023. ISSN 2150-8097. doi: 10.14778/3611479.3611506. URL https://doi.org/10.14778/3611479.3611506.

Philipp Skavantzos and Sebastian Link. Third and boyce–codd normal form for property graphs. *The VLDB Journal*, 34(2):23, 2025. doi: 10.1007/s00778-025-00902-2. URL https://doi.org/10.1007/s00778-025-00902-2.

Derek A. Smith, Lorenzo Traldi, and William Watkins. A note on 2-isomorphisms and the signed laplacian matrix of a graph. *Linear Algebra and its Applications*, 563:277–286, 2019. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2018.11.002. URL https://www.sciencedirect.com/science/article/pii/S0024379518305251.

Yifei Sun, Haoran Deng, Yang Yang, Chunping Wang, Jiarong Xu, Renhong Huang, Linfeng Cao, Yang Wang, and Lei Chen. Beyond homophily: Structure-aware path aggregation graph neural network. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2233–2240. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/310. URL https://doi.org/10.24963/ijcai.2022/310. Main Track.

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. URL https://openreview.net/forum?id=HkgEQnRqYQ.

Erik Henning Thiede, Wenda Zhou, and Risi Kondor. Autobahn: Automorphism-based graph neural nets. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=NU69dglcsS.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Hu Tian, Bowei Ye, Xiaolong Zheng, and Desheng Dash Wu. Robust graph neural networks against adversarial attacks via jointly adversarial training. *IFAC-PapersOnLine*, 53(5):420–425, 2020. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2021.04.225. URL https://www.sciencedirect.com/science/article/pii/S2405896321004109. 3rd IFAC Workshop on Cyber-Physical & Human Systems CPHS 2020.

K Trivodaliev, A Bogojeska, and L Kocarev. Exploring function prediction in protein interaction networks via clustering methods. *PLoS ONE*, 9(6), 2014. URL https://doi.org/10.1371/journal.pone.0099755.

K. Truemper. On the delta-wye reduction for planar graphs. *Journal of Graph Theory*, 13(2):141–148, 1989. doi: https://doi.org/10.1002/jgt.3190130202. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190130202.

J.H. VAN LINT and J.J. SEIDEL. Equilateral point sets in elliptic geometry. pp. 3–16, 1991. doi: https://doi.org/10.1016/B978-0-12-189420-7.50008-6. URL https://www.sciencedirect.com/science/article/pii/B9780121894207500086.

Ameya Velingker, Ali Sinop, Ira Ktena, Petar Veličković, and Sreenivas Gollapudi. Affinity-aware graph networks. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d642b0633afad94f660554e05b40608e-Abstract-Conference.html.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *6th International Conference on Learning Representations*, 2018.

Guangtao Wang, Rex Ying, Jing Huang, and Jure Leskovec. Multi-hop attention graph neural networks. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 3089–3096. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/425. URL https://doi.org/10.24963/ijcai.2021/425. Main Track.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pp. 2022–2032, 2019.

Zaitian Wang, Pengfei Wang, Kunpeng Liu, Pengyang Wang, Yanjie Fu, Chang-Tien Lu, Charu C. Aggarwal, Jian Pei, and Yuanchun Zhou. A comprehensive survey on data augmentation, 2025. URL https://arxiv.org/abs/2405.09591.

H. Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55:245–254, 1933a.

H. Whitney. On the classification of graphs. *American Journal of Mathematics*, 55(1):236–244, 1933b.

E. C. Wood, Amy K. Glen, Lindsey G. Kvarfordt, Finn Womack, Liliana Acevedo, Timothy S. Yoon, Chunyu Ma, Veronica Flores, Meghamala Sinha, Yodsawalai Chodpathumwan, Arash Termehchy, Jared C. Roach, Luis Mendoza, Andrew S. Hoffman, Eric W. Deutsch, David Koslicki, and Stephen A. Ramsey. Rtx-kg2: a system for building a semantically standardized knowledge graph for translational biomedicine. *bioRxiv*, 2021. doi: 10.1101/2021.10.17.464747. URL https://www.biorxiv.org/content/early/2021/10/25/2021.10.17.464747.

Donglin Xia, Xiao Wang, Nian Liu, and Chuan Shi. Learning invariant representations of graph neural networks via cluster generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=zrCmeqV3Sz.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4854–4873, 2020.

Xiao Yang, Xuejiao Zhao, and Zhiqi Shen. Are heterogeneous graph neural networks truly effective? a causal perspective. *arXiv preprint arXiv:2510.05750*, 2025.

Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10816–10824, 2023.

Cong Yu and H. V. Jagadish. Xml schema refinement through redundancy detection and normalization. *The VLDB Journal*, 17(2):203–223, March 2008. ISSN 1066-8888. doi: 10.1007/s00778-007-0063-0. URL https://doi.org/10.1007/s00778-007-0063-0.

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of gnns via graph biconnectivity. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4685–4693, 2021a.

Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11015–11023, 2021b.

Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. Graph data augmentation for graph machine learning: A survey. *IEEE Data Engineering Bulletin*, 46(2):140–165, 2023. URL http://sites.computer.org/debull/A23june/p140.pdf.

Ziang Zhou, Jieming Shi, Renchi Yang, Yuanhang Zou, and Qing Li. Slotgat: slot-based message passing for heterogeneous graphs. In *International Conference on Machine Learning*, pp. 42644–42657. PMLR, 2023.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: a general graph neural network framework for link prediction. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NeurIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

A LIMITATIONS

Our approach has two main limitations.

First, there are bijective mappings that are expressed using more complex languages than switchings, e.g., graph languages based on first order logic with/out recursion (Fan & Bohannon, 2008; Gutierrez et al., 2004). Since they are bijective and constitute a group, our framework extends to these mappings by redefining conditional graphs using a more complex language. In this paper, we focus on the ones expressed using switchings because we believe that there are frequent graph variations that can be expressed in the form of switching, e.g., data normalization or examples in (Arenas, 2006; Silva et al., 2007; Trivodaliev et al., 2014; Truemper, 1989; Wood et al., 2021). Moreover, it is generally hard to define/check the bijectivity conditions for complex mappings on relational structures (Fagin, 2007). More importantly, as the resulting conditional graphs become too expressive, the training may be inefficient. Hence, it requires more than the space and scope of a single paper to address the challenges of these mappings. Our work introduces a new direction for informational equivalence and opens the door for a much broader scope of future work.

Second, like current effort on equivariant learning over graphs, our formal framework targets exact and invertible mappings without any noise. The modified structural bias $\mathcal{S}^f_\tau$ is constructed so that the resulting GNN is invariant to the specified mapping τ, while remaining minimal with respect to the proposed ordering over structural biases. However, in some applications, exact invariance may be stronger than necessary. Certain transformations may preserve most, but not all, task-relevant information; enforcing full invariance to them could therefore reduce expressivity or obscure meaningful structural differences. A promising direction is to leverage this empirical observation and formalize the notion of approximately invariant mappings, where the model is encouraged to behave similarly across transformed variants without requiring identical behavior. Our method can use incomplete and partial mappings to encode the given information in the structural bias. We expect that the resulting method outperforms the baseline GNN method over the same dataset. If the mappings are approximate, the induced structural bias by our method will propagate information across nodes as if there are noises in the input graph. The robustness of the resulting methods may also depend on the robustness of the underlying GNN to edge noises. Since our methods can be used in any GNN, we can use our method with GNNs that are robust to noise (In et al., 2025). We expect the resulting model to outperform its underlying GNN as it uses additional information about some variations of the data in learning. Such an extension could provide a tunable trade-off between invariance and expressivity, allowing the model to retain useful sensitivity to transformations that are only partially semantics-preserving.

B BROADER IMPACTS

This work contributes to more reliable and reproducible graph learning by reducing dependence on incidental choices in graph representation. As graph data are increasingly used in scientific, engineering, social, and information systems, the same underlying information may be encoded using different schemas, node types, edge types, or relational structures. If model behavior changes substantially across such encodings, downstream conclusions may reflect representation artifacts rather than meaningful properties of the data. By inducing invariance with respect to specified mappings, our approach can make GNN-based predictions more stable across equivalent graph representations and support more consistent comparisons between models, datasets, and preprocessing pipelines.

The broader societal benefit is most relevant in domains where graph schemas are designed by humans or integrated from heterogeneous sources. In biomedical and scientific knowledge graphs, invariant mappings may reduce sensitivity to database-specific modeling conventions and improve the portability of learned models across resources. In infrastructure, transportation, and engineering networks, they may support more consistent analysis when systems are represented at different levels of abstraction. In recommender systems, fraud detection, and information retrieval, they may reduce unintended variation caused by schema design choices, making deployed systems easier to audit and reproduce.

Potential negative impacts may arise when invariant mappings are mis-specified or applied without sufficient domain validation. If a mapping is too broad, the resulting model may suppress task-relevant

distinctions; if invariance metrics are treated as evidence of general fairness or correctness, they may obscure other sources of error or bias. This is particularly important in high-stakes graph applications involving biomedical, social, economic, or institutional data. Invariant mappings should therefore be documented, domain-justified, and evaluated together with task performance and subgroup-level analyses.

More generally, this work encourages a shift from evaluating GNNs only on fixed benchmark representations toward evaluating whether their predictions are stable under information-preserving changes in representation. Such evaluation can improve transparency by making explicit which mappings are assumed to preserve task-relevant information. The intended societal impact is therefore methodological: to support graph learning systems whose behavior is less tied to arbitrary encoding decisions and more aligned with the underlying information that the graph is meant to represent.

C Related Work

GNN Invariance & Equivariance. In addition to the invariance/equivariance of GNNs with respect to the isomorphism (Maron et al., 2019b;a; Keriven & Peyré, 2019), researchers have investigated the invariance / equivariance of GNNs to other types of permutations/isomorphisms. For example, the authors in (Thiede et al., 2021) and (de Haan et al., 2020) have investigated the invariance/equivariance of GNNs with respect to automorphisms and isomorphisms in the local structures in the graph, respectively. Researchers have also investigated the properties of GNN that are approximately equivariant to a set of permutations (Huang et al., 2023a).

Invariance & Equivariance of Higher-Order Networks. There have also been investigations of the GNN invariance/equivariance with respect to permutations in higher-order networks, e.g., hyper-graphs where each edge may connect more than two nodes (Kim et al., 2022; Hy et al., 2018; Papillon et al., 2024; Papamarkou et al., 2024; Huang et al., 2024), for which dedicated architectures and benchmarks continue to be developed (Li et al., 2025). Our work differs from this line of research in that we explore the invariance/equivariance of GNNs to variations other than permutations over graphs.

Geometric Graphs. Sometimes, graphs contain information about geometric objects, e.g., positions of objects in a 3D space (geometric graphs). Researchers have proposed deep learning architectures for geometric graphs that are invariant/equivariant to equivalence-preserving geometrical variations, such as rotation or translation (Bronstein et al., 2021; Schütt et al., 2017; Gasteiger et al., 2020; Joshi et al., 2023; Thomas et al., 2018). These variations are specific to geometrical domains, whereas we consider mappings that are applied in general graphs. Moreover, as opposed to our mappings, the aforementioned geometrical mappings do not modify the relationships between nodes/objects.

GNN Robustness Against Noise. Several GNN architectures have been proposed to improve robustness against both node-level and graph-level noise (Xia et al., 2023; Tian et al., 2020). Prior work primarily addresses robustness to adversarial attacks or out-of-distribution generalization, often under the assumption that one graph contains more or less information than another. In contrast, our work examines scenarios in which two graphs encode the *same* information but differ in their topological structure.

D τ-Invariance/Equivariance of Current Methods

D.1 Message Passing Neural Networks

There are various neural architectures for learning graph representations (Ju et al., 2025; Hamilton, 2020; Yang et al., 2020; Dai et al., 2020; Cao et al., 2024). **Message passing neural networks (MPNN)** are a prominent architecture for GNN's that use the graph structure to learn representations (Hamilton, 2020). Let $h_v^{(l)}$ represent the feature of node v at layer l. Each layer of an MPNN is composed of two learnable functions, the permutation invariant aggregate function and the update function:

$$h_v^{(l+1)} = \text{UPDATE}\left(h_v^{(l)}, \text{AGGREGATE}\left(\{h_u^{(l)} : u \in \mathcal{N}(v)\}\right)\right). \tag{3}$$

Repeated layers allow information to propagate through the graph. As (basic) MPNN has some limits in distinguishing non-isomorphic graphs (e.g., (Xu et al., 2019)), researchers incorporate additional

information on the structure of the graph, e.g., sets of paths (Fu et al., 2020; Velingker et al., 2023; Zhang et al., 2023). or add a higher-order structure (Morris et al., 2019) to MPNNs. Due to the popularity of MPNNs, in this paper, we focus on these architectures.

D.2 τ-INVARIANCE/EQUIVARIANCE OF MPNNS

Current MPNN-based architectures use the graph structure to learn the graph representation (Hamilton, 2020). For example, the MPNN architecture in Equation 3 shares the information of each node v with its neighbors in each layer (Veličković et al., 2018; Schlichtkrull et al., 2018; Kipf & Welling, 2016; Gilmer et al., 2017; Xu et al., 2019). Given GNN f, graph G, and $v \in V(G)$, let $\mathcal{M}^f(G, v)$ be a permutation equivariant function that returns a non-empty and proper subset of $V(G)$, i.e., $\mathcal{M}^f(\pi(G), \pi(v)) = \pi(\mathcal{M}^f(G, v))$. Many node-centric architectures f extend Equation 3 by sharing information between each node $v \in V(G)$ and $\mathcal{M}^f(G, v)$ in each layer.

$$h_v^{(l+1)} = \text{UPDATE}\left(h_v^{(l)}, \text{AGGREGATE}\left(\{h_u^{(l)} : u \in \mathcal{M}^f(G, v)\}\right)\right). \tag{4}$$

$\mathcal{M}^f(G, v)$ is often defined using the structural properties of the connections between v and $u \in \mathcal{M}^f(G, v)$. For example, some architectures set $\mathcal{M}^f(G, v)$ to (all) nodes on all simple or shortest paths from v whose lengths are at most $K > 0$ (Michel et al., 2023; Graziani et al., 2024; Sun et al., 2022) and some methods set $\mathcal{M}^f(G, v)$ to $k \leq K$-hop neighbors of v where $K = 1$ recovers the basic MPNN (Nikolentzos et al., 2020; Abu-El-Haija et al., 2019; Wang et al., 2021; Papp & Wattenhofer, 2022; Feng et al., 2022). There are also architectures that define $\mathcal{M}^f(G, v)$ using homomorphism or isomorphism from a fixed set of (labeled) graphs $\mathbf{F}$: $v \cup \mathcal{M}^f(G, v)$ are the nodes of homomorphism (isomorphism) images from $H \in \mathbf{H}$ in G (Maehara & NT, 2025; Fu et al., 2020; Zhao et al., 2021a; Yang et al., 2023). These methods may aggregate additional structural information about the nodes in $\mathcal{M}^f(G, v)$, e.g., their distances to v (Graziani et al., 2024). Since a bijective mapping τ modifies the structure of the graphs, e.g., the neighborhood structure of some nodes as in Figure 1, $\mathcal{M}^f(G, v)$ and $\mathcal{M}^f(\tau(G), v)$ are often different. Hence, these architectures will share information between different sets of nodes in some G and $\tau(G)$ and may not be τ-invariant/equivariant. Given architecture f, mapping τ, graph G, and node $v \in V(G)$, let UPDATE and AGGREGATE return different results for $\mathcal{M}^f(G, v)$ and $\mathcal{M}^f(\tau(G), v)$ if $\mathcal{M}^f(G, v) \cap \mathcal{M}^f(\tau(G), v) = \emptyset$ and $|\mathcal{M}^f(\tau(G), v)| \neq |\mathcal{M}^f(\tau(G), v)|$. These are mild conditions and are often satisfied by typical AGGREGATE and UPDATE functions. In this case, we have the following result.

Proposition 8. *Consider the architecture f with $\mathcal{M}^f(G, v)$ in Equation 4. There is a bijective mapping τ such that f is not τ-invariant or τ-equivariant.*

Proof. If the graph is directed, we set τ to the bijective mapping that reverses the direction of the edges. We can choose G so that we have $\mathcal{M}^f(G, v) \cap \mathcal{M}^f(\tau(G), v) = \emptyset$ and $|\mathcal{M}^f(G, v)| \neq |\mathcal{M}^f(\tau(G), v)|$. f will learn different representations for G and $\tau(G)$ in each layer. Assume that G is not directed, i.e., if $(u, v) \in E(G)$ we have $(v, u) \in E(G)$. The mapping τ that complements the graph G, i.e., $(u, v) \in E(G)$ iff $(u, v) \notin E(\tau(G))$, is bijective. We can choose G so that the degree of each v in G is not equal to its degree in $\tau(G)$. Since $\mathcal{M}^f(G, v) \cap \mathcal{M}^f(\tau(G), v) = \emptyset$ and $|\mathcal{M}^f(G, v)| \neq |\mathcal{M}^f(\tau(G), v)|$, f will learn different representations over G and $\tau(G)$. □

The proof of Proposition 8 holds for both directed and undirected graphs, as well as labeled and unlabeled ones (all nodes share the same label).

Subgraph MPNNs. To improve the accuracy of node-centric GNNs, some GNNs partition the input graph into multiple subgraphs based on a fixed policy, learn the representation for each subgraph using the methods described in the preceding paragraph, and aggregate these representations (Papp et al., 2021; Bevilacqua et al., 2021; Frasca et al., 2022). Since they use node-centric MPNNs to learn the representation of each subgraph, the result of Proposition 4 extends to these architectures.

D.3 τ-INVARIANCE/EQUIVARIANCE OF HIGHER-ORDER MPNN

Higher-order MPNN (HO-MPNN) learn representations for higher order structures (Morris et al., 2019; 2020; Besta et al., 2026; Lanzinger & Barcelo, 2024; Zhu et al., 2021; Huang et al., 2023b). Most notably. they extend the MPNN architecture to learn the representation for every tuple of $k > 1$ nodes in $V(G)$, $\bar{t}_k \in V(G)^k$. The initial representation of $\bar{t}_k$ encodes the relationship between its nodes in the graph. That is, let $G[\bar{t}_k]$ denote the induced subgraph of nodes in $\bar{t}_k$ in G. Tuples $\bar{t}_k$ and

$\bar{t'}_k$ have identical initial representation iff there is an isomorphism mapping between $G[\bar{t}_k]$ and $G[\bar{t'}_k]$. Let $C(\bar{t}_k, u, j)$ denote the tuple in which node $v_j \in \bar{t}$ is replaced by node $u \in V(G)$, $1 \leq j \leq k$ and let $\mathcal{C}(\bar{t}_k)$ be the set of all $C(\bar{t}_k, u, j)$. In each layer, the HO-MPNN shares the information between the tuple $\bar{t}_k$ and the tuples in $\mathcal{C}(\bar{t}_k)$ using the UPDATE and AGGREGATION functions similar to Equation 3.

Since it often takes too long to learn representations using this approach due to the large number of tuples, some HO-MPNN share the information of each tuple with a subset of $\mathcal{C}(\bar{t}_k)$ where u and v_j are neighbors in G, i.e., **local HO-MPNN** (Zhu et al., 2021; Huang et al., 2023b). In this paper. we refer to both variations as HO-MPNN.

Since the initial values of the tuples $\bar{t}_k$ reflect the structure of the graph and since bijective mappings τ modify this structure, these GNNs are not τ-invariant. More precisely, assume that the UPDATE function returns a different value if at least one of its input elements is different. Also, assume that the AGGREGATION function returns a distinct value if at least one of the aggregated tuples is different. This is a mild condition that is satisfied by typical UPDATE and AVERAGE functions. We will have the following results for the HO-MPNN described above.

Proposition 9. *There is a bijective mapping τ such that HO-MPNN f is not τ-invariant or τ-equivariant.*

Proof. Let $\bar{t}^G_k$ ($k > 1$) denote a tuple of k nodes on G. We assume that τ does not permute the node ids. Our proof will extend to the cases where τ also permutes the node ids. Since τ does not permute nodes, there is an identity mapping between tuples of k nodes on G and $\tau(G)$, i.e., for each $\bar{t}^G_k$ there is one and only one $\bar{t}^{\tau(G)}_k$ where $\bar{t}^G_k = \bar{t}^{\tau(G)}_k$. We will use the same mapping used in the proof of Proposition 4. In both mappings, for every $\bar{t}^G_k$ $\bar{t^\tau}(G)_k$, $G[\bar{t}^G_k]$ is not isomorphic to $\tau(G)[\bar{t}^{\tau(G)}_k]$. Thus, in each layer, f will learn a distinct representation for $G[\bar{t}^G_k]$ and $\tau(G)[\bar{t}^{\tau(G)}_k]$. □

The proof of Proposition 9 covers graphs with/without directed edges and labels.

E PROOFS

E.1 PROOFS FOR SECTION 2

Proof of Proposition 1. Given the switching $\tau = \{P_i, P'_i\}^t_{i=1}$, we use the Chase algorithm from the data/graph transformation literature (Barcelo et al., 2013; Fagin et al., 2005) to implement the mapping for a given set of switching τ. More precisely, to compute $\tau(G)$, Chase starts with an empty graph G'_k, $k = 0$. In step $k > 0$, for each $\{P_i, P'_i\} \in \tau$, Chase finds every homomorphism image of P_i in G (p_i), computes the corresponding isomorphic subgraph to P'_i based on switching $\{P_i, P'_i\}$, p'_i. The node id, label, and features of p'_i is set using node id, label, and features of p_i to satisfy the conditions of mappings under switchings in Section 2. It sets $G'_k = G'_k \cup p'_i$. The algorithm ends when there is no application of any switching left with G'_k as output. We represent each switching $\{P_i, P'_i\}$ using a first order logic formula over G in the form $\phi(\bar{x}) \rightarrow \psi(\bar{y})$, where ϕ (ψ) is a conjunction of binary predicates. Each predicate in ϕ (ψ) corresponds to an edge in P_i (P'_i) and expresses a relationship between two labeled nodes, e.g., $(R(x_1), S(x_2))$. Each variable $x \in \bar{x}$ and $y \in \bar{y}$ denotes a distinct node in P_i (P'_i). Due to the definition of mappings under switchings, $\bar{y} = \bar{x}$ and all variables in ϕ and ψ are universally quantified. The termination of the algorithm follows from Theorem 5.2 in (Barcelo et al., 2013). The uniqueness follows from (Fagin et al., 2005). Since the conditions of a mapping expressed using a set of switching is symmetric, these results also hold for the switching $\{P'_i, P_i\}^t_{i=1}$ over $\tau(G)$ as input. Hence, τ defines a unique bijective mapping. □

Proof of Proposition 2. If the conclusion does not hold, then the mapping τ will not be bijective. □

Proof of Proposition 3. From the definition of switchings, it follows that given the mapping $\tau = \{(P_i, P'_i)\}^t_{i=1}$, we can represent τ^{-1} as $\{(P'_i, P_i)\}^t_{i=1}$.

Using the definition of switching and mapping simplification method in (Barcelo et al., 2013), the set of switchings σ is equivalent to the mapping σ' where for each switching (S_i, S'_i) in σ, σ' has a set of pairs (S_i, s'^k_i) such that each s'^k_i is a distinct edge in S'_i and the union of s'^k_i is S'_i. Although

σ' is equivalent to σ, it is not expressed as a set of switchings because there may not be a bijection between the nodes in some pairs of graphs in σ'. Next, we construct the switching for $\tau \circ \sigma$. First, for each switching (P_j, P'_j) in τ and every s'^k_i in σ', let p^m_j be a homomorphism image (preserving the labels) of s'^k_i in P_j. Let τ' be the mapping represented by replacing every homomorphism image of s'^k_i in P_i with S_i. The mapping τ' describes the transformation of the graph, but it may not be a set of switchings because there may not be any bijection between the nodes in some of the pairs of graphs in τ'. Let (R, L) be such a pair. Let f be function that maps nodes in R to those in L. Let node v in R be the node that does not have any corresponding node in L using f. Since $\tau \circ \sigma$ is bijective on $\mathcal{G}$, τ' must map a subgraph of R, $R' \subseteq R$, that contains v, therefore, R' must appear as the first component of a pair (R', L') in τ'. We replace (R, L) in τ' with $(R, L \cup L')$ to create a switching. If we apply this for all uncovered nodes and pairs in τ', the result will be a set of switchings. □

Proof of Theorem 1.

We prove the theorem for both cases that the mapping τ can be expressed as a set of switching and the case when it may not be expressed using switchings. We first prove the following lemma that indicates that we can construct a set of switchings $\iota = \{(P_i, P'_i)\}^t_{i=1}$ that implement identity mapping: the permutation that preserves node ids.

Lemma 2. *Given a family of graphs $\mathcal{G}$, there is a set of switchings ι equivalent to the permutation that preserves node ids on $\mathcal{G}$.*

Proof. The set of switchings $\tau = \{(P_i, P'_i)\}^t_{i=1}$ maps each graph to itself (identity) if every P_i is either a graph with a single node or a simple path with two nodes, every node and edge in the graph is isomorphic to a single P_i and P_i is isomorphic to P'_i where the isomorphism mapping preserves labels. Since every set of switchings preserves the node ids, ι is equivalent to the permutation that preserves node ids. □

Next, we prove Theorem 1:

Proof. Given a family of graphs $\mathcal{G}$ and the set of switchings τ on $\mathcal{G}$, $\Gamma^\tau \cup \iota$ forms a group with composition operation with ι as the identity element. Generally, given the family of graphs $\mathcal{G}$, let Φ be the set of all mappings created by compositions of the members in a set of distinct switchings $\mathcal{T} = \{\tau_i\}^k_{i=1}$ or their inverses. Since switchings are bijective, the inverse of the composition of two switchings $(\tau \circ \rho)^{-1}$ is the composition of their inverses $\rho^{-1} \circ \tau^{-1}$. Hence, $\Gamma^\tau \cup \iota$ is a group with composition operation on $\mathcal{G}$. Let $\mathcal{G}^n$ be the family of graphs with n nodes. Given an isomorphism-closed set of graphs $\mathcal{G} \subseteq \mathcal{G}^n$, let Π^n be the group of permutations with composition on $\mathcal{G}$. Let $\pi_\mathbf{I}$ be the permutation mapping on $\mathcal{G}$ that maps the id of each node to itself, i.e., the identity element in Π^n. Since ι and $\pi_\mathbf{I}$ express the same mapping on $\mathcal{G}$, $\Gamma^\mathcal{T} \cup \Pi^n$ is a group with composition on $\mathcal{G}$.

If the mapping τ cannot be expressed as switching, we use the fact that an identity permutation is a bijection. The rest of the proof is similar to the case where the mapping is expressed as a set of switchings. □

E.2 Proofs for Section 3

Proof of Proposition 4. If the graph is directed, we set τ to the bijective mapping that reverses the direction of the edges. We can choose G so that we have $\mathcal{M}^f(G, v) \cap \mathcal{M}^f(\tau(G), v) = \emptyset$ and $|\mathcal{M}^f(G, v)| \neq |\mathcal{M}^f(\tau(G), v)|$. f will learn different representations for G and $\tau(G)$ in each layer. Assume that G is not directed, i.e., if $(u, v) \in E(G)$ we have $(v, u) \in E(G)$. The mapping τ that complements the graph G, i.e., $(u, v) \in E(G)$ iff $(u, v) \notin E(\tau(G))$, is bijective. We can choose G so that the degree of each v in G is not equal to its degree in $\tau(G)$. Since $\mathcal{M}^f(G, v) \cap \mathcal{M}^f(\tau(G), v) = \emptyset$ and $|\mathcal{M}^f(G, v)| \neq |\mathcal{M}^f(\tau(G), v)|$, f will learn different representations over G and $\tau(G)$. □

E.3 Proofs for Section 4

Proof of the Lemma 1. Since the PAGG function in Equation 2 receives equal inputs on G and $\tau(G)$, given the same initializations, in each layer, f will learn the same representation on G and $\tau(G)$. □

Proof of Proposition 5. Given GNN f, we construct a mapping τ where there is a switching (P_i, P'_i), $1 \leq i \leq t$, where P_i is not isomorphic to P'_i and P_i and P'_i do not belong to $\mathcal{S}^f$. An example of τ

is the mapping shown in Figure 1 and MPNN that shares the information between 1-hop neighbors where P_i and P'_i are not isomorphic and do not belong to $\mathcal{S}^f$ of MPNN. If the structural bias contains P_i and P'_i, we create another mapping by adding a single node to P_i and P'_i. □

Proof of Theorem 2. Since τ is minimal, there are no two switchings (P_i, P'_i) (P_j, P'_j), $i \neq j$ such that P_i (P'_i) and P_j (P'_j) are isomorphic. There are no two isomorphic graphs in $\mathcal{S}^f$. Thus, the steps of the algorithms are applied to each $S^u \in \mathcal{S}^f$ separately, and the algorithm does not generate any two isomorphic CGs.

The mapping b_τ is isomorphism-preserving for CG S^u if it is an isomorphism-preserving mapping with respect to S^u_C as defined before and there is a bijection between $pom(S^u, G^v)$ and $pom(b_\tau(S^u), \tau(G^v))$ that maps every member of $pom(S^u, G^v)$ to an isomorphic member in $pom(b_\tau(S^u), \tau(G^v))$ where the isomorphism mapping preserves node ids/labels/features. For every S^u in the input structural bias $\mathcal{S}^f$, the algorithm places one and only one CG S^u where every edge is context and propagation. For every such CG S^u, the algorithm produces a CG such that there is an isomorphism-preserving mappings between them. Also, for each newly generated CG in step 2 S^w, the algorithm produces one and only one CG $S'^{w'}$ in step 3 such that there is an isomorphism-preserving mapping between S^w and $S'^{w'}$. Since the algorithm does not generate any other types of CG, there is a isomorphism-preserving for every CG in $\mathcal{S}^f_\tau$. Thus, f using $\mathcal{S}^f_\tau$ will be τ-invariant. □

Proof of Proposition 6. Suppose we remove a CG S^u from $\mathcal{S}^f_\tau$, then there will be no isomorphism-preserving mapping in $\mathcal{S}^f_\tau$. Let G be a graph that has at least one occurrence of S^u_C. Then, the GNN will not be τ-invariant on G and $\tau(G)$. Additionally, if we remove any context or propagation edge from a $S \in \mathcal{S}^f_\tau$ there will be no $\mathcal{S}^f_\tau$. □

E.4 Proofs for Section 5

Proof of Theorem 3. Following standard and established assumptions in the theoretical analysis of expressivity with regard to isomorphism (Xu et al., 2019; Morris et al., 2019; Geerts & Reutter, 2022), we assume that the functions PAGG, TAGG, and UPDATE in Equation 2 in the input GNN to the algorithm in Section 4.3 are injective. To simplify our proof, we assume that all nodes in the graph share the same label and that none have any features. Our proof extends to other cases. Given the non-isomorphism mapping τ, we have $\mathcal{S}^f \subset \mathcal{S}_\tau{}^f$. Thus, for a graph G and each node $v \in V(G)$, the inputs to PAGG, TAGG, and UPDATE in each layer of f_τ are superset or equal to their inputs for v in the same layer in f. Consider graph G' where the results of TAGG and PAGG are different on G and G' in f. Thus, since PAGG, TAGG, and UPDATE are injective, for the same node initializations, their results will be different for G and G' in f_τ. Thus, f_τ is at least as expressive as f.

To prove the second part, we also show that improving the structural bias using this algorithm enables f_τ to access more information about the graph that is not present in the original f bias. That is, there are GNN f and mapping τ such that f_τ is strictly more expressive than f. Consider the mapping δ that reverses the direction of all edges in the input graph. Assume that we use the algorithm in Section 4.3 to produce a δ-invariant f (f_δ). f_δ will propagate the information in both directions of an edge. For example, consider the graphs G and H where $V(G) = V(H) = 1, 2, 3, 4$, $E(G) = \{(1, 2), (3, 2), \}$ and $E(H) = \{(1, 2), (3, 4), \}$. The MPNN that shares information only between 1-hop neighbors (1-WL) does not distinguish G and H, but MPNN$_\delta$ does. □

Proof of Proposition 7. If $f \preceq f'$, we have $\mathcal{S}^f \subseteq \mathcal{S}^{\{'}$. Given a mapping τ, we will have $\mathcal{S}^{f_\tau} \subseteq \mathcal{S}^{\{'_\tau}$, therefore; $f_\tau \preceq_\tau f'_\tau$. □

F More Details on Structural Bias

We explain how MPNN based on structural bias in Equation 2 generalizes multiple families of popular (node-centric) MPNN architectures. Some of these families cover only unlabeled graphs. Our formulation covers these cases by assigning the same label to all nodes. Some of them consider undirected graphs. Our abstraction generalizes these cases by directed edges in both directions between all two neighboring nodes in the graph.

K-hop Message Passing. Some MPNN architectures share information between node $v \in V(G)$ and all its $0 < k \leq K$-hop neighbors in G (Nikolentzos et al., 2020; Abu-El-Haija et al., 2019; Feng et al., 2022; Wang et al., 2021; Papp & Wattenhofer, 2022). Since these methods ignore labels on the nodes, the structural bias for these architectures contains only one simple path of length (number of edges) K. In this case, the occurrences of the graph of the structural bias in the input graph are computed without the restriction of label-preservation. The PAGG function varies according to the structural information shared between v and its k-hop neighbors. For example, some methods share with v the shortest distance of its k-hop neighbor, which might be less than k. The PAGG function has access to this information using its input. The exat function PAGG varies on the basis of the underlying architecture. The TAGG function is a function that maps a set of a single vector to the vector, and the UPDATE function will be the same as UPDATE in these architectures.

Path-based MPNNs. Some MPNNs share information of $v \in V(G)$ with all nodes in $V(G)$ whose shortest path to v has a length shorter than or equal to $K > 0$ (Michel et al., 2023; Graziani et al., 2024; Sun et al., 2022). The structural bias in these methods is similar to the preceding family. They have PAGG functions different from those of the preceding family and extract different types of information from these paths.

MPNNs Based on Meta-Paths. Some MPNNs graphs share information between $v \in V(G)$ and all other nodes on a set of simple paths that start in v with a given label l_1 and end with a node with a label l_1, and the node labels on each path follow a given sequence of labels $l_2, \ldots, l_k, \ldots, l_2$ (symmetric paths) (Fu et al., 2020; Yang et al., 2023). The structural bias for these families contains a set of paths where the labels at the nodes of each path follow the given sequence of labels $(l_1, l_2, \ldots, l_k, \ldots, l_2, l_1)$.

Homomorphism-based MPNNs Like our abstraction for structural bias, homomorphism-based MPNNs propagate and aggregate information by computing the occurrences of a set of fixed rooted graphs in the input graph (Maehara & NT, 2025). For each occurrence of a graph in the structural bias in the input graph, the PAGG functions in these methods may require both the occurrence and the homomorphism mapping used to produce the occurrence. We can slightly modify Equation 2 and include both ocurrences and their homomorphism mappings as the input. Our results in the paper extend to these cases.

Different Structural Bias per Layer. Some algorithms use a different $\mathcal{S}$ in each layer (Maehara & NT, 2025). Our definition of structural bias covers these cases by a distinct set of rooted pattern graphs for each layer in $\mathcal{S}$.

G Completeness of Structural Bias

We call a structural bias $\mathcal{S}^f$ *complete* if for every graph G and every node $v \in G$ there is at least one $S^u \in \mathcal{S}^f$ such that $hom(S^u, G^v)$ is not empty. Intuitively, a complete structural bias for $\mathcal{G}$ ensures that the information for each node in the input graph G is shared with some other nodes in G. It is often desirable for a structural bias to be complete, so GNN learns the representation for every node in the graph. If the original structural bias is complete, the algorithm generates a complete structural bias.

Proposition 10. *If $\mathcal{S}^f$ is complete, ${\mathcal{S}_\tau}^f$ is complete.*

Proof. Since the algorithm preserves all original structural biases $\mathcal{S}^f$ (first step), and $\mathcal{S}^f$ is complete, $\mathcal{S}^f_\tau$ is complete. □

H Experiment Details

Code to run experiments can be found here: `https://anonymous.4open.science/r/INV-GNN-12D9/README.md`

H.1 Datasets & Mappings

We evaluate on four datasets. In every case the variants are obtained by normalizing or denormalizing the graph with respect to a dependency that holds in the data (Skavantzos & Link, 2023; Egger

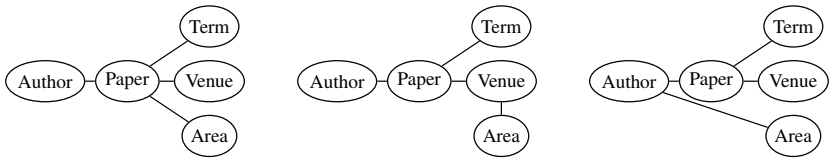
et al., 2026), which is the process that Section 2 represents as switchings; what differs is how the dependency is obtained. On IMDb and DBLP, the heterogeneous graphs of (Fu et al., 2020), the dependency is read off the documented schema of the dataset and the switching is written down from it, so the semantics of the mapping are known exactly. On WN18RR and Freebase, knowledge graphs one to four orders of magnitude larger, no such schema is available and the candidate dependencies are enumerated from the data and filtered by an explicit criterion; these are also graphs in which the redundancies our mappings encode — inverse relations, transitive and composed relations — occur naturally (Sun et al., 2019; Hogan et al., 2014). The two groups play complementary roles: the first lets us verify invariance against a mapping we can state in closed form, while the second tests whether the approach scales to large graphs with hundreds of relation types and whether it depends on switchings being supplied by a person. In every dataset, the variants share the same nodes, features, labels, and train/validation/test splits; only the way relationships are encoded differs. Held-out target edges (link prediction) and held-out labels (node classification) are never used to construct a variant. Statistics are in Tables 2–5; the universal graph of each dataset, denoted $\bigcup_i \mathrm{D}_i$, contains the edges of all of its variants (Figure 4); for WN18RR it also contains additional mined relations, as described below.

What licenses a switching. What the dependency has to do is the same in all four cases: it must make the rewrite invertible occurrence by occurrence, which is the condition Section 2 imposes on a switching, that the occurrences of P in G correspond one to one with the occurrences of P' in $\tau(G)$. The datasets discharge it in three ways. IMDb uses a key: the link node is in one-to-one correspondence with the movie, so an attribute moved onto it can always be moved back (Figure 2, Table 2). DBLP uses uniqueness within an occurrence: a paper has exactly one area, and for the one variant that does not record which occurrence an area belongs to, that dependency is supplied alongside the graph (Figure 3, Table 3). WN18RR and Freebase only ever *add* relations that the base graph already determines, so the inverse of the switching is to drop them again, and the criteria below decide which compositions qualify (Tables 4 and 5).

IMDb. IMDb (Fu et al., 2020) contains movies, directors, and actors; we add a *year* node per release year and a *link* node that is a one-to-one copy of each movie. The mapping relocates the two attribute relations, movie–director and movie–actor, between the movie and its link node: both stay on the movie (IMDb1), both move to the link node (IMDb2), the actor alone moves (IMDb3), or the director alone moves (IMDb4) (Figure 2(a)–(d)). Every variant keeps the movie–link and movie–year edges. This is a denormalization of the movie, with the link node playing the role of the movie's key. Movie–link is 1:1 by construction, one link node per movie, so Link $\to$ Movie and Movie $\to$ Link are both functional dependencies. An occurrence of the switching is therefore a single movie with its link node and its attribute edge, and each of the two nodes determines the other. The relocation loses no information and is invertible from the graph alone: nothing outside the occurrence is needed to decide which movie an attribute attached to a link node belongs to. Movie features are a 3,008-dimensional bag of words over plot keywords; the features of the other node types are the mean of their movies' features. Tasks: movie genre classification (3 classes) and Movie–Director link prediction (IMDb1 and IMDb3, the two variants in which the director stays attached to the movie).

DBLP. DBLP (Fu et al., 2020) contains authors, papers, terms, and venues; we add an *area* node per research area (four areas). The mapping attaches a paper's area to the paper (DBLP1), to its venue (DBLP2), or to its authors (DBLP3) (Figure 3). The task is Paper–Venue link prediction, ranked over all 20 venues. Area edges are only present for papers in the training set, since the area of a held-out paper would narrow down its venue.

The three variants are three normalizations of the same schema, and the dependencies in the data are what make each of them invertible. Every author carries exactly one of the four areas (Author $\to$ Area is N:1), and a paper's area is *defined* as the area of its authors; the 866 papers whose authors disagree are removed so that Paper $\to$ Area is N:1 as well. Each variant stores that single area in a different place, and all three can be inverted; what differs is what the inversion needs. DBLP1 attaches it to the paper and DBLP3 to the paper's authors, who all carry the same one, so both invert from the graph alone — DBLP3 even though Paper–Author is $N{:}M$, since the composition Paper $\to$ Author $\to$ Area stays functional. DBLP2 attaches it to the paper's venue, whose area edges record the areas of the papers published there rather than the venue's own subject label, so Venue–Area is $N{:}M$ and the composition Paper $\to$ Venue $\to$ Area is $N{:}M$ with it. That composition is nonetheless *unique within an occurrence*: an occurrence of this switching is one paper with its venue and its area, and the paper has exactly one area, so it is one-to-many only once occurrences are pooled over a venue.



(a) DBLP1 **(b)** DBLP2 **(c)** DBLP3

Figure 3: Mappings for DBLP.

What DBLP2 omits is therefore not the value but which occurrence each of a venue's areas belongs to, and inverting it takes exactly that, the Paper → Area dependency, which we supply alongside the variant (Section H.3).

WN18RR. WN18RR (Dettmers et al., 2018) is the standard link prediction benchmark derived from WordNet (Miller, 1995): 40,943 entities of a single type and 11 relations, with the official training, validation, and test triples. The training triples form the graph of WN18RR1. We derive two variants by adding relations that encode information already present in the training graph. WN18RR2 adds, for every relation r, a named inverse relation r^{-1} (e.g., *hypernym* and *hyponym*). WN18RR3 adds *shortcut* relations: for every sequence of three relations $r_1; r_2; r_3$, we connect entity u to entity w by a new relation named $r_1; r_2; r_3$ whenever there is a path from u to w following r_1, r_2, and r_3; a sequence is kept if it has at least one such path in the training graph, which yields 526 shortcut relations. The universal graph contains the inverse and shortcut relations of both variants and, in addition, 1,539 shortcut relations for sequences that mix relations and their inverses; these are kept only when the resulting relation is reversible, i.e., it defines a one-to-one correspondence between groups of source and target entities, so that the paths it summarizes can be recovered from it. Every shortcut is mined from the training triples only; validation and test triples are never used. The task is link prediction on the official test triples with filtered ranking over all entities.

Freebase. Freebase (Bollacker et al., 2008) is a large knowledge graph; we use the version of the Heterogeneous Graph Benchmark (Lv et al., 2021) with 180,098 entities of eight types (book, film, music, sports, people, location, organization, business), 36 typed relations, and 1,057,688 edges. The task is genre classification of the 7,954 labelled books. Freebase1 is the original graph. As in WN18RR, the other variants add shortcut relations that summarize paths, here typed paths that pass through a book: Freebase2 adds one relation per kept two-hop sequence (6 relations) and Freebase3 one per kept three-hop sequence (22 relations). A sequence is kept only if no existing relation already connects the same pair of node types, so that each shortcut occupies a new position in the schema rather than adding edges to an existing relation; this matters for architectures whose information propagation is keyed on node-type pairs, e.g., SeHGNN and SlotGAT. The shortcut variants are large: Freebase3 has 447 million edges and the universal graph 493 million.

H.2 Methods

GNN architectures. We evaluate GNN architectures that differ in the structural object that determines how information is propagated, i.e., in their structural bias (Section 4): RGCN (Schlichtkrull et al., 2018) (relation-typed local message passing), SlotGAT (Zhou et al., 2023) (node-type-aware local attention), CMPNN (Huang et al., 2023b) (source- and relation-conditioned path propagation), MAGNN (Fu et al., 2020) (metapath-instance message passing), SeHGNN (Yang et al., 2023) (pre-computed metapath feature propagation), PAIN (Graziani et al., 2024) (path-based message passing), and PathNet (Sun et al., 2022) (sampled-path aggregation). Node classification is run on IMDb and Freebase with RGCN, MAGNN, SlotGAT, SeHGNN, PAIN, and PathNet; link prediction on IMDb, DBLP, and WN18RR with RGCN, CMPNN, and SlotGAT, and additionally MAGNN on IMDb and DBLP (MAGNN's node-type metapaths degenerate on the single-type WN18RR graph). PAIN and PathNet are run on node classification only, since PathNet is designed for node classification under heterophily and provides no link scoring.

RGCN. RGCN (Schlichtkrull et al., 2018) extends the graph convolutional network to multi-relational graphs. In each layer, a node aggregates normalized messages from its one-hop neighbors separately for every relation type, applies a relation-specific linear transformation (with basis decomposition to share parameters across relations), and sums the results. On IMDb and DBLP the relation types are the node-type pairs of the incident nodes; on WN18RR and Freebase they are the named relations of

Table 2: Statistics of the IMDb variants. Node counts are the same in every variant; edge counts are undirected pairs per relation (M: movie, D: director, A: actor, L: link, Y: year). The held-out target pairs of each link prediction task are absent from the graph, hence the smaller M–D count in that column. Movie–Director uses IMDb1 and IMDb3 only.

Nodes: # movie (M): 4,180 # director (D): 2,081 # actor (A): 5,257 # link (L): 4,180 # year (Y): 91		
Variant	**Edges (task dependent)**	
	NC: Genre	**LP: Movie–Director**
IMDb1	M–D: 4,180 M–A: 12,534 M–L: 4,180 M–Y: 4,177	M–D: 2,926 M–A: 12,534 M–L: 4,180 M–Y: 4,177
IMDb2	L–D: 4,180 L–A: 12,534 M–L: 4,180 M–Y: 4,177	–
IMDb3	M–D: 4,180 L–A: 12,534 M–L: 4,180 M–Y: 4,177	M–D: 2,926 L–A: 12,534 M–L: 4,180 M–Y: 4,177
IMDb4	L–D: 4,180 M–A: 12,534 M–L: 4,180 M–Y: 4,177	–
$\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	M–D: 4,180 M–A: 12,534 M–L: 4,180 M–Y: 4,177 L–D: 4,180 L–A: 12,534	M–D: 2,926 M–A: 12,534 M–L: 4,180 M–Y: 4,177 L–A: 12,534 ($i \in \{1, 3\}$)
Supervision	2,926 / 418 / 836 movies 3 classes	2,926 / 418 / 836 pairs 19 negatives per pair

Table 3: Statistics of the DBLP variants for Paper–Venue link prediction. Edge counts are undirected pairs per relation (A: author, P: paper, T: term, V: venue, Ar: area). Held-out Paper–Venue pairs are absent from the graph; area edges exist for training papers only.

Nodes: # author (A): 3,948 # paper (P): 13,462 # term (T): 8,642 # venue (V): 20 # area (Ar): 4	
Variant	**Edges**
DBLP1	A–P: 17,568 P–T: 107,154 P–V: 9,448 P–Ar: 9,448
DBLP2	A–P: 17,568 P–T: 107,154 P–V: 9,448 V–Ar: 75
DBLP3	A–P: 17,568 P–T: 107,154 P–V: 9,448 A–Ar: 3,474
$\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	A–P: 17,568 P–T: 107,154 P–V: 9,448 P–Ar: 9,448 V–Ar: 75 A–Ar: 3,474
Supervision	9,448 / 1,348 / 2,666 pairs ranked over all 20 venues

Table 4: Statistics of the WN18RR variants for link prediction. All variants have 40,943 entities of a single type and contain the 11 base relations (the official training triples); the last column lists what each variant adds.

Variant	**# Relations**	**# Edges**	**Added relations (edges)**
WN18RR1	11	86,835	–
WN18RR2	22	173,670	11 inverse relations (86,835)
WN18RR3	537	706,128	526 three-hop shortcut relations (619,293)
$\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	2,087	893,629	11 inverse relations (86,835) 2,065 shortcut relations (719,959)
Supervision	86,835 / 3,034 / 3,134 triples; filtered ranking over all 40,943 entities		

Table 5: Statistics of the Freebase variants for book genre classification. Node counts are the same in every variant. Every variant contains the 36 base relations; the last column lists the added shortcut relations.

Nodes: # book: 40,402 # film: 19,427 # music: 82,351 # sports: 1,025 # people: 17,641 # location: 9,368 # organization: 2,731 # business: 7,153			
Variant	**# Relations**	**# Edges**	**Added relations (edges)**
Freebase1	36	1,057,688	–
Freebase2	42	47,368,991	6 two-hop shortcut relations (46,311,303)
Freebase3	58	446,967,136	22 three-hop shortcut relations (445,909,448)
$\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	64	493,278,439	28 shortcut relations (492,220,751)
Supervision	7,954 labelled books, 4,772 / 1,591 / 1,591; 7 classes		

the knowledge graph. RGCN is our representative of purely local, relation-typed message passing. For link prediction, node representations from the encoder are scored with a dot-product (IMDb, DBLP) or relation-specific (WN18RR) decoder.

SlotGAT. SlotGAT (Zhou et al., 2023) is a heterogeneous graph attention network that keeps the feature spaces of different node types separate. Every node maintains one representation *slot* per node type; a message from a neighbor is written into the slot of the neighbor's type, and edge-type features enter the attention scores. After message passing, a learned slot attention combines the slots into the node's output representation. SlotGAT is our representative of local, node-type-aware attention.

CMPNN. CMPNN (Huang et al., 2023b) (conditional message passing) is designed for link prediction on knowledge graphs. Given a query $(h, r, ?)$, it computes representations of all nodes *conditioned* on the source h and the queried relation r: the source is marked by a boundary condition, messages are relation-specific (DistMult) and aggregated with PNA, and the score of a candidate tail is read from its conditioned representation. The propagation thus follows the relational paths that start at the source, in the style of NBFNet, and its expressive power matches the relational Weisfeiler–Leman test. CMPNN is our representative of source-conditioned path propagation.

MAGNN. MAGNN (Fu et al., 2020) propagates information along *metapath instances*, i.e., concrete paths in the graph that follow a given sequence of node types. After a type-specific input projection, each instance is encoded from all of its nodes (we use the RotatE-style encoder), the instances of a metapath are aggregated with attention (intra-metapath aggregation), and the resulting per-metapath vectors are combined with a second attention (inter-metapath aggregation). MAGNN is our representative of metapath-instance message passing; the metapaths used for each dataset and variant are listed in Section H.3.

SeHGNN. SeHGNN (Yang et al., 2023) avoids neighbor aggregation during training. For every metapath up to a maximum length, it pre-computes once, before training, the mean of the neighbors' raw features along that metapath (a product of row-normalized adjacency matrices), optionally together with label-propagation channels. Each such feature channel is projected and the channels are fused by a transformer over the channel dimension. SeHGNN is our representative of metapath-based feature propagation, in which the structural bias is materialized as a set of pre-computed feature channels rather than through message passing.

PAIN. PAIN (Graziani et al., 2024) (Path Isomorphism Network) aggregates, for every node, over the set of paths of bounded length that start at that node, encoding each path as a sequence of node representations. Its expressive power is characterized by the PATH-WL hierarchy, which is incomparable to the standard k-WL hierarchy and can count cycles. For node classification we replace its graph pooling by a node-wise classifier. PAIN is our representative of path-based message passing, and its structural bias is the set of rooted paths of bounded length.

PathNet. PathNet (Sun et al., 2022) samples paths that start at the target node and aggregates them with a path-aware encoder, so that the representation of a node depends on the structure of the paths around it rather than only on its neighbors' labels; it is designed for node classification in heterophilous graphs. PathNet is our representative of sampled-path aggregation, and its structural bias is the set of sampled rooted walks.

Baselines and our approach. For every GNN architecture and task, we compare our approach against three baselines. Each of the four is trained on the same data with the same architecture, loss, optimizer, training budget, seeds, splits, labels, positive and negative pairs, and node features; they differ only in the structural input given to the GNN.

- *Original* (baseline): the GNN as published, trained and evaluated separately on each dataset variant. This measures how much the outputs of current GNNs depend on the representation of the graph.
- *Universal graph (canonicalization)* $\bigcup_i \mathrm{D}_i$ (baseline): the GNN trained once on the universal graph of the dataset (Section H.1, Figure 4). This is the canonicalization approach discussed in Section 4: every variant is mapped to the same canonical graph, so a single model is trained and its invariance is trivial. It requires the edges of every variant to be available at training time and serves as a reference for the effectiveness that a canonical form can reach.
- *Augmentation* (baseline): the GNN trained jointly on all variants, i.e., the data augmentation approach discussed in Section 4. One super-epoch is one epoch over every variant in a seeded random order with a single shared optimizer; the model is then evaluated on each variant. This

Table 6: Hyper-parameters for IMDb node classification (features: 3,008-d pooled bag of words for every node type). Settings are identical for the original, universal-graph, and augmentation baselines and for our invariant approach. "lr" is the learning rate and "wd" the weight decay; the optimizer is Adam throughout.

Model	Architecture	Optimization	Selection / stopping
RGCN	2 layers; input/hidden/output 128; 8 bases; dropout 0.3; mean normalization	lr 2×10^{-3}, wd 10^{-5}; full batch; cross-entropy; $\leq$ 300 epochs	val. Micro-F1; patience 50
MAGNN	hidden 64; 8 heads; attention vector 128; RotatE0 instance encoder; dropout 0.5	lr 5×10^{-3}, wd 10^{-3}; mini-batches of 8 target movies; 100 sampled instances per (node, metapath); cross-entropy; $\leq$ 100 epochs	val. Micro-F1; patience 5
SlotGAT	2 hidden layers + 1 output layer; hidden 64; 8 heads; slot attention (dim. 3); edge-type features 64; feature dropout 0.5, attention dropout 0.2; LeakyReLU slope 0.05	lr 5×10^{-3}, wd 10^{-4}; full batch; $\leq$ 300 epochs	val. Micro-F1; patience 40
SeHGNN	one pre-computed channel per metapath of length $\leq$ 4 plus label channels; per-type input projection; embedding/hidden 512; 2 projection layers; transformer fusion; 4 task layers; dropout 0.5	lr 5×10^{-3}, wd 10^{-4}; batch 10,000; $\leq$ 200 epochs	val. Micro-F1; patience 50
PAIN	5 layers; hidden 128; exact rooted paths of length 0–3; shared 2-layer LSTM with learned path-position embeddings and a root-neighbour marker; sum aggregation; 2-layer node head; dropout 0	lr 10^{-3}, wd 0; full graph; cross-entropy; gradient-norm clip 5; lr halved after 20 epochs without validation improvement (min 10^{-5}); $\leq$ 1000 epochs or 12 h	val. Macro-F1, val. loss breaking ties; patience 80
PathNet	hetero PathNetNC; hidden 64; 40 MERW walks $\times$ length 4; distance-specific linear + shared LSTM + walk attention; structure-only head; dropout 0.2; MAGNN movie features	lr 10^{-2}, wd 5×10^{-4}; full batch; cross-entropy; $\leq$ 300 epochs	val. Macro-F1; patience 100

is the standard non-architectural route to invariance: training on the orbit of τ is an averaging operation over that orbit (Chen et al., 2020; Zhao et al., 2023), and it is the usual baseline against which invariant-by-construction models are measured (Gerken et al., 2022; Nordenfors et al., 2025). Invariance is encouraged by the training data but not guaranteed (Lyle et al., 2020; Moskalev et al., 2023), which is why we report the invariance metrics of this arm rather than assuming them.

- *Invariant* (ours): the GNN with the invariant structural bias $\mathcal{S}_\tau^f$ of Section 4.3, instantiated for the architecture as described in Section H.3, trained and evaluated separately on each variant exactly as the original. Each model sees only its own variant, and invariance follows from Theorem 2 rather than from the training data.

Training protocol. Every cell is run with three seeds (1566911444, 20241017, 20251017), and every reported number is the mean and population standard deviation over the three. Model selection and early stopping use the reported validation metric (Micro-F1 for node classification, MRR for link prediction), and the augmentation baseline selects on the mean over variants; its budget is the single-run budget divided by the number of variants, with the patience scaled in the same way. In link prediction, every held-out pair is ranked against a fixed candidate set that is identical across variants, methods, and models (Section H.4). Following the convention of CMPNN, models that score a query with a query-dependent encoder (CMPNN, MAGNN, and SlotGAT on IMDb and DBLP) remove the training query's edge, together with any edge that is a copy of it in the variant (e.g., the movie's link node on IMDb, the paper's area edge on DBLP), from the message graph for the positive and every negative candidate, so that a training query is scored the way a held-out one is; RGCN's link prediction encoder is transductive and uses the full training graph. For PAIN's independently trained node-classification arms, each epoch makes one full-graph update using cross-entropy on the labeled training nodes. The checkpoint is selected by validation Macro-F1, with validation loss breaking ties; early stopping uses the same metric. IMDb augmentation trains one shared model and optimizer, visits the four variants in seeded random order once per super-epoch, and selects on mean validation Macro-F1 across variants. Its limit of 250 super-epochs corresponds to at most 1,000 optimizer updates, with early-stopping patience of 20 super-epochs. The original, universal-graph, and invariant PAIN runs use the same optimization settings within each dataset.

Hyper-parameters. Tables 6–10 list the architecture, optimization, and stopping settings of every model, one table per dataset and task; they are the same for the three baselines and our approach.

Table 7: Hyper-parameters for IMDb link prediction (Movie–Director; 19 negatives per pair). Settings are identical for the original, universal-graph, and augmentation baselines and for our invariant approach. "lr" is the learning rate and "wd" the weight decay; the optimizer is Adam throughout.

Model	Architecture	Optimization	Selection / stopping
RGCN	3 layers; input 128, hidden/output 256; 16 bases; dropout 0.1; mean normalization; dot-product decoder; embedding regularization 10^{-6}	lr 2×10^{-3}, wd 10^{-5}; full batch; pairwise loss over the 19 negatives; ≤ 1000 epochs	val. MRR; patience 200
MAGNN	movie-side and target-side metapath groups; 3 averaged layers; hidden 64; 8 heads; attention vector 128; RotatE0; dropout 0.5; dot-product score	lr 5×10^{-3}, wd 10^{-3}; batch 64 positives $\times$ 19 negatives; 100 sampled instances; negative log-sigmoid loss; ≤ 100 epochs	val. MRR; patience 10
CMPNN	6 layers; input/hidden 32; DistMult messages; PNA aggregation; layer normalization; short-cut connections; no node features	lr 10^{-3}; batch 8 queries $\times$ 20 candidates; negative log-sigmoid loss; ≤ 20 epochs	val. MRR; patience 5
SlotGAT	2 hidden layers $\times$ 4 heads + 1 output layer (1 head); hidden 128; per-type input projections; concatenated-layer + LayerNorm readout; diagonal DistMult decoder	lr 10^{-3}, wd 0; batch 512 positives $\times$ 19 negatives; negative log-sigmoid loss; ≤ 400 epochs	val. MRR; patience 20

Table 8: Hyper-parameters for DBLP link prediction (Paper–Venue; ranked over all 20 venues; featureless: learned embeddings / identity inputs). Settings are identical for the original, universal-graph, and augmentation baselines and for our invariant approach. "lr" is the learning rate and "wd" the weight decay; the optimizer is Adam throughout.

Model	Architecture	Optimization	Selection / stopping
RGCN	3 layers; input 128, hidden/output 256; 16 bases; dropout 0.1; mean normalization; learned node embeddings; dot-product decoder; embedding regularization 10^{-6}	lr 2×10^{-3}, wd 10^{-5}; full batch; pairwise loss over the 19 negatives; ≤ 1000 epochs	val. MRR; patience 200
MAGNN	paper-side and target-side metapath groups; 3 averaged layers; hidden 64; 8 heads; attention vector 128; RotatE0; dropout 0.5; sparse identity features per type; dot-product score	lr 5×10^{-3}, wd 10^{-3}; batch 64 positives $\times$ 19 negatives; 100 sampled instances; negative log-sigmoid loss; ≤ 100 epochs	val. MRR; patience 10
CMPNN	6 layers; input/hidden 32; DistMult messages; PNA aggregation; layer normalization; short-cut connections; no node features	lr 10^{-3}; batch 8 queries $\times$ 20 candidates; negative log-sigmoid loss; ≤ 20 epochs	val. MRR; patience 5
SlotGAT	2 hidden layers $\times$ 4 heads + 1 output layer (1 head); hidden 128; per-node embedding input; concatenated-layer + LayerNorm readout; diagonal DistMult decoder	lr 10^{-3}, wd 0; batch 512 positives $\times$ 19 negatives (the other venues); negative log-sigmoid loss; ≤ 400 epochs	val. MRR; patience 20

Hardware and software. Unless stated otherwise, experiments were run on a single NVIDIA V100-SXM3 GPU with 32 GB of memory per run, with 4 CPU cores (8 for RGCN on Freebase and WN18RR) and 16–128 GB of host memory depending on the preprocessing; PAIN on IMDb and all PathNet runs used the same GPU, while PAIN on Freebase used a single NVIDIA H100. RGCN, MAGNN, SlotGAT, and SeHGNN use Python 3.11, PyTorch 2.4.0 (CUDA 12.4), and DGL 2.4.0; CMPNN uses PyTorch 2.0.1 (CUDA 11.8) with torchdrug 0.2.1 and torch_scatter 2.1.2; PAIN uses Python 3.11.15 and PyTorch 2.1.3 (CUDA 12.6) without PyTorch Geometric or DGL; and PathNet uses Python 3.11 with PyTorch 1.11.0 (CUDA 10.2) and PyTorch Geometric 2.0.4, without DGL. All runs use deterministic kernels (`torch.use_deterministic_algorithms`, a fixed cuBLAS workspace (`CUBLAS_WORKSPACE_CONFIG=:4096:8`), seeded generators, sorted edge insertion, and segment rather than atomic reductions in the invariant RGCN and CMPNN runners), which is what allows the invariance of our approach to be verified exactly: across the variants of a seed, the invariant models have identical per-epoch parameter hashes and identical final checkpoints, rather than merely similar outputs. Determinism of this kind is tied to the GPU architecture, so the bitwise checks for a given model are all run on one GPU type.

H.3 Invariant Structural Biases per Architecture

The algorithm of Section 4.3 takes the structural bias $\mathcal{S}^f$ of a GNN f and the mapping τ and returns the invariant structural bias $\mathcal{S}^f_\tau$, a set of conditional graphs. This section describes how $\mathcal{S}^f_\tau$ is materialized for each architecture, since different architectures consume their structural bias in different forms: as a typed edge list (RGCN, CMPNN, SlotGAT), as a set of metapaths (MAGNN),

Table 9: Hyper-parameters for WN18RR link prediction (filtered ranking over all entities; featureless). Settings are identical for the original, universal-graph, and augmentation baselines and for our invariant approach. "lr" is the learning rate and "wd" the weight decay; the optimizer is Adam throughout.

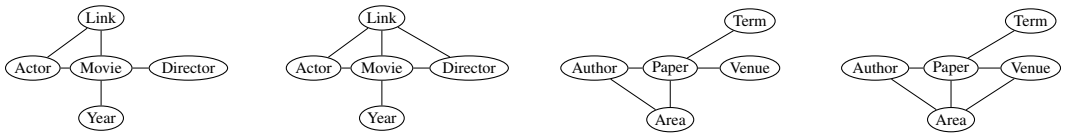
Model	Architecture	Optimization	Selection / stopping
RGCN	hidden 200; 30 bases; edge dropout 0.4 (0.2 on self-loops); relation-specific decoder over the 11 base relations; full batch on 200k-edge chunks	lr 10^{-2}, wd 10^{-2}; 1 negative per positive; $\leq$ 8000 epochs, evaluated every 10	val. filtered MRR; patience 500
CMPNN	6 layers; input/hidden 32; DistMult messages; sum aggregation; no node features	lr 5×10^{-3}; batch 64 queries; 32 negatives per positive; $\leq$ 10 epochs	val. filtered MRR; patience 3
SlotGAT	single slot (one node type); 2 hidden layers $\times$ 4 heads + 1 output layer (1 head); hidden 128; concatenated-layer + LayerNorm readout; diagonal DistMult over the 11 base relations	lr 10^{-3}, wd 0; batch 2,048 positives $\times$ 256 uniform negatives; negative log-sigmoid loss; $\leq$ 400 epochs	val. filtered MRR every 5 epochs; patience 8 evaluations

Table 10: Hyper-parameters for Freebase node classification (featureless: learned per-type embeddings). Settings are identical for the original, universal-graph, and augmentation baselines and for our invariant approach. "lr" is the learning rate and "wd" the weight decay; the optimizer is Adam throughout.

Model	Architecture	Optimization	Selection / stopping
RGCN	2 layers with neighbor sampling (fan-out 4 + 4); embedding/hidden 128; 8 bases; dropout 0.3	lr 2×10^{-3}, wd 10^{-5}; batch 64; $\leq$ 60 epochs	val. Micro-F1; patience 20
MAGNN	hidden 64; 8 heads; attention vector 128; RotatE0; dropout 0.5; sparse identity features per type	lr 5×10^{-3}, wd 10^{-3}; mini-batches of 8 books; 100 sampled instances per (book, metapath); $\leq$ 100 epochs	val. Micro-F1; patience 5
SlotGAT	2 hidden layers + 1 output layer; hidden 64; 8 heads; slot attention (dim. 3); edge-type features 64; L2 readout; projected residual connections	lr 5×10^{-3}, wd 10^{-3}; full graph; $\leq$ 150 epochs	select on val. Micro-F1, stop on val. loss; patience 25
SeHGNN	relation-aware channels of length $\leq$ 2 over learned per-type embeddings, 500 sampled walks per (target, channel); embedding/hidden 256; 2 projection layers; 4 task layers; dropout 0.5	lr 5×10^{-3}, wd 10^{-4}; batch 1,024; $\leq$ 100 epochs	val. Micro-F1; patience 30
PAIN	5 layers; hidden 128; sampled rooted paths of length 0–3; shared 2-layer LSTM; sum aggregation; learned node-ID and node-type embeddings; 4 neighbours/relation, fanout 8, 256 paths/root; dropout 0	lr 10^{-3}, wd 0; full graph; cross-entropy; gradient-norm clip 5; lr halved after 10 epochs without validation improvement (min 10^{-5}); $\leq$ 300 epochs or 48 h	val. Macro-F1, val. loss breaking ties; patience 30
PathNet	hetero PathNetNC; hidden 64; 40 MERW walks $\times$ length 4; structure-only head; dropout 0.5; learned 64-d node embeddings (featureless)	lr 10^{-2}, wd 10^{-3}; full batch; cross-entropy; $\leq$ 200 epochs; overfit stop if train Acc $\geq$ 0.999 for 15 epochs w/o val. gain	val. Macro-F1; patience 35

or as a set of pre-computed feature channels (SeHGNN). In every case the architecture, loss, and training protocol are unchanged (Section H.2); only this structural input differs between the original GNN and its invariant version. Table 11 summarizes the instantiations.

Common construction. For every variant, a preprocessing step finds the occurrences of the context subgraph of each conditional graph in $\mathcal{S}_\tau^f$ and emits the corresponding propagation edges, each labelled by the *semantic relation* it realizes (e.g., movie–director, whether the variant encodes it as a direct edge or through the link node). We call the resulting labelled edge multiset the *compiled* structural bias of the variant. Three rules make the compiled bias identical across variants: (i) nodes that belong to a context subgraph but not to its propagation subgraph (*skip nodes*, e.g., the link node on IMDb) license a propagation edge but contribute no features; (ii) several occurrences that realize the same propagation edge are collapsed into one (set semantics), so that a relation encoded by two paths in one variant and by one edge in another produces the same messages; (iii) wherever an architecture samples or caps occurrences, the sample is a deterministic function of the semantic identity of the edge rather than of the variant's layout. Under these rules, the compiled bias of every variant is identical (we verify this by hashing), and since supervision, initialization, and reduction order are also shared, the training trajectories of the invariant models are identical across variants. This is why the invariant rows of the result tables reach a Kendall-τ of exactly 1: the per-variant models have bitwise identical parameters, not merely similar outputs. The universal graphs of Figure 4 are



(a) $\bigcup_{i\in\{1,3\}}$ IMDb$_i$ **(b)** $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$ **(c)** $\bigcup_{i\in\{1,3\}}$ DBLP$_i$ **(d)** $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$

Figure 4: Universal graphs for IMDb and DBLP.

related to the compiled bias as follows. On DBLP, the compiled bias contains exactly the edges of the universal graph, so the invariant model and the universal-graph model compute the same function (their rows in the result tables coincide); the contribution of the invariant version is that each variant alone recovers it. Elsewhere the two differ: on IMDb node classification and Movie–Director the compiled bias contains propagation edges that no variant and no union materializes (link–year and link–director); on WN18RR and Freebase the compiled relations are labelled by the composition they realize rather than by the materialized shortcut; and for MAGNN and SeHGNN the invariant bias is a set of metapaths or channels, not a graph.

Reading the dependency, or supplying it. The compiler reads the value a switching carries out of the variant's own edges whenever the dependency that licenses the switching survives in that variant. On IMDb and on DBLP1 and DBLP3 it does, so nothing further is needed. DBLP2 is the one variant in our datasets where it does not: the area sits on the venue, which carries the areas of all of its papers, so the context Paper $\rightarrow$ Venue $\rightarrow$ Area offers 3.96 candidate areas and does not say which is the paper's own (Section H.1). There the compiler is given the Paper $\rightarrow$ Area dependency explicitly, as a table over the training papers listing each paper's single area, derived from its authors' labels; we call this table the paper–area *certificate*. It is used only to choose which candidate is the true one and is never added to the variant's edges or to the compiled structural bias, and it is built from training papers alone. The DBLP2 results are therefore invariant *given* this dependency, which is a stronger requirement than the other variants make.

RGCN (Invariant). RGCN's structural bias is one rooted edge per relation type. Its invariant version runs the unchanged RGCN layer on the compiled semantic relations, with the relation-specific weights indexed by the semantic relation name, so the same weight is used for a relation regardless of how a variant encodes it. On IMDb, the compiled bias contains, for every attribute $X \in \{\text{director}, \text{actor}, \text{year}\}$, both the movie–$X$ and the link–X relation on every variant, whichever of the two the variant materializes (the other is obtained through the link node as context). On DBLP, the paper–area relation is reconstructed on every variant through the variant's own encoding (directly, through the venue, or through the authors), restricted to the area that belongs to the paper's own block so that a venue shared by papers of several areas does not over-generate. On WN18RR, the semantic relations are the 11 base relations, their inverses, and the 2,065 mined compositions; on each variant a composition is read from the materialized shortcut edge when the variant has it and composed from the base relations otherwise (inverse steps are traversed as transposes where the variant has no inverse edges), and a materialized shortcut is checked against its composition. On Freebase, the semantic relations are the 36 base relations and the 28 two- and three-hop compositions through books that pass the selection criterion of Section H.1, i.e. exactly the relation set of the universal graph (64 semantic relations, 128 directed channels). They are evaluated per source node one block at a time and sampled with a deterministic fan-out of $4 + 4$ neighbors per layer, so that the 987 million compiled edges are never materialized. Table 12 gives the sizes.

CMPNN (Invariant). CMPNN propagates along relational paths from the query source, so its structural bias is the set of typed edges it can traverse; identical labelled edge multisets therefore give identical path sets for every source. Its invariant version consumes the same compiled semantic relations as RGCN (the forward direction only, since CMPNN adds inverse edges itself), with the training query's edge and its copies masked for every candidate as described in Section H.2. Two details matter for exact invariance: duplicate edges are removed because CMPNN's PNA aggregation is sensitive to message counts, and messages are reduced with deterministic segment sums instead of atomic adds. On WN18RR, the inverse and shortcut relations enter only as context: CMPNN's six propagation layers already compose the base relations along the paths that the shortcuts summarize, and a shortcut edge that was mined from the training triples would re-introduce the masked training query. Its propagation edges are therefore the 11 base relations (86,835 edges) on every variant.

SlotGAT (Invariant). SlotGAT's structural bias is the set of typed one-hop edges, with each message written into the slot of the neighbor's type and the edge type entering the attention. Its invariant version receives a *message plan*: the compiled propagation edges, typed by the semantic relation, so that the attention of every node normalizes over the same multiset of (slot, message) pairs on every variant. On IMDb, direct movie–X contexts and two-hop movie–link–X contexts are mapped to the same semantic message, giving 15 message types and 107,713 messages per variant. On Freebase, the plan reproduces, for the 129 channels of the relation family, one message per (destination, channel, source) record of the universal graph without materializing the dense compositions (750.5 million messages). Because the published SlotGAT layer uses non-deterministic scatter operations, the message plan runs on our deterministic port of the same architecture (the same computation to floating-point precision and the same parameter count; segment softmax instead of atomic adds). Every reported SlotGAT row uses the port, on IMDb node classification as well as on Freebase and the link prediction tasks.

MAGNN (Invariant + Intersect, Invariant + Union). MAGNN's structural bias is its set of metapaths. A switching changes which node-type sequence realizes a semantic relation (movie–director–movie on IMDb1 becomes movie–link–director–link–movie on IMDb2), so the original metapath sets differ per variant (Tables 14 and 15). The invariant version uses *skip metapaths*: a metapath is matched on the variant's graph including the intermediate nodes, but the skip nodes, written (X), are dropped from the stored instance, so the encoded instance (endpoints and non-skip intermediates) is the same on every variant. We report two invariant metapath sets. *Intersect* contains one skip metapath per semantic relation incident to the target node type, i.e., the metapaths of the variant in which every relation is a direct edge, realized on every other variant through skip nodes; this is the smallest set produced by the algorithm. *Union* adds the long forms whose interior nodes exist on every variant (e.g., M-L-D-L-M, which passes through the link node on every IMDb variant), i.e., the metapaths that appear in some variant's original set and are informative on all of them. On DBLP, Union is the set that matters for accuracy: the query paper's own area edges are masked, so the area reaches the query only through P-A-Ar-A-P. Instances are enumerated with set semantics and capped at 500 per (source node, metapath) with a cap keyed on the semantic identity; the resulting instance tables are byte-identical across variants. On Freebase, the semantic relations are the 26 relations incident to books in the relation family (8 base and 18 shortcut compositions); Intersect uses the 26 skip metapaths Book $\xrightarrow{r}$ X $\xrightarrow{r^{-1}}$ Book (a composition is traversed through its interior nodes as skip nodes on a variant that does not materialize it), and Union adds the 18 long forms with real interior nodes, for 44 metapaths. The original sets have 8, 12, 22, and 26 metapaths on Freebase1, Freebase2, Freebase3, and the universal graph, respectively.

SeHGNN (Invariant, Invariant + relation-aware). SeHGNN's structural bias is the set of pre-computed feature channels, one per type path of length at most K from the target type. Its invariant version applies the unchanged channel operator (a product of row-normalized adjacency matrices) to the compiled semantic relations instead of the physical ones, so each channel is a deterministic function of the compiled bias and is bit-identical across variants; skip types contribute connectivity but no features. Because SeHGNN parameterizes each channel separately, channels that are τ-equivalent but not isomorphic (a path and its image through the link node) would otherwise be computed twice; the invariant channel set therefore contains one representative per class: on IMDb, the type paths over the schema $\{\text{movie}, \text{director}, \text{actor}, \text{year}\}$ realized on the semantic relations, with the link type excluded, giving 25 channels plus 13 label-propagation channels on every variant, against 41/27/32/32 (+21/9/14/14 label) channels on IMDb1–4 and 101 (+31) on the universal graph. On IMDb every pair of node types carries exactly one relation, so a channel per type path and a channel per relation-typed path coincide and a single SeHGNN version is reported. On Freebase, up to nine relations connect the same pair of node types (e.g., book and music), and the published SeHGNN merges them into one step; we therefore report both the published, type-collapsed channels (73 channels on every variant, the universal graph, and the invariant version) and *relation-aware* channels in which every semantic relation is its own step (rows marked "+ relation-aware": 91/159/443 channels on Freebase1–3, 597 on the universal graph, and 511 for the invariant version, namely the channels that occur in at least one variant's own channel set, i.e., the union of the variants' structural biases mapped through the switchings; the 86 remaining compositions of the universal graph occur in no variant). Channels on Freebase are estimated with 500 sampled walks per (target, channel), seeded by the semantic identity of the channel. Table 13 lists the channel counts.

Table 11: How the invariant structural bias $\mathcal{S}_\tau^f$ is materialized for each architecture. In every case the architecture, loss, and training protocol are unchanged; only the structural input differs.

Architecture	Structural bias	Invariant version (as reported)
RGCN	typed one-hop edges (relation-specific weights)	compiled semantic relations; weights indexed by semantic relation
CMPNN	relational paths from the query source	compiled semantic relations (forward direction), duplicates removed, deterministic reductions; base relations only on WN18RR
SlotGAT	typed one-hop edges into per-type slots	message plan of compiled propagation edges typed by semantic relation, on a deterministic port of the layer
MAGNN	set of metapaths	skip metapaths: Intersect (one per semantic relation) and Union (adds long forms with real interior nodes)
SeHGNN	pre-computed channels, one per type path	channels computed on the compiled semantic relations with skip types excluded; on Freebase also relation-aware channels restricted to those occurring in some variant
PAIN	rooted simple paths of bounded length	paths enumerated on the compiled semantic relations; exact on IMDb, deterministically sampled by semantic identity on Freebase, with skip nodes as context only
PathNet	sampled rooted walks around the target node	walks compiled to semantic steps, with a skip-node gather in front of the unchanged PathNetNC encoder

Table 12: Compiled invariant structural bias consumed by RGCN, CMPNN, and SlotGAT: relation types and directed edges per variant of the original graphs, and of the compiled bias, which is identical on every variant.

	Original (per variant)		Compiled (every variant)	
Dataset / task	relations	edges	relations	edges
IMDb, NC	8	50,142	14	91,924
IMDb, Movie–Director	8	47,634	14	86,908
DBLP, Paper–Venue	8	268,490–287,236	12	294,334
WN18RR	11 / 22 / 537	86,835 / 173,670 / 706,128	2,087	893,629
Freebase	36 / 42 / 58	1.06 M / 47.4 M / 447.0 M	64	986.6 M

PAIN (Invariant). PAIN's structural bias is the multiset of rooted simple paths of length at most three, each encoded by a shared LSTM and summed at its root. Its invariant version leaves the PAIN layers and the path encoder unchanged and alters only what the paths are enumerated on: each variant's conditional graphs are compiled to semantic relations first, and the paths are enumerated over those, with skip nodes licensing a relation but contributing no features. On IMDb node classification the movie–link edge pairs each movie with its link node, so an attribute edge attached to either endpoint licenses both the movie–X and the link–X semantic relation on every variant; the compiled bias therefore coincides with the universal graph even though it is compiled from each variant separately, and all rooted simple paths of length zero to three are enumerated exactly. On Freebase the semantic relations are the 36 base relations and the 28 compositions of Section H.1, a composition being reconstructed from its base-relation sequence on a variant that does not materialize its shortcut. Endpoints are evaluated one source block at a time so that the dense semantic graph is never materialized, and since exhaustive enumeration is infeasible the paths are sampled deterministically, keyed on the semantic identity of the relation rather than on the variant's layout: at most four neighbours per source and relation, fan-out eight, and 256 paths per root.

PathNet (Invariant).

Table 13: SeHGNN feature channels, with label-propagation channels in parentheses. Original counts are listed per variant (IMDb1–4, Freebase1–3).

Dataset	Channels	Original	Universal Graph	Invariant
IMDb	per type path, $K \leq 4$	41 (21) / 27 (9) / 32 (14) / 32 (14)	101 (31)	25 (13)
Freebase	type-collapsed, $K \leq 2$	73 / 73 / 73	73	73
Freebase	relation-aware, $K \leq 2$	91 / 159 / 443	597	511

Table 14: MAGNN metapaths on IMDb, written as node-type sequences (M: movie, D: director, A: actor, L: link, Y: year); a parenthesized node (X) is a skip node, traversed but excluded from the encoded instance. The sets shown are the movie-side sets, used for node classification; Movie–Director (IMDb1 and IMDb3 only) adds the director-side metapaths DMD and, in the invariant sets, D(M)L(M)D.

Variant	**Original**	**Invariant + Intersect**	**Invariant + Union**
IMDb1	MDM, MAM, MLM, MYM	MDM, MAM, MLM, MYM	MDM, MAM, MLM, MYM, ML(M)D(M)LM, ML(M)A(M)LM, ML(M)Y(M)LM
IMDb2	MLDLM, MLALM, MLM, MYM	M(L)D(L)M, M(L)A(L)M, MLM, MYM	M(L)D(L)M, M(L)A(L)M, MLM, MYM, MLDLM, MLALM, ML(M)Y(M)LM
IMDb3	MDM, MLALM, MLM, MYM	MDM, M(L)A(L)M, MLM, MYM	MDM, M(L)A(L)M, MLM, MYM, ML(M)D(M)LM, MLALM, ML(M)Y(M)LM
IMDb4	MLDLM, MAM, MLM, MYM	M(L)D(L)M, MAM, MLM, MYM	M(L)D(L)M, MAM, MLM, MYM, MLDLM, ML(M)A(M)LM, ML(M)Y(M)LM
$\bigcup_i$ IMDb$_i$	MDM, MAM, MLM, MYM, MLDLM, MLALM	–	–

Table 15: MAGNN metapaths on DBLP for Paper–Venue link prediction (P: paper, A: author, T: term, V: venue, Ar: area; (X) is a skip node). The paper-side metapaths are listed; every set also contains the venue-side metapaths VPV and VPAPV.

Variant	**Original**	**Invariant + Intersect**	**Invariant + Union**
DBLP1	PAP, PTP, PVP, PArP	PAP, PTP, PVP, PArP	PAP, PTP, PVP, PArP, PA(P)Ar(P)AP, PV(P)Ar(P)VP
DBLP2	PAP, PTP, PVP, PVArVP	PAP, PTP, PVP, P(V)Ar(V)P	PAP, PTP, PVP, P(V)Ar(V)P, PA(P)(V)Ar(V)(P)AP, PVArVP
DBLP3	PAP, PTP, PVP, PAArAP	PAP, PTP, PVP, P(A)Ar(A)P	PAP, PTP, PVP, P(A)Ar(A)P, PAArAP, PV(P)(A)Ar(A)(P)VP
$\bigcup_i$ DBLP$_i$	PAP, PTP, PVP, PArP, PAArAP, PVArVP	–	–

H.4 METRICS

Every number in the result tables is computed from the raw test outputs of the runs (class logits for node classification, candidate-score matrices for link prediction), with one definition per metric shared by all models. We report effectiveness, degree of invariance, and efficiency.

Setup. Let $\mathcal{V} = \{G^{(1)}, \ldots, G^{(m)}\}$ be the variants of a dataset (IMDb1–4, DBLP1–3, WN18RR1–3, Freebase1–3) and $R = 3$ the number of seeds. We write $s^{(v,r)}(x, c)$ for the score that the model trained on variant v with seed r assigns to output c for instance x. For node classification, $\mathcal{X}$ is the set of labelled test nodes, identical in every variant, with C classes. For link prediction, $\mathcal{Q}$ is the set of test queries $(h, r, ?)$, each with a true answer t_q and a candidate set $\mathcal{C}(q)$ that is *identical across variants, models, and methods*: on IMDb the true tail and 19 uniformly sampled directors; on DBLP the true venue and the other 19 venues; on WN18RR every entity, filtered by the other known true answers, for the ranking metrics, and the true entity plus 50 fixed negatives for the classification and invariance metrics. Every WN18RR test triple yields a tail-corrupted and a head-corrupted query.

Effectiveness. For node classification we report Accuracy, macro-averaged Precision and Recall, Macro-F1, and Micro-F1 in their standard form over the predictions $\hat{y}_x = \arg\max_c s(x, c)$, with

$$\text{Macro-F1} = \frac{1}{C}\sum_{c=1}^{C} \frac{2\,\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}, \qquad \text{Micro-F1} = \frac{2P_\mu R_\mu}{P_\mu + R_\mu},$$

where P_μ and R_μ pool true and false positives over classes. The task is single-label, so Micro-F1 equals Accuracy and the two columns coincide.

For link prediction we report Hits@1, Hits@3, and MRR, computed from the rank of the true answer among its candidates,

$$r_q = 1 + \big|\{c \in \mathcal{C}(q) : s(q, c) > s(q, t_q)\}\big| + \tfrac{1}{2}\big|\{c \in \mathcal{C}(q) \setminus \{t_q\} : s(q, c) = s(q, t_q)\}\big|,$$

so that ties receive their expected rank under a random tie-break rather than a best or a worst case; r_q may be fractional. On WN18RR, Hits@k and MRR instead use the standard filtered ranking over all entities, in which a tie counts against the true entity. For reference, random scores over 20 distinct candidates give MRR $= 0.180$, Hits@1 $= 0.050$, and Hits@3 $= 0.150$.

We also report Precision, Recall, and F1, treating every (query, candidate) pair as a binary instance with label $\mathbf{1}[c = t_q]$ and prediction $\mathbf{1}[\sigma(s(q, c)) \geq 0.5]$ and pooling over all pairs. Unlike the ranking metrics, these depend on where a model's scores fall relative to the threshold, which the training objectives do not calibrate; they are therefore comparable across the methods of one model but not across models. Where a model's scores concentrate on one side of the threshold, the three columns saturate and the ranking metrics are the informative ones.

Degree of invariance. For each pair of variants $u < v$ and each seed, we compare the outputs of the two models on the same test instances and the same candidate sets. The per-instance agreement is Kendall's τ_b between the two score vectors, which accounts for ties; two identical vectors count as 1 even where τ_b is undefined. The tables report

$$\text{Kendall-}\tau^{\,u,v} = \frac{1}{R}\sum_{r=1}^{R} \frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X}} \tau_x^{u,v,r}$$

for every pair of variants, as mean $\pm$ standard deviation over seeds, and the main text reports the mean over pairs. A value of 1 means that the two models rank the outputs of every test instance identically. Because the per-instance τ_b over $C = 3$ classes takes only the values $\{-1, -\frac{1}{3}, \frac{1}{3}, 1\}$, the IMDb node classification values are coarser than the link prediction ones.

For link prediction we additionally ask whether the two models succeed at top-k on the *same* queries: Kendall-τ@k is τ_b between the two models' vectors of per-query Hits@k indicators, $k \in \{1, 3\}$. It equals 1 exactly when the two vectors are identical and is undefined, and reported as missing, when either is constant. Because it correlates sparse indicator vectors its value is coupled to the Hits@k base rate, so for models near chance it typically falls below the full-ranking Kendall-τ.

Not every method has invariance rows. The original and invariant methods train one model per variant, so their rows compare two models; the augmentation method trains one model per seed, and

Table 16: RGCN: node classification results (IMDb, Freebase). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Accuracy	Precision	Recall	Micro-F1	Macro-F1
IMDb	Original	IMDb1	0.6515 ± 0.0171	0.6554 ± 0.0169	0.6507 ± 0.0182	0.6515 ± 0.0171	0.6518 ± 0.0175
		IMDb2	0.6033 ± 0.0030	0.6089 ± 0.0064	0.6000 ± 0.0022	0.6033 ± 0.0030	0.6018 ± 0.0028
		IMDb3	0.6491 ± 0.0157	0.6578 ± 0.0123	0.6465 ± 0.0187	0.6491 ± 0.0157	0.6492 ± 0.0171
		IMDb4	0.6180 ± 0.0089	0.6225 ± 0.0103	0.6156 ± 0.0102	0.6180 ± 0.0089	0.6164 ± 0.0091
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.6535 ± 0.0089	0.6595 ± 0.0111	0.6522 ± 0.0080	0.6535 ± 0.0089	0.6541 ± 0.0089
	Augmentation	IMDb1	0.6136 ± 0.0186	0.6221 ± 0.0180	0.6092 ± 0.0203	0.6136 ± 0.0186	0.6123 ± 0.0204
		IMDb2	0.5981 ± 0.0134	0.6051 ± 0.0109	0.5940 ± 0.0158	0.5981 ± 0.0134	0.5963 ± 0.0155
		IMDb3	0.6077 ± 0.0161	0.6141 ± 0.0144	0.6046 ± 0.0185	0.6077 ± 0.0161	0.6069 ± 0.0181
		IMDb4	0.5993 ± 0.0164	0.6080 ± 0.0156	0.5936 ± 0.0178	0.5993 ± 0.0164	0.5964 ± 0.0181
	Invariant	IMDb1	0.6531 ± 0.0107	0.6613 ± 0.0084	0.6512 ± 0.0120	0.6531 ± 0.0107	0.6539 ± 0.0107
		IMDb2	0.6531 ± 0.0107	0.6613 ± 0.0084	0.6512 ± 0.0120	0.6531 ± 0.0107	0.6539 ± 0.0107
		IMDb3	0.6531 ± 0.0107	0.6613 ± 0.0084	0.6512 ± 0.0120	0.6531 ± 0.0107	0.6539 ± 0.0107
		IMDb4	0.6531 ± 0.0107	0.6613 ± 0.0084	0.6512 ± 0.0120	0.6531 ± 0.0107	0.6539 ± 0.0107
Freebase	Original	Freebase1	0.6824 ± 0.0053	0.5838 ± 0.0102	0.5046 ± 0.0292	0.6824 ± 0.0053	0.5296 ± 0.0252
		Freebase2	0.6857 ± 0.0095	0.5889 ± 0.0269	0.5012 ± 0.0074	0.6857 ± 0.0095	0.5303 ± 0.0130
		Freebase3	0.6885 ± 0.0038	0.5805 ± 0.0098	0.5092 ± 0.0113	0.6885 ± 0.0038	0.5313 ± 0.0088
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	0.6864 ± 0.0031	0.6100 ± 0.0091	0.5064 ± 0.0033	0.6864 ± 0.0031	0.5381 ± 0.0034
	Augmentation	Freebase1	0.6906 ± 0.0039	0.6239 ± 0.0153	0.5134 ± 0.0105	0.6906 ± 0.0039	0.5463 ± 0.0126
		Freebase2	0.6897 ± 0.0057	0.6132 ± 0.0142	0.5135 ± 0.0104	0.6897 ± 0.0057	0.5453 ± 0.0137
		Freebase3	0.6916 ± 0.0033	0.6182 ± 0.0067	0.5136 ± 0.0093	0.6916 ± 0.0033	0.5469 ± 0.0101
	Invariant	Freebase1	0.6895 ± 0.0047	0.6198 ± 0.0210	0.5041 ± 0.0152	0.6895 ± 0.0047	0.5358 ± 0.0153
		Freebase2	0.6895 ± 0.0047	0.6198 ± 0.0210	0.5041 ± 0.0152	0.6895 ± 0.0047	0.5358 ± 0.0153
		Freebase3	0.6895 ± 0.0047	0.6198 ± 0.0210	0.5041 ± 0.0152	0.6895 ± 0.0047	0.5358 ± 0.0153

its rows compare that model's outputs when it is given variant u against variant v; the universal-graph method trains a single model on a single graph, so its invariance is trivially 1 and no row is shown. For the invariant method the value of 1 is exact rather than approximate, since the models trained on different variants have identical parameters (Section H.3).

Efficiency. All four quantities are taken from each run's own record. *Training time* is the wall-clock time of the training loop, including validation passes and checkpointing and excluding preprocessing and test inference (the main text reports it in minutes); *epochs* is the number of epochs trained until early stopping, where one super-epoch of the augmentation method counts as one epoch per variant; *parameters* is the size of all trainable tensors in MiB at their stored precision; and *peak GPU memory* is the maximum allocated on the GPU (`torch.cuda.max_memory_allocated`) over the training loop and, separately, over test inference. All runs used the same GPU model (Section H.2), so times are comparable across methods within a model; across models they also reflect different implementations and frameworks.

Reporting. Every cell is the mean ± population standard deviation over the three seeds. Effectiveness tables have one row per variant for the original, augmentation, and invariant methods (the augmentation rows evaluate the one shared model on each variant's graph) and one row for the universal graph; efficiency tables have one row per variant for the original and invariant methods and one row each for the augmentation and universal-graph methods; invariance tables have one row per pair of variants. Main-text values are the means of the corresponding variant rows, or of the pair rows for invariance. The spread of a metric M is $\Delta M = \max_v \bar{M}_v - \min_v \bar{M}_v$, where $\bar{M}_v$ is the seed-mean of the row of variant v.

H.5 Extended Results & Analysis

Table 17: RGCN: node classification degree of invariance (IMDb, Freebase). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ
IMDb	Original	IMDb1 vs. IMDb2	0.7350 ± 0.0779
		IMDb1 vs. IMDb3	0.8740 ± 0.0715
		IMDb1 vs. IMDb4	0.8022 ± 0.0746
		IMDb2 vs. IMDb3	0.7344 ± 0.0730
		IMDb2 vs. IMDb4	0.8227 ± 0.0674
		IMDb3 vs. IMDb4	0.7379 ± 0.0261
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	IMDb1 vs. IMDb2	1.000 ± 0.0000 (trivial)
		IMDb1 vs. IMDb3	
		IMDb1 vs. IMDb4	
		IMDb2 vs. IMDb3	
		IMDb2 vs. IMDb4	
		IMDb3 vs. IMDb4	
	Augmentation	IMDb1 vs. IMDb2	0.9282 ± 0.0157
		IMDb1 vs. IMDb3	0.9623 ± 0.0125
		IMDb1 vs. IMDb4	0.9418 ± 0.0273
		IMDb2 vs. IMDb3	0.9452 ± 0.0268
		IMDb2 vs. IMDb4	0.9641 ± 0.0143
		IMDb3 vs. IMDb4	0.9264 ± 0.0215
	Invariant	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
Freebase	Original	Freebase1 vs. Freebase2	0.7135 ± 0.0100
		Freebase1 vs. Freebase3	0.7187 ± 0.0154
		Freebase2 vs. Freebase3	0.7381 ± 0.0343
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Augmentation	Freebase1 vs. Freebase2	0.9865 ± 0.0022
		Freebase1 vs. Freebase3	0.9793 ± 0.0008
		Freebase2 vs. Freebase3	0.9797 ± 0.0041
	Invariant	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000

Table 18: RGCN: node classification efficiency (IMDb, Freebase). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb	Original	IMDb1	40.90 ± 0.52	62.00 ± 0.82	2.60 ± 0.00	638.11 ± 0.00	490.90 ± 0.00
		IMDb2	52.23 ± 14.17	79.33 ± 21.70	2.60 ± 0.00	637.54 ± 0.00	490.72 ± 0.00
		IMDb3	40.50 ± 0.76	61.33 ± 1.25	2.60 ± 0.00	637.61 ± 0.00	491.23 ± 0.15
		IMDb4	43.36 ± 2.03	65.67 ± 3.09	2.60 ± 0.00	638.38 ± 0.19	491.73 ± 0.59
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	42.04 ± 1.23	61.33 ± 1.89	2.60 ± 0.00	714.75 ± 0.00	492.08 ± 0.00
	Augmentation	IMDb1-4	14.09 ± 1.03	65.33 ± 4.99	2.60 ± 0.00	1187.33 ± 0.00	1041.91 ± 0.00
	Invariant	IMDb1	42.28 ± 0.29	60.33 ± 0.47	2.60 ± 0.00	734.85 ± 0.00	490.74 ± 0.00
		IMDb2	42.08 ± 0.29	60.33 ± 0.47	2.60 ± 0.00	734.85 ± 0.00	490.74 ± 0.00
		IMDb3	42.03 ± 0.28	60.33 ± 0.47	2.60 ± 0.00	734.85 ± 0.00	490.74 ± 0.00
		IMDb4	42.05 ± 0.32	60.33 ± 0.47	2.60 ± 0.00	734.85 ± 0.00	490.74 ± 0.00
Freebase	Original	Freebase1	495.44 ± 112.20	38.00 ± 8.60	88.54 ± 0.00	595.62 ± 0.00	607.50 ± 0.09
		Freebase2	600.92 ± 64.06	36.67 ± 3.86	88.54 ± 0.00	595.63 ± 0.00	609.89 ± 0.08
		Freebase3	1001.06 ± 150.98	34.33 ± 5.73	88.54 ± 0.00	595.64 ± 0.00	633.44 ± 0.01
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	1224.91 ± 208.52	33.00 ± 4.55	88.54 ± 0.00	595.64 ± 0.00	638.77 ± 1.52
	Augmentation	Freebase1-3	747.09 ± 170.62	45.00 ± 10.68	88.54 ± 0.00	547.89 ± 0.00	495.48 ± 0.99
	Invariant	Freebase1	1544.00 ± 320.75	42.33 ± 8.65	88.54 ± 0.00	595.64 ± 0.00	637.69 ± 1.54
		Freebase2	1496.13 ± 322.59	42.33 ± 8.65	88.54 ± 0.00	595.64 ± 0.00	637.69 ± 1.54
		Freebase3	1523.10 ± 313.96	42.33 ± 8.65	88.54 ± 0.00	595.64 ± 0.00	637.69 ± 1.54

Table 19: RGCN: link prediction results (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Precision	Recall	F1	Hits@1	Hits@3	MRR
IMDb (Movie–Director)	Original	IMDb1	0.0843 ± 0.0250	0.0534 ± 0.0697	0.0441 ± 0.0512	0.0191 ± 0.0228	0.1846 ± 0.1266	0.1845 ± 0.0498
		IMDb3	0.0719 ± 0.0267	0.2105 ± 0.1992	0.0684 ± 0.0507	0.0363 ± 0.0471	0.2129 ± 0.1036	0.2063 ± 0.0579
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.0584 ± 0.0094	0.2456 ± 0.1811	0.0636 ± 0.0389	0.0108 ± 0.0051	0.1663 ± 0.0959	0.1766 ± 0.0272
	Augmentation	IMDb1	0.0660 ± 0.0114	0.2169 ± 0.1582	0.0813 ± 0.0543	0.0355 ± 0.0451	0.2743 ± 0.0183	0.2194 ± 0.0322
		IMDb3	0.0502 ± 0.0111	0.4163 ± 0.3194	0.0701 ± 0.0471	0.0255 ± 0.0285	0.2580 ± 0.0236	0.2077 ± 0.0206
	Invariant	IMDb1	0.0778 ± 0.0135	0.2771 ± 0.1994	0.0810 ± 0.0570	0.0052 ± 0.0023	0.2628 ± 0.0006	0.2001 ± 0.0047
		IMDb3	0.0778 ± 0.0135	0.2771 ± 0.1994	0.0810 ± 0.0570	0.0052 ± 0.0023	0.2628 ± 0.0006	0.2001 ± 0.0047
DBLP	Original	DBLP1	0.0927 ± 0.0233	0.6699 ± 0.1684	0.1589 ± 0.0331	0.1397 ± 0.0152	0.3710 ± 0.0365	0.3254 ± 0.0177
		DBLP2	0.0959 ± 0.0469	0.5315 ± 0.4025	0.0876 ± 0.0433	0.1487 ± 0.0177	0.3622 ± 0.0378	0.3289 ± 0.0172
		DBLP3	0.0833 ± 0.0119	0.7283 ± 0.0585	0.1490 ± 0.0190	0.2054 ± 0.0025	0.4322 ± 0.0434	0.3767 ± 0.0161
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	0.0762 ± 0.0618	0.4854 ± 0.3589	0.1279 ± 0.0991	0.2079 ± 0.0537	0.4977 ± 0.1168	0.4024 ± 0.0606
	Augmentation	DBLP1	0.3317 ± 0.2370	0.1770 ± 0.1258	0.1357 ± 0.0955	0.1908 ± 0.0029	0.4305 ± 0.0685	0.3632 ± 0.0249
		DBLP2	0.2672 ± 0.1647	0.1084 ± 0.0766	0.1045 ± 0.0736	0.1622 ± 0.0115	0.3397 ± 0.0398	0.3161 ± 0.0259
		DBLP3	0.2650 ± 0.1662	0.1097 ± 0.0773	0.1039 ± 0.0730	0.1614 ± 0.0124	0.3397 ± 0.0401	0.3163 ± 0.0265
	Invariant	DBLP1	0.0762 ± 0.0618	0.4854 ± 0.3589	0.1279 ± 0.0991	0.2079 ± 0.0537	0.4977 ± 0.1168	0.4024 ± 0.0606
		DBLP2	0.0762 ± 0.0618	0.4854 ± 0.3589	0.1279 ± 0.0991	0.2079 ± 0.0537	0.4977 ± 0.1168	0.4024 ± 0.0606
		DBLP3	0.0762 ± 0.0618	0.4854 ± 0.3589	0.1279 ± 0.0991	0.2079 ± 0.0537	0.4977 ± 0.1168	0.4024 ± 0.0606
WN18RR	Original	WN18RR1	0.9311 ± 0.0180	0.4458 ± 0.0120	0.6026 ± 0.0075	0.3492 ± 0.0100	0.3984 ± 0.0049	0.3819 ± 0.0076
		WN18RR2	0.9623 ± 0.0057	0.4538 ± 0.0051	0.6168 ± 0.0036	0.3621 ± 0.0024	0.4116 ± 0.0035	0.3977 ± 0.0029
		WN18RR3	0.9640 ± 0.0012	0.4315 ± 0.0037	0.5961 ± 0.0034	0.3231 ± 0.0022	0.3904 ± 0.0031	0.3663 ± 0.0017
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	$\bigcup_i$ WN18RR$_i$	0.9457 ± 0.0155	0.4684 ± 0.0088	0.6263 ± 0.0054	0.3310 ± 0.0032	0.3980 ± 0.0033	0.3761 ± 0.0025
	Augmentation	WN18RR1	0.9185 ± 0.0158	0.4728 ± 0.0142	0.6239 ± 0.0091	0.3103 ± 0.0220	0.3874 ± 0.0079	0.3591 ± 0.0142
		WN18RR2	0.9301 ± 0.0095	0.4839 ± 0.0187	0.6363 ± 0.0145	0.3304 ± 0.0185	0.3968 ± 0.0078	0.3752 ± 0.0124
		WN18RR3	0.8941 ± 0.0179	0.4939 ± 0.0205	0.6358 ± 0.0128	0.2687 ± 0.0149	0.3591 ± 0.0101	0.3278 ± 0.0110
	Invariant	WN18RR1	0.9507 ± 0.0095	0.4699 ± 0.0079	0.6288 ± 0.0051	0.3332 ± 0.0031	0.4046 ± 0.0010	0.3802 ± 0.0019
		WN18RR2	0.9507 ± 0.0095	0.4699 ± 0.0079	0.6288 ± 0.0051	0.3332 ± 0.0031	0.4046 ± 0.0010	0.3802 ± 0.0019
		WN18RR3	0.9507 ± 0.0095	0.4699 ± 0.0079	0.6288 ± 0.0051	0.3332 ± 0.0031	0.4046 ± 0.0010	0.3802 ± 0.0019

Table 20: RGCN: link prediction degree of invariance (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ	Kendall-τ@1	Kendall-τ@3
IMDb (Movie–Director)	Original	IMDb1 vs. IMDb3	0.5363 ± 0.2498	0.5803 ± 0.3148	0.4595 ± 0.3720
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	IMDb1 vs. IMDb3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	IMDb1 vs. IMDb3	0.7469 ± 0.2246	0.7974 ± 0.1435	0.8231 ± 0.1763
	Invariant	IMDb1 vs. IMDb3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
DBLP	Original	DBLP1 vs. DBLP2	0.6212 ± 0.1821	0.6682 ± 0.0650	0.7386 ± 0.1848
		DBLP1 vs. DBLP3	0.4037 ± 0.0477	0.4398 ± 0.1217	0.2823 ± 0.1550
		DBLP2 vs. DBLP3	0.3833 ± 0.0796	0.4945 ± 0.0615	0.2997 ± 0.1636
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	DBLP1 vs. DBLP2 DBLP1 vs. DBLP3 DBLP2 vs. DBLP3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	DBLP1 vs. DBLP2	0.8131 ± 0.0277	0.7651 ± 0.0612	0.7324 ± 0.0508
		DBLP1 vs. DBLP3	0.8322 ± 0.0209	0.7605 ± 0.0550	0.7383 ± 0.0498
		DBLP2 vs. DBLP3	0.9643 ± 0.0123	0.9805 ± 0.0039	0.9818 ± 0.0051
	Invariant	DBLP1 vs. DBLP2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP1 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP2 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WN18RR	Original	WN18RR1 vs. WN18RR2	0.2113 ± 0.0514	0.7151 ± 0.0046	0.6486 ± 0.0095
		WN18RR1 vs. WN18RR3	0.3747 ± 0.0141	0.7301 ± 0.0094	0.7102 ± 0.0043
		WN18RR2 vs. WN18RR3	0.1300 ± 0.0296	0.6709 ± 0.0089	0.6099 ± 0.0179
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	WN18RR1 vs. WN18RR2 WN18RR1 vs. WN18RR3 WN18RR2 vs. WN18RR3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	WN18RR1 vs. WN18RR2	0.5744 ± 0.0122	0.8589 ± 0.0113	0.7979 ± 0.0270
		WN18RR1 vs. WN18RR3	0.8078 ± 0.0047	0.8763 ± 0.0049	0.8651 ± 0.0040
		WN18RR2 vs. WN18RR3	0.5216 ± 0.0187	0.8334 ± 0.0091	0.7914 ± 0.0150
	Invariant	WN18RR1 vs. WN18RR2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		WN18RR1 vs. WN18RR3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		WN18RR2 vs. WN18RR3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000

Table 21: RGCN: link prediction efficiency (IMDb (Movie–Director), DBLP, WN18RR). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb (Movie–Director)	Original	IMDb1	204.83 ± 14.62	222.00 ± 5.35	10.85 ± 0.00	1166.37 ± 0.00	583.45 ± 0.00
		IMDb3	238.12 ± 31.68	243.67 ± 24.55	10.85 ± 0.00	1166.28 ± 0.00	583.15 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	209.62 ± 8.81	227.00 ± 5.10	10.85 ± 0.00	1331.87 ± 0.58	584.48 ± 0.28
	Augmentation	IMDb1,3	114.06 ± 5.71	211.33 ± 7.36	10.85 ± 0.00	1353.58 ± 0.27	770.24 ± 0.41
	Invariant	IMDb1	217.77 ± 24.83	226.33 ± 21.85	10.85 ± 0.00	1439.42 ± 0.57	582.91 ± 0.12
		IMDb3	228.72 ± 34.73	226.33 ± 21.85	10.85 ± 0.00	1439.42 ± 0.57	582.91 ± 0.12
DBLP	Original	DBLP1	47.81 ± 0.14	202.33 ± 0.47	20.86 ± 0.00	2812.39 ± 0.00	596.48 ± 0.20
		DBLP2	47.94 ± 3.03	211.33 ± 12.55	20.86 ± 0.00	2681.43 ± 0.57	596.79 ± 0.55
		DBLP3	47.29 ± 0.98	208.33 ± 3.40	20.86 ± 0.00	2730.20 ± 0.32	595.89 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	55.38 ± 0.42	205.33 ± 2.49	20.86 ± 0.00	2863.49 ± 0.68	596.45 ± 0.63
	Augmentation	DBLP1-3	252.36 ± 65.46	264.00 ± 67.93	20.86 ± 0.00	1842.74 ± 0.61	1316.07 ± 0.34
	Invariant	DBLP1	55.45 ± 0.99	205.33 ± 2.49	20.86 ± 0.00	2863.33 ± 0.68	595.13 ± 0.63
		DBLP2	55.69 ± 0.75	205.33 ± 2.49	20.86 ± 0.00	2864.64 ± 0.98	596.26 ± 0.05
		DBLP3	55.68 ± 0.89	205.33 ± 2.49	20.86 ± 0.00	2864.64 ± 0.98	596.26 ± 0.05
WN18RR	Original	WN18RR1	1798.42 ± 594.14	2976.67 ± 986.25	36.22 ± 0.00	964.45 ± 0.00	608.93 ± 0.36
		WN18RR2	1832.12 ± 112.46	2776.67 ± 173.08	36.22 ± 0.00	1014.73 ± 0.22	768.11 ± 0.75
		WN18RR3	3532.24 ± 439.89	4080.00 ± 477.70	36.22 ± 0.00	2071.11 ± 0.12	1647.74 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	$\bigcup_i$ WN18RR$_i$	2300.75 ± 430.04	2440.00 ± 450.85	36.22 ± 0.00	2535.64 ± 0.26	1975.92 ± 0.09
	Augmentation	WN18RR1-3	19366.00 ± 4141.42	2939.00 ± 570.44	37.80 ± 0.00	11288.02 ± 0.53	11229.00 ± 27.49
	Invariant	WN18RR1	2519.12 ± 272.50	2710.00 ± 287.17	36.22 ± 0.00	2536.15 ± 0.23	1976.37 ± 0.53
		WN18RR2	2536.24 ± 273.24	2710.00 ± 287.17	36.22 ± 0.00	2536.15 ± 0.23	1976.37 ± 0.53
		WN18RR3	2523.68 ± 284.47	2710.00 ± 287.17	36.22 ± 0.00	2536.15 ± 0.23	1976.37 ± 0.53

Table 22: SlotGAT: node classification results (IMDb, Freebase). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Accuracy	Precision	Recall	Micro-F1	Macro-F1
IMDb	Original	IMDb1	0.6507 ± 0.0039	0.6562 ± 0.0045	0.6502 ± 0.0056	0.6507 ± 0.0039	0.6511 ± 0.0044
		IMDb2	0.6575 ± 0.0084	0.6645 ± 0.0038	0.6551 ± 0.0111	0.6575 ± 0.0084	0.6574 ± 0.0092
		IMDb3	0.6571 ± 0.0006	0.6622 ± 0.0027	0.6559 ± 0.0014	0.6571 ± 0.0006	0.6582 ± 0.0009
		IMDb4	0.6555 ± 0.0153	0.6591 ± 0.0175	0.6553 ± 0.0132	0.6555 ± 0.0153	0.6556 ± 0.0153
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.6519 ± 0.0102	0.6531 ± 0.0112	0.6545 ± 0.0062	0.6519 ± 0.0102	0.6527 ± 0.0095
	Augmentation	IMDb1	0.6435 ± 0.0122	0.6455 ± 0.0112	0.6449 ± 0.0137	0.6435 ± 0.0122	0.6442 ± 0.0132
		IMDb2	0.6463 ± 0.0195	0.6492 ± 0.0196	0.6474 ± 0.0185	0.6463 ± 0.0195	0.6475 ± 0.0198
		IMDb3	0.6455 ± 0.0187	0.6479 ± 0.0188	0.6462 ± 0.0190	0.6455 ± 0.0187	0.6462 ± 0.0198
		IMDb4	0.6531 ± 0.0141	0.6555 ± 0.0135	0.6540 ± 0.0153	0.6531 ± 0.0141	0.6540 ± 0.0154
	Invariant	IMDb1	0.6611 ± 0.0145	0.6624 ± 0.0156	0.6623 ± 0.0131	0.6611 ± 0.0145	0.6621 ± 0.0145
		IMDb2	0.6611 ± 0.0145	0.6624 ± 0.0156	0.6623 ± 0.0131	0.6611 ± 0.0145	0.6621 ± 0.0145
		IMDb3	0.6611 ± 0.0145	0.6624 ± 0.0156	0.6623 ± 0.0131	0.6611 ± 0.0145	0.6621 ± 0.0145
		IMDb4	0.6611 ± 0.0145	0.6624 ± 0.0156	0.6623 ± 0.0131	0.6611 ± 0.0145	0.6621 ± 0.0145
Freebase	Original	Freebase1	0.6602 ± 0.0100	0.6092 ± 0.0356	0.3969 ± 0.0223	0.6602 ± 0.0100	0.4350 ± 0.0157
		Freebase2	0.6570 ± 0.0065	0.6349 ± 0.0593	0.3607 ± 0.0006	0.6570 ± 0.0065	0.3906 ± 0.0022
		Freebase3	0.6478 ± 0.0119	0.5810 ± 0.0703	0.3644 ± 0.0124	0.6478 ± 0.0119	0.3899 ± 0.0212
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	0.6396 ± 0.0061	0.5918 ± 0.0471	0.3529 ± 0.0004	0.6396 ± 0.0061	0.3787 ± 0.0035
	Augmentation	Freebase1	0.6503 ± 0.0100	0.6203 ± 0.0483	0.3827 ± 0.0092	0.6503 ± 0.0100	0.4202 ± 0.0095
		Freebase2	0.6489 ± 0.0057	0.5527 ± 0.0267	0.3656 ± 0.0108	0.6489 ± 0.0057	0.3921 ± 0.0124
		Freebase3	0.5990 ± 0.0027	0.5229 ± 0.0043	0.3011 ± 0.0079	0.5990 ± 0.0027	0.3250 ± 0.0093
	Invariant	Freebase1	0.6396 ± 0.0061	0.5918 ± 0.0471	0.3529 ± 0.0004	0.6396 ± 0.0061	0.3787 ± 0.0035
		Freebase2	0.6396 ± 0.0061	0.5918 ± 0.0471	0.3529 ± 0.0004	0.6396 ± 0.0061	0.3787 ± 0.0035
		Freebase3	0.6396 ± 0.0061	0.5918 ± 0.0471	0.3529 ± 0.0004	0.6396 ± 0.0061	0.3787 ± 0.0035

Table 23: SlotGAT: node classification degree of invariance (IMDb, Freebase). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ
IMDb	Original	IMDb1 vs. IMDb2	0.7937 ± 0.0246
		IMDb1 vs. IMDb3	0.7602 ± 0.0360
		IMDb1 vs. IMDb4	0.8131 ± 0.0289
		IMDb2 vs. IMDb3	0.8161 ± 0.0209
		IMDb2 vs. IMDb4	0.8147 ± 0.0248
		IMDb3 vs. IMDb4	0.7924 ± 0.0287
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	IMDb1 vs. IMDb2	1.000 ± 0.0000 (trivial)
		IMDb1 vs. IMDb3	
		IMDb1 vs. IMDb4	
		IMDb2 vs. IMDb3	
		IMDb2 vs. IMDb4	
		IMDb3 vs. IMDb4	
	Augmentation	IMDb1 vs. IMDb2	0.9327 ± 0.0225
		IMDb1 vs. IMDb3	0.9351 ± 0.0191
		IMDb1 vs. IMDb4	0.9325 ± 0.0314
		IMDb2 vs. IMDb3	0.9258 ± 0.0263
		IMDb2 vs. IMDb4	0.9163 ± 0.0163
		IMDb3 vs. IMDb4	0.9415 ± 0.0134
	Invariant	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
Freebase	Original	Freebase1 vs. Freebase2	0.6063 ± 0.0925
		Freebase1 vs. Freebase3	0.4956 ± 0.0578
		Freebase2 vs. Freebase3	0.5522 ± 0.0428
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Augmentation	Freebase1 vs. Freebase2	0.8661 ± 0.0140
		Freebase1 vs. Freebase3	0.7054 ± 0.0451
		Freebase2 vs. Freebase3	0.7331 ± 0.0425
	Invariant	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000

Table 24: SlotGAT: node classification efficiency (IMDb, Freebase). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb	Original	IMDb1	37.74 ± 7.79	55.67 ± 6.80	14.68 ± 0.00	4173.73 ± 0.80	2274.81 ± 0.80
		IMDb2	39.82 ± 2.44	53.00 ± 3.27	14.68 ± 0.00	4171.56 ± 0.00	2196.46 ± 0.00
		IMDb3	40.16 ± 2.77	57.67 ± 8.73	14.68 ± 0.00	4172.02 ± 0.00	2194.16 ± 0.00
		IMDb4	39.93 ± 9.25	56.00 ± 5.89	14.68 ± 0.00	4171.46 ± 0.00	2196.50 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	53.80 ± 3.97	52.00 ± 4.90	14.68 ± 0.00	5632.68 ± 1.80	2709.32 ± 0.77
	Augmentation	IMDb1-4	37.40 ± 3.68	57.33 ± 4.99	14.68 ± 0.00	4181.45 ± 1.15	2282.91 ± 1.02
	Invariant	IMDb1	54.19 ± 0.84	52.00 ± 2.16	14.68 ± 0.00	5997.66 ± 0.00	3338.35 ± 0.00
		IMDb2	54.86 ± 0.64	52.00 ± 2.16	14.68 ± 0.00	5997.66 ± 0.00	3338.35 ± 0.00
		IMDb3	51.81 ± 1.83	52.00 ± 2.16	14.68 ± 0.00	5997.66 ± 0.00	3338.35 ± 0.00
		IMDb4	53.54 ± 3.36	52.00 ± 2.16	14.68 ± 0.00	5997.66 ± 0.00	3338.35 ± 0.00
Freebase	Original	Freebase1	86.08 ± 14.59	33.00 ± 0.82	44.34 ± 0.00	4460.76 ± 0.00	2361.48 ± 0.00
		Freebase2	1908.23 ± 43.47	33.67 ± 0.94	44.34 ± 0.00	5164.77 ± 0.00	3082.54 ± 0.00
		Freebase3	45960.03 ± 17904.69	84.00 ± 32.78	44.34 ± 0.00	10170.89 ± 0.00	8090.57 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	36365.66 ± 7189.45	68.00 ± 13.49	44.34 ± 0.00	10173.09 ± 0.00	8092.00 ± 0.00
	Augmentation	Freebase1-3	19509.95 ± 1497.36	97.00 ± 7.48	44.34 ± 0.00	10896.37 ± 0.00	8813.75 ± 0.00
	Invariant	Freebase1	36484.00 ± 7273.66	68.00 ± 13.49	44.34 ± 0.00	10173.09 ± 0.00	8092.00 ± 0.00
		Freebase2	36379.80 ± 7204.16	68.00 ± 13.49	44.34 ± 0.00	10173.09 ± 0.00	8092.00 ± 0.00
		Freebase3	36364.23 ± 7256.29	68.00 ± 13.49	44.34 ± 0.00	10173.09 ± 0.00	8092.00 ± 0.00

Table 25: SlotGAT: link prediction results (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Precision	Recall	F1	Hits@1	Hits@3	MRR
IMDb (Movie–Director)	Original	IMDb1	0.1221 ± 0.0123	0.1120 ± 0.0223	0.1148 ± 0.0082	0.1144 ± 0.0089	0.2659 ± 0.0205	0.2637 ± 0.0149
		IMDb3	0.1054 ± 0.0082	0.0981 ± 0.0373	0.0966 ± 0.0174	0.0933 ± 0.0045	0.2408 ± 0.0065	0.2431 ± 0.0066
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.1139 ± 0.0042	0.0993 ± 0.0213	0.1051 ± 0.0126	0.1112 ± 0.0045	0.2612 ± 0.0123	0.2610 ± 0.0066
	Augmentation	IMDb1	0.1078 ± 0.0248	0.1866 ± 0.0432	0.1307 ± 0.0115	0.1148 ± 0.0148	0.2532 ± 0.0225	0.2543 ± 0.0190
		IMDb3	0.1073 ± 0.0223	0.1691 ± 0.0331	0.1267 ± 0.0082	0.1120 ± 0.0126	0.2444 ± 0.0192	0.2526 ± 0.0167
	Invariant	IMDb1	0.1246 ± 0.0231	0.0793 ± 0.0118	0.0956 ± 0.0098	0.0901 ± 0.0049	0.2508 ± 0.0137	0.2514 ± 0.0088
		IMDb3	0.1246 ± 0.0231	0.0793 ± 0.0118	0.0956 ± 0.0098	0.0901 ± 0.0049	0.2508 ± 0.0137	0.2514 ± 0.0088
DBLP	Original	DBLP1	0.1869 ± 0.0081	0.8178 ± 0.0334	0.3039 ± 0.0085	0.3421 ± 0.0038	0.7294 ± 0.0115	0.5561 ± 0.0064
		DBLP2	0.1990 ± 0.0252	0.7519 ± 0.0901	0.3114 ± 0.0214	0.3320 ± 0.0055	0.6977 ± 0.0181	0.5418 ± 0.0079
		DBLP3	0.1924 ± 0.0060	0.8117 ± 0.0269	0.3109 ± 0.0062	0.3432 ± 0.0075	0.7266 ± 0.0159	0.5557 ± 0.0081
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	0.1954 ± 0.0156	0.8117 ± 0.0244	0.3144 ± 0.0182	0.3423 ± 0.0124	0.7277 ± 0.0134	0.5565 ± 0.0101
	Augmentation	DBLP1	0.2014 ± 0.0153	0.7518 ± 0.0212	0.3172 ± 0.0177	0.3182 ± 0.0182	0.7012 ± 0.0156	0.5330 ± 0.0145
		DBLP2	0.2159 ± 0.0122	0.7164 ± 0.0439	0.3311 ± 0.0121	0.3181 ± 0.0202	0.6992 ± 0.0148	0.5332 ± 0.0154
		DBLP3	0.2054 ± 0.0179	0.7516 ± 0.0209	0.3220 ± 0.0204	0.3218 ± 0.0196	0.7041 ± 0.0143	0.5367 ± 0.0154
	Invariant	DBLP1	0.1954 ± 0.0156	0.8117 ± 0.0244	0.3144 ± 0.0182	0.3423 ± 0.0124	0.7277 ± 0.0134	0.5565 ± 0.0101
		DBLP2	0.1954 ± 0.0156	0.8117 ± 0.0244	0.3144 ± 0.0182	0.3423 ± 0.0124	0.7277 ± 0.0134	0.5565 ± 0.0101
		DBLP3	0.1954 ± 0.0156	0.8117 ± 0.0244	0.3144 ± 0.0182	0.3423 ± 0.0124	0.7277 ± 0.0134	0.5565 ± 0.0101
WN18RR	Original	WN18RR1	0.9712 ± 0.0035	0.3996 ± 0.0059	0.5662 ± 0.0053	0.3481 ± 0.0014	0.3647 ± 0.0013	0.3658 ± 0.0011
		WN18RR2	0.9624 ± 0.0131	0.4098 ± 0.0200	0.5744 ± 0.0175	0.3466 ± 0.0015	0.3653 ± 0.0010	0.3660 ± 0.0003
		WN18RR3	0.9520 ± 0.0166	0.4148 ± 0.0120	0.5776 ± 0.0084	0.3495 ± 0.0015	0.3653 ± 0.0005	0.3652 ± 0.0009
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	$\bigcup_i$ WN18RR$_i$	0.9770 ± 0.0081	0.3828 ± 0.0128	0.5499 ± 0.0121	0.3499 ± 0.0006	0.3618 ± 0.0007	0.3635 ± 0.0001
	Augmentation	WN18RR1	0.9743 ± 0.0038	0.3946 ± 0.0072	0.5617 ± 0.0067	0.3471 ± 0.0017	0.3637 ± 0.0022	0.3638 ± 0.0013
		WN18RR2	0.9766 ± 0.0036	0.3954 ± 0.0075	0.5629 ± 0.0070	0.3459 ± 0.0028	0.3643 ± 0.0016	0.3635 ± 0.0016
		WN18RR3	0.9684 ± 0.0054	0.3909 ± 0.0074	0.5569 ± 0.0067	0.3485 ± 0.0005	0.3607 ± 0.0026	0.3624 ± 0.0018
	Invariant	WN18RR1	0.9714 ± 0.0053	0.3982 ± 0.0157	0.5646 ± 0.0150	0.3501 ± 0.0011	0.3660 ± 0.0023	0.3662 ± 0.0017
		WN18RR2	0.9714 ± 0.0053	0.3982 ± 0.0157	0.5646 ± 0.0150	0.3501 ± 0.0011	0.3660 ± 0.0023	0.3662 ± 0.0017
		WN18RR3	0.9714 ± 0.0053	0.3982 ± 0.0157	0.5646 ± 0.0150	0.3501 ± 0.0011	0.3660 ± 0.0023	0.3662 ± 0.0017

Table 26: SlotGAT: link prediction degree of invariance (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ	Kendall-τ@1	Kendall-τ@3
IMDb (Movie–Director)	Original	IMDb1 vs. IMDb3	0.4358 ± 0.0459	0.4241 ± 0.0658	0.5140 ± 0.0489
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	IMDb1 vs. IMDb3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	IMDb1 vs. IMDb3	0.8860 ± 0.0192	0.8874 ± 0.0317	0.9063 ± 0.0154
	Invariant	IMDb1 vs. IMDb3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
DBLP	Original	DBLP1 vs. DBLP2	0.6793 ± 0.0255	0.5220 ± 0.0420	0.6440 ± 0.0313
		DBLP1 vs. DBLP3	0.6682 ± 0.0667	0.4892 ± 0.0435	0.6417 ± 0.0661
		DBLP2 vs. DBLP3	0.6453 ± 0.0072	0.4754 ± 0.0108	0.6047 ± 0.0311
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	DBLP1 vs. DBLP2 DBLP1 vs. DBLP3 DBLP2 vs. DBLP3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	DBLP1 vs. DBLP2	0.9615 ± 0.0132	0.9502 ± 0.0189	0.9608 ± 0.0111
		DBLP1 vs. DBLP3	0.9645 ± 0.0025	0.9425 ± 0.0112	0.9554 ± 0.0032
		DBLP2 vs. DBLP3	0.9627 ± 0.0057	0.9438 ± 0.0294	0.9552 ± 0.0145
	Invariant	DBLP1 vs. DBLP2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP1 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP2 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WN18RR	Original	WN18RR1 vs. WN18RR2	0.3075 ± 0.0134	0.7484 ± 0.0086	0.7192 ± 0.0111
		WN18RR1 vs. WN18RR3	0.2528 ± 0.0046	0.7140 ± 0.0086	0.6723 ± 0.0036
		WN18RR2 vs. WN18RR3	0.2341 ± 0.0054	0.7019 ± 0.0053	0.6556 ± 0.0043
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	WN18RR1 vs. WN18RR2 WN18RR1 vs. WN18RR3 WN18RR2 vs. WN18RR3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	WN18RR1 vs. WN18RR2	0.9223 ± 0.0061	0.9516 ± 0.0018	0.9527 ± 0.0022
		WN18RR1 vs. WN18RR3	0.7700 ± 0.0279	0.8603 ± 0.0120	0.8499 ± 0.0185
		WN18RR2 vs. WN18RR3	0.7687 ± 0.0226	0.8626 ± 0.0113	0.8541 ± 0.0142
	Invariant	WN18RR1 vs. WN18RR2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		WN18RR1 vs. WN18RR3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		WN18RR2 vs. WN18RR3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000

Table 27: SlotGAT: link prediction efficiency (IMDb (Movie–Director), DBLP, WN18RR). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb (Movie–Director)	Original	IMDb1	37.75 ± 7.82	48.33 ± 10.14	21.33 ± 0.00	5369.56 ± 0.00	2363.86 ± 0.00
		IMDb3	36.41 ± 9.05	46.67 ± 11.81	21.33 ± 0.00	5369.56 ± 0.00	2364.17 ± 0.43
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	64.83 ± 19.57	49.00 ± 5.10	21.33 ± 0.00	6898.24 ± 0.41	2880.82 ± 0.54
	Augmentation	IMDb1,3	55.66 ± 8.98	52.00 ± 4.32	21.33 ± 0.00	5371.07 ± 0.38	2366.58 ± 0.36
	Invariant	IMDb1	54.86 ± 6.68	49.67 ± 7.41	21.33 ± 0.00	7715.22 ± 0.79	3172.96 ± 0.60
		IMDb3	54.83 ± 6.66	49.67 ± 7.41	21.33 ± 0.00	7715.22 ± 0.79	3172.96 ± 0.60
DBLP	Original	DBLP1	177.30 ± 40.73	24.67 ± 0.94	26.72 ± 0.00	21258.79 ± 0.00	7824.10 ± 0.25
		DBLP2	172.54 ± 37.44	26.33 ± 1.25	26.72 ± 0.00	20141.99 ± 0.76	7439.08 ± 1.33
		DBLP3	151.85 ± 10.47	24.67 ± 0.94	26.72 ± 0.00	20555.04 ± 0.42	7579.42 ± 1.38
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	149.47 ± 2.77	24.67 ± 0.47	26.72 ± 0.00	21691.27 ± 0.24	7971.00 ± 0.11
	Augmentation	DBLP1-3	399.90 ± 29.90	66.00 ± 0.00	26.72 ± 0.00	21286.68 ± 0.00	7852.29 ± 0.00
	Invariant	DBLP1	149.40 ± 2.68	24.67 ± 0.47	26.72 ± 0.00	21691.27 ± 0.24	7971.00 ± 0.11
		DBLP2	160.82 ± 17.93	24.67 ± 0.47	26.72 ± 0.00	21691.27 ± 0.24	7971.00 ± 0.11
		DBLP3	160.92 ± 17.90	24.67 ± 0.47	26.72 ± 0.00	21691.27 ± 0.24	7971.00 ± 0.11
WN18RR	Original	WN18RR1	972.35 ± 74.79	195.00 ± 14.72	25.98 ± 0.00	9391.90 ± 0.00	1391.33 ± 0.84
		WN18RR2	1333.73 ± 409.77	201.67 ± 18.41	25.98 ± 0.00	9752.41 ± 0.46	1541.23 ± 0.41
		WN18RR3	1625.15 ± 208.47	110.00 ± 7.07	25.98 ± 0.00	22718.33 ± 0.00	6958.58 ± 0.32
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	$\bigcup_i$ WN18RR$_i$	2190.94 ± 293.84	150.00 ± 21.60	25.98 ± 0.00	23672.24 ± 0.00	7357.07 ± 0.43
	Augmentation	WN18RR1-3	1202.49 ± 79.53	120.00 ± 8.49	25.98 ± 0.00	22729.33 ± 0.00	1428.12 ± 0.32
	Invariant	WN18RR1	1929.48 ± 105.42	140.00 ± 8.16	25.98 ± 0.00	23672.24 ± 0.00	7356.48 ± 0.21
		WN18RR2	2150.56 ± 234.67	140.00 ± 8.16	25.98 ± 0.00	23672.24 ± 0.00	7356.48 ± 0.21
		WN18RR3	1922.36 ± 113.22	140.00 ± 8.16	25.98 ± 0.00	23672.24 ± 0.00	7356.48 ± 0.21

Table 28: CMPNN: link prediction results (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Precision	Recall	F1	Hits@1	Hits@3	MRR
IMDb (Movie–Director)	Original	IMDb1	0.0979 ± 0.0050	0.7835 ± 0.0269	0.1740 ± 0.0073	0.2604 ± 0.0071	0.4394 ± 0.0264	0.4152 ± 0.0109
		IMDb3	0.0893 ± 0.0028	0.7951 ± 0.0182	0.1605 ± 0.0042	0.1707 ± 0.0057	0.3724 ± 0.0139	0.3430 ± 0.0072
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.0985 ± 0.0056	0.7959 ± 0.0295	0.1751 ± 0.0083	0.2576 ± 0.0023	0.4426 ± 0.0190	0.4171 ± 0.0061
	Augmentation	IMDb1	0.0972 ± 0.0003	0.7967 ± 0.0076	0.1732 ± 0.0004	0.2580 ± 0.0091	0.4362 ± 0.0132	0.4140 ± 0.0093
		IMDb3	0.0911 ± 0.0009	0.7628 ± 0.0290	0.1628 ± 0.0010	0.1750 ± 0.0060	0.3796 ± 0.0104	0.3478 ± 0.0041
	Invariant	IMDb1	0.1020 ± 0.0066	0.7504 ± 0.0531	0.1792 ± 0.0088	0.2608 ± 0.0045	0.4625 ± 0.0104	0.4228 ± 0.0066
		IMDb3	0.1020 ± 0.0066	0.7504 ± 0.0531	0.1792 ± 0.0088	0.2608 ± 0.0045	0.4625 ± 0.0104	0.4228 ± 0.0066
DBLP	Original	DBLP1	0.2211 ± 0.0134	0.8878 ± 0.0107	0.3537 ± 0.0164	0.4121 ± 0.0038	0.8017 ± 0.0033	0.6195 ± 0.0026
		DBLP2	0.2105 ± 0.0247	0.8726 ± 0.0360	0.3378 ± 0.0295	0.4125 ± 0.0064	0.7819 ± 0.0042	0.6132 ± 0.0039
		DBLP3	0.2232 ± 0.0235	0.8087 ± 0.0310	0.3486 ± 0.0251	0.4060 ± 0.0080	0.7967 ± 0.0032	0.6150 ± 0.0053
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	0.2083 ± 0.0067	0.8395 ± 0.0075	0.3337 ± 0.0092	0.4115 ± 0.0009	0.8021 ± 0.0027	0.6194 ± 0.0012
	Augmentation	DBLP1	0.1831 ± 0.0624	0.8867 ± 0.0959	0.2947 ± 0.0760	0.4147 ± 0.0101	0.7923 ± 0.0094	0.6178 ± 0.0089
		DBLP2	0.2318 ± 0.0249	0.8443 ± 0.0501	0.3620 ± 0.0248	0.4157 ± 0.0025	0.7873 ± 0.0048	0.6166 ± 0.0029
		DBLP3	0.1811 ± 0.0513	0.9060 ± 0.0635	0.2966 ± 0.0672	0.4082 ± 0.0054	0.7997 ± 0.0038	0.6161 ± 0.0050
	Invariant	DBLP1	0.2083 ± 0.0067	0.8395 ± 0.0075	0.3337 ± 0.0092	0.4115 ± 0.0009	0.8021 ± 0.0027	0.6194 ± 0.0012
		DBLP2	0.2083 ± 0.0067	0.8395 ± 0.0075	0.3337 ± 0.0092	0.4115 ± 0.0009	0.8021 ± 0.0027	0.6194 ± 0.0012
		DBLP3	0.2083 ± 0.0067	0.8395 ± 0.0075	0.3337 ± 0.0092	0.4115 ± 0.0009	0.8021 ± 0.0027	0.6194 ± 0.0012
WN18RR	Original	WN18RR1	0.1397 ± 0.0022	0.9399 ± 0.0026	0.2433 ± 0.0032	0.3680 ± 0.0105	0.4418 ± 0.0107	0.4238 ± 0.0095
		WN18RR2	0.9426 ± 0.0647	0.3009 ± 0.0749	0.4532 ± 0.0961	0.2879 ± 0.0697	0.3125 ± 0.0538	0.3075 ± 0.0559
		WN18RR3	0.4584 ± 0.0437	0.6406 ± 0.0259	0.5340 ± 0.0382	0.2186 ± 0.0281	0.3300 ± 0.0319	0.2927 ± 0.0306
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	$\bigcup_i$ WN18RR$_i$	0.8021 ± 0.0633	0.3902 ± 0.0047	0.5241 ± 0.0123	0.3385 ± 0.0037	0.3616 ± 0.0035	0.3524 ± 0.0036
	Augmentation	WN18RR1	0.4357 ± 0.0229	0.6540 ± 0.1420	0.5157 ± 0.0527	0.3220 ± 0.0165	0.3748 ± 0.0224	0.3625 ± 0.0185
		WN18RR2	0.3650 ± 0.2593	0.6149 ± 0.1262	0.3699 ± 0.1712	0.2774 ± 0.0574	0.3277 ± 0.0494	0.3136 ± 0.0486
		WN18RR3	0.3945 ± 0.0972	0.6358 ± 0.0959	0.4856 ± 0.1033	0.2066 ± 0.0253	0.3107 ± 0.0361	0.2748 ± 0.0358
	Invariant	WN18RR1	0.1397 ± 0.0022	0.9399 ± 0.0026	0.2433 ± 0.0032	0.3680 ± 0.0105	0.4418 ± 0.0107	0.4238 ± 0.0095
		WN18RR2	0.1397 ± 0.0022	0.9399 ± 0.0026	0.2433 ± 0.0032	0.3680 ± 0.0105	0.4418 ± 0.0107	0.4238 ± 0.0095
		WN18RR3	0.1397 ± 0.0022	0.9399 ± 0.0026	0.2433 ± 0.0032	0.3680 ± 0.0105	0.4418 ± 0.0107	0.4238 ± 0.0095

Table 29: CMPNN: link prediction degree of invariance (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ	Kendall-τ@1	Kendall-τ@3
IMDb (Movie–Director)	Original	IMDb1 vs. IMDb3	0.7819 ± 0.0238	0.5066 ± 0.0189	0.4818 ± 0.0287
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	IMDb1 vs. IMDb3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	IMDb1 vs. IMDb3	0.7996 ± 0.0552	0.5459 ± 0.0508	0.5478 ± 0.0683
	Invariant	IMDb1 vs. IMDb3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
DBLP	Original	DBLP1 vs. DBLP2	0.6863 ± 0.0157	0.7306 ± 0.0308	0.7431 ± 0.0100
		DBLP1 vs. DBLP3	0.8298 ± 0.0176	0.7490 ± 0.0122	0.8202 ± 0.0068
		DBLP2 vs. DBLP3	0.7108 ± 0.0150	0.6908 ± 0.0353	0.7393 ± 0.0080
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	DBLP1 vs. DBLP2 DBLP1 vs. DBLP3 DBLP2 vs. DBLP3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	DBLP1 vs. DBLP2	0.8342 ± 0.0189	0.8280 ± 0.0262	0.8265 ± 0.0290
		DBLP1 vs. DBLP3	0.8750 ± 0.0135	0.7899 ± 0.0220	0.8335 ± 0.0105
		DBLP2 vs. DBLP3	0.8092 ± 0.0296	0.7657 ± 0.0156	0.7591 ± 0.0128
	Invariant	DBLP1 vs. DBLP2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP1 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP2 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WN18RR	Original	WN18RR1 vs. WN18RR2	-0.0706 ± 0.0827	0.4796 ± 0.0132	0.4168 ± 0.0138
		WN18RR1 vs. WN18RR3	0.1974 ± 0.0362	0.5846 ± 0.0231	0.5433 ± 0.0146
		WN18RR2 vs. WN18RR3	-0.0399 ± 0.1129	0.6329 ± 0.0269	0.5472 ± 0.0060
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	WN18RR1 vs. WN18RR2 WN18RR1 vs. WN18RR3 WN18RR2 vs. WN18RR3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	WN18RR1 vs. WN18RR2	0.3145 ± 0.1855	0.7041 ± 0.1035	0.6996 ± 0.0677
		WN18RR1 vs. WN18RR3	0.4354 ± 0.0991	0.7299 ± 0.0262	0.7128 ± 0.0051
		WN18RR2 vs. WN18RR3	0.2576 ± 0.1786	0.6590 ± 0.0789	0.6642 ± 0.0390
	Invariant	WN18RR1 vs. WN18RR2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		WN18RR1 vs. WN18RR3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		WN18RR2 vs. WN18RR3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000

Table 30: CMPNN: link prediction efficiency (IMDb (Movie–Director), DBLP, WN18RR). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb (Movie–Director)	Original	IMDb1	660.61 ± 239.16	12.67 ± 4.64	0.33 ± 0.00	3637.63 ± 0.00	613.21 ± 0.00
		IMDb3	802.61 ± 50.77	15.33 ± 0.94	0.33 ± 0.00	3637.63 ± 0.00	613.22 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	812.46 ± 254.59	14.33 ± 4.50	0.33 ± 0.00	4235.33 ± 0.00	665.72 ± 0.00
	Augmentation	IMDb1,3	699.08 ± 99.16	13.33 ± 1.89	0.33 ± 0.00	3640.93 ± 0.00	617.18 ± 0.00
	Invariant	IMDb1	764.02 ± 242.36	12.67 ± 4.03	0.33 ± 0.00	4571.96 ± 0.00	695.10 ± 0.00
		IMDb3	761.82 ± 243.93	12.67 ± 4.03	0.33 ± 0.00	4571.96 ± 0.00	695.10 ± 0.00
DBLP	Original	DBLP1	16862.43 ± 10.62	20.00 ± 0.00	0.33 ± 0.00	10918.23 ± 0.69	1408.45 ± 1.11
		DBLP2	13006.48 ± 3968.06	15.67 ± 4.78	0.33 ± 0.00	10480.52 ± 0.76	1368.82 ± 0.19
		DBLP3	16629.12 ± 8.52	20.00 ± 0.00	0.33 ± 0.00	10631.00 ± 0.57	1381.55 ± 0.17
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	17148.52 ± 6.49	20.00 ± 0.00	0.33 ± 0.00	11091.10 ± 0.65	1436.76 ± 0.89
	Augmentation	DBLP1-3	17519.95 ± 18.40	21.00 ± 0.00	0.33 ± 0.00	10964.34 ± 1.42	1455.61 ± 0.92
	Invariant	DBLP1	17172.04 ± 20.50	20.00 ± 0.00	0.33 ± 0.00	11091.10 ± 0.65	1436.76 ± 0.89
		DBLP2	17178.16 ± 54.63	20.00 ± 0.00	0.33 ± 0.00	11091.10 ± 0.65	1436.76 ± 0.89
		DBLP3	17174.11 ± 37.32	20.00 ± 0.00	0.33 ± 0.00	11091.10 ± 0.65	1436.76 ± 0.89
WN18RR	Original	WN18RR1	9697.72 ± 1406.58	8.67 ± 1.25	0.08 ± 0.00	16359.06 ± 0.48	7560.76 ± 0.22
		WN18RR2	14895.44 ± 2346.62	9.00 ± 1.41	0.10 ± 0.00	17439.83 ± 0.80	8407.53 ± 0.93
		WN18RR3	48017.92 ± 2359.54	9.67 ± 0.47	0.98 ± 0.00	17758.32 ± 0.47	9184.92 ± 0.02
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ WN18RR$_i$	$\bigcup_i$ WN18RR$_i$	59442.69 ± 8.67	10.00 ± 0.00	3.63 ± 0.00	17887.12 ± 1.64	9244.44 ± 1.63
	Augmentation	WN18RR1-3	20669.40 ± 3626.06	8.00 ± 1.41	1.00 ± 0.00	17798.21 ± 1.18	9227.05 ± 0.76
	Invariant	WN18RR1	9760.47 ± 1414.38	8.67 ± 1.25	0.08 ± 0.00	16359.06 ± 0.48	7560.76 ± 0.22
		WN18RR2	9766.54 ± 1398.71	8.67 ± 1.25	0.08 ± 0.00	16359.06 ± 0.48	7560.76 ± 0.22
		WN18RR3	9762.34 ± 1394.54	8.67 ± 1.25	0.08 ± 0.00	16359.06 ± 0.48	7560.76 ± 0.22

Table 31: MAGNN: node classification results (IMDb, Freebase). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Accuracy	Precision	Recall	Micro-F1	Macro-F1
IMDb	Original	IMDb1	0.6248 ± 0.0073	0.6345 ± 0.0142	0.6215 ± 0.0068	0.6248 ± 0.0073	0.6239 ± 0.0067
		IMDb2	0.6272 ± 0.0065	0.6353 ± 0.0018	0.6262 ± 0.0074	0.6272 ± 0.0065	0.6280 ± 0.0071
		IMDb3	0.6192 ± 0.0015	0.6244 ± 0.0042	0.6201 ± 0.0016	0.6192 ± 0.0015	0.6192 ± 0.0024
		IMDb4	0.6168 ± 0.0176	0.6206 ± 0.0166	0.6168 ± 0.0182	0.6168 ± 0.0176	0.6180 ± 0.0176
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}} \text{IMDb}_i$	$\bigcup_i \text{IMDb}_i$	0.6320 ± 0.0060	0.6398 ± 0.0070	0.6300 ± 0.0065	0.6320 ± 0.0060	0.6321 ± 0.0064
	Augmentation	IMDb1	0.6248 ± 0.0143	0.6314 ± 0.0179	0.6240 ± 0.0146	0.6248 ± 0.0143	0.6256 ± 0.0154
		IMDb2	0.6100 ± 0.0076	0.6188 ± 0.0102	0.6086 ± 0.0074	0.6100 ± 0.0076	0.6110 ± 0.0074
		IMDb3	0.6248 ± 0.0174	0.6301 ± 0.0187	0.6242 ± 0.0180	0.6248 ± 0.0174	0.6260 ± 0.0182
		IMDb4	0.6172 ± 0.0093	0.6262 ± 0.0137	0.6152 ± 0.0091	0.6172 ± 0.0093	0.6172 ± 0.0096
	Invariant + Union	IMDb1	0.6308 ± 0.0117	0.6376 ± 0.0117	0.6297 ± 0.0118	0.6308 ± 0.0117	0.6325 ± 0.0122
		IMDb2	0.6308 ± 0.0117	0.6376 ± 0.0117	0.6297 ± 0.0118	0.6308 ± 0.0117	0.6325 ± 0.0122
		IMDb3	0.6308 ± 0.0117	0.6376 ± 0.0117	0.6297 ± 0.0118	0.6308 ± 0.0117	0.6325 ± 0.0122
		IMDb4	0.6308 ± 0.0117	0.6376 ± 0.0117	0.6297 ± 0.0118	0.6308 ± 0.0117	0.6325 ± 0.0122
	Invariant + Intersect	IMDb1	0.6248 ± 0.0073	0.6345 ± 0.0142	0.6215 ± 0.0068	0.6248 ± 0.0073	0.6239 ± 0.0067
		IMDb2	0.6248 ± 0.0073	0.6345 ± 0.0142	0.6215 ± 0.0068	0.6248 ± 0.0073	0.6239 ± 0.0067
		IMDb3	0.6248 ± 0.0073	0.6345 ± 0.0142	0.6215 ± 0.0068	0.6248 ± 0.0073	0.6239 ± 0.0067
		IMDb4	0.6248 ± 0.0073	0.6345 ± 0.0142	0.6215 ± 0.0068	0.6248 ± 0.0073	0.6239 ± 0.0067
Freebase	Original	Freebase1	0.6135 ± 0.0103	0.5952 ± 0.0618	0.3370 ± 0.0116	0.6135 ± 0.0103	0.3760 ± 0.0174
		Freebase2	0.6059 ± 0.0037	0.5847 ± 0.0832	0.3171 ± 0.0006	0.6059 ± 0.0037	0.3423 ± 0.0048
		Freebase3	0.6036 ± 0.0008	0.5514 ± 0.0365	0.3147 ± 0.0106	0.6036 ± 0.0008	0.3403 ± 0.0158
	Universal Graph $\bigcup_{i\in\{1,2,3\}} \text{Freebase}_i$	$\bigcup_i \text{Freebase}_i$	0.6021 ± 0.0062	0.6218 ± 0.0562	0.3111 ± 0.0120	0.6021 ± 0.0062	0.3385 ± 0.0151
	Augmentation	Freebase1	0.6107 ± 0.0028	0.6020 ± 0.0543	0.3249 ± 0.0113	0.6107 ± 0.0028	0.3527 ± 0.0172
		Freebase2	0.6103 ± 0.0019	0.6014 ± 0.0540	0.3249 ± 0.0113	0.6103 ± 0.0019	0.3526 ± 0.0171
		Freebase3	0.6099 ± 0.0031	0.6019 ± 0.0547	0.3245 ± 0.0116	0.6099 ± 0.0031	0.3523 ± 0.0177
	Invariant + Union	Freebase1	0.6107 ± 0.0093	0.5789 ± 0.0396	0.3241 ± 0.0272	0.6107 ± 0.0093	0.3511 ± 0.0296
		Freebase2	0.6107 ± 0.0093	0.5789 ± 0.0396	0.3241 ± 0.0272	0.6107 ± 0.0093	0.3511 ± 0.0296
		Freebase3	0.6107 ± 0.0093	0.5789 ± 0.0396	0.3241 ± 0.0272	0.6107 ± 0.0093	0.3511 ± 0.0296
	Invariant + Intersect	Freebase1	0.6021 ± 0.0062	0.6218 ± 0.0562	0.3111 ± 0.0120	0.6021 ± 0.0062	0.3385 ± 0.0151
		Freebase2	0.6021 ± 0.0062	0.6218 ± 0.0562	0.3111 ± 0.0120	0.6021 ± 0.0062	0.3385 ± 0.0151
		Freebase3	0.6021 ± 0.0062	0.6218 ± 0.0562	0.3111 ± 0.0120	0.6021 ± 0.0062	0.3385 ± 0.0151

Table 32: MAGNN: node classification degree of invariance (IMDb, Freebase). Mean $\pm$ standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ
IMDb	Original	IMDb1 vs. IMDb2	0.6871 ± 0.0269
		IMDb1 vs. IMDb3	0.6837 ± 0.0205
		IMDb1 vs. IMDb4	0.8033 ± 0.0787
		IMDb2 vs. IMDb3	0.7105 ± 0.0192
		IMDb2 vs. IMDb4	0.7259 ± 0.0900
		IMDb3 vs. IMDb4	0.7108 ± 0.0226
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	IMDb1 vs. IMDb2	1.000 ± 0.0000 (trivial)
		IMDb1 vs. IMDb3	
		IMDb1 vs. IMDb4	
		IMDb2 vs. IMDb3	
		IMDb2 vs. IMDb4	
		IMDb3 vs. IMDb4	
	Augmentation	IMDb1 vs. IMDb2	0.9208 ± 0.0151
		IMDb1 vs. IMDb3	0.9591 ± 0.0075
		IMDb1 vs. IMDb4	0.9423 ± 0.0075
		IMDb2 vs. IMDb3	0.9362 ± 0.0116
		IMDb2 vs. IMDb4	0.9556 ± 0.0103
		IMDb3 vs. IMDb4	0.9317 ± 0.0163
	Invariant + Union	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
	Invariant + Intersect	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
Freebase	Original	Freebase1 vs. Freebase2	0.6695 ± 0.0774
		Freebase1 vs. Freebase3	0.7149 ± 0.0317
		Freebase2 vs. Freebase3	0.7342 ± 0.0431
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Augmentation	Freebase1 vs. Freebase2	0.9960 ± 0.0006
		Freebase1 vs. Freebase3	0.9945 ± 0.0015
		Freebase2 vs. Freebase3	0.9945 ± 0.0016
	Invariant + Union	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000
	Invariant + Intersect	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000

Table 33: MAGNN: node classification efficiency (IMDb, Freebase). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb	Original	IMDb1	108.07 ± 6.36	6.33 ± 0.47	3.94 ± 0.00	319.13 ± 0.16	320.10 ± 0.24
		IMDb2	217.29 ± 24.05	10.33 ± 1.25	3.94 ± 0.00	310.50 ± 0.09	311.20 ± 0.56
		IMDb3	178.28 ± 11.81	9.33 ± 0.47	3.94 ± 0.00	311.92 ± 0.14	312.84 ± 0.06
		IMDb4	139.57 ± 26.21	7.33 ± 1.25	3.94 ± 0.00	317.62 ± 0.16	318.20 ± 0.51
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	242.31 ± 26.55	8.67 ± 0.94	3.94 ± 0.00	320.02 ± 0.22	319.84 ± 0.72
	Augmentation	IMDb1-4	242.27 ± 48.17	9.33 ± 1.89	3.94 ± 0.00	317.47 ± 0.17	319.28 ± 0.52
	Invariant + Union	IMDb1	258.73 ± 39.81	7.67 ± 1.25	3.95 ± 0.00	351.51 ± 0.02	354.54 ± 1.68
		IMDb2	257.52 ± 42.38	7.67 ± 1.25	3.95 ± 0.00	351.51 ± 0.02	354.54 ± 1.68
		IMDb3	261.98 ± 45.30	7.67 ± 1.25	3.95 ± 0.00	351.51 ± 0.02	354.54 ± 1.68
		IMDb4	263.29 ± 47.86	7.67 ± 1.25	3.95 ± 0.00	351.51 ± 0.02	354.54 ± 1.68
	Invariant + Intersect	IMDb1	106.39 ± 8.29	6.33 ± 0.47	3.94 ± 0.00	319.13 ± 0.16	320.10 ± 0.24
		IMDb2	106.43 ± 8.06	6.33 ± 0.47	3.94 ± 0.00	319.13 ± 0.16	320.10 ± 0.24
		IMDb3	106.35 ± 7.75	6.33 ± 0.47	3.94 ± 0.00	319.13 ± 0.16	320.10 ± 0.24
		IMDb4	105.82 ± 8.05	6.33 ± 0.47	3.94 ± 0.00	319.13 ± 0.16	320.10 ± 0.24
Freebase	Original	Freebase1	716.18 ± 146.74	11.67 ± 2.49	44.25 ± 0.00	417.83 ± 0.58	407.51 ± 1.50
		Freebase2	1328.24 ± 228.08	14.67 ± 2.49	44.26 ± 0.00	461.08 ± 0.86	439.11 ± 1.03
		Freebase3	1680.59 ± 388.36	11.33 ± 2.49	44.28 ± 0.00	571.04 ± 0.33	537.83 ± 0.90
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	1736.84 ± 390.31	10.00 ± 2.16	44.29 ± 0.00	610.11 ± 0.80	574.75 ± 2.64
	Augmentation	Freebase1-3	2616.95 ± 1292.73	18.00 ± 8.49	44.29 ± 0.00	569.76 ± 2.49	543.75 ± 3.12
	Invariant + Union	Freebase1	5135.86 ± 1496.35	13.33 ± 3.68	44.34 ± 0.00	800.82 ± 1.44	740.83 ± 2.66
		Freebase2	5090.19 ± 1428.75	13.33 ± 3.68	44.34 ± 0.00	800.82 ± 1.44	740.83 ± 2.66
		Freebase3	5298.55 ± 1567.59	13.33 ± 3.68	44.34 ± 0.00	800.82 ± 1.44	740.83 ± 2.66
	Invariant + Intersect	Freebase1	1704.41 ± 371.07	10.00 ± 2.16	44.29 ± 0.00	610.11 ± 0.80	574.75 ± 2.64
		Freebase2	1741.65 ± 398.96	10.00 ± 2.16	44.29 ± 0.00	610.11 ± 0.80	574.75 ± 2.64
		Freebase3	1741.68 ± 380.78	10.00 ± 2.16	44.29 ± 0.00	610.11 ± 0.80	574.75 ± 2.64

Table 34: MAGNN: link prediction results (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Precision	Recall	F1	Hits@1	Hits@3	MRR
IMDb (Movie–Director)	Original	IMDb1	0.0867 ± 0.0012	0.7755 ± 0.0295	0.1560 ± 0.0014	0.1077 ± 0.0068	0.2763 ± 0.0122	0.2846 ± 0.0055
		IMDb3	0.0867 ± 0.0008	0.7775 ± 0.0096	0.1560 ± 0.0011	0.1100 ± 0.0080	0.2747 ± 0.0066	0.2852 ± 0.0037
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.0873 ± 0.0009	0.7683 ± 0.0215	0.1568 ± 0.0010	0.1172 ± 0.0076	0.2855 ± 0.0083	0.2922 ± 0.0055
	Augmentation	IMDb1	0.0874 ± 0.0007	0.7751 ± 0.0167	0.1571 ± 0.0015	0.1089 ± 0.0043	0.2751 ± 0.0117	0.2850 ± 0.0043
		IMDb3	0.0875 ± 0.0007	0.7751 ± 0.0167	0.1572 ± 0.0014	0.1089 ± 0.0043	0.2751 ± 0.0117	0.2847 ± 0.0042
	Invariant + Union	IMDb1	0.0822 ± 0.0008	0.7500 ± 0.0142	0.1481 ± 0.0011	0.1085 ± 0.0031	0.2727 ± 0.0096	0.2845 ± 0.0045
		IMDb3	0.0822 ± 0.0008	0.7500 ± 0.0142	0.1481 ± 0.0011	0.1085 ± 0.0031	0.2727 ± 0.0096	0.2845 ± 0.0045
	Invariant + Intersect	IMDb1	0.0846 ± 0.0043	0.7476 ± 0.0138	0.1519 ± 0.0070	0.1085 ± 0.0151	0.2683 ± 0.0205	0.2811 ± 0.0120
		IMDb3	0.0846 ± 0.0043	0.7476 ± 0.0138	0.1519 ± 0.0070	0.1085 ± 0.0151	0.2683 ± 0.0205	0.2811 ± 0.0120
DBLP	Original	DBLP1	0.0635 ± 0.0450	0.4754 ± 0.3366	0.1120 ± 0.0793	0.1362 ± 0.0000	0.3468 ± 0.0076	0.3181 ± 0.0048
		DBLP2	0.0995 ± 0.0151	0.5874 ± 0.2378	0.1605 ± 0.0044	0.1267 ± 0.0134	0.3448 ± 0.0087	0.3138 ± 0.0071
		DBLP3	0.1611 ± 0.0444	0.6978 ± 0.0550	0.2582 ± 0.0606	0.2277 ± 0.0687	0.5291 ± 0.1606	0.4237 ± 0.0929
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	0.1255 ± 0.0346	0.6887 ± 0.0025	0.2104 ± 0.0483	0.2058 ± 0.0532	0.4854 ± 0.1030	0.4014 ± 0.0631
	Augmentation	DBLP1	0.0306 ± 0.0433	0.2449 ± 0.3464	0.0544 ± 0.0770	0.1207 ± 0.0219	0.2941 ± 0.0282	0.2868 ± 0.0265
		DBLP2	0.0306 ± 0.0433	0.2449 ± 0.3464	0.0544 ± 0.0770	0.1207 ± 0.0219	0.2941 ± 0.0282	0.2868 ± 0.0265
		DBLP3	0.0306 ± 0.0433	0.2449 ± 0.3464	0.0544 ± 0.0770	0.1207 ± 0.0219	0.2941 ± 0.0282	0.2869 ± 0.0264
	Invariant + Union	DBLP1	0.1255 ± 0.0346	0.6887 ± 0.0025	0.2104 ± 0.0483	0.2058 ± 0.0532	0.4854 ± 0.1030	0.4014 ± 0.0631
		DBLP2	0.1255 ± 0.0346	0.6887 ± 0.0025	0.2104 ± 0.0483	0.2058 ± 0.0532	0.4854 ± 0.1030	0.4014 ± 0.0631
		DBLP3	0.1255 ± 0.0346	0.6887 ± 0.0025	0.2104 ± 0.0483	0.2058 ± 0.0532	0.4854 ± 0.1030	0.4014 ± 0.0631
	Invariant + Intersect	DBLP1	0.0635 ± 0.0450	0.4754 ± 0.3366	0.1120 ± 0.0793	0.1362 ± 0.0000	0.3468 ± 0.0076	0.3181 ± 0.0048
		DBLP2	0.0635 ± 0.0450	0.4754 ± 0.3366	0.1120 ± 0.0793	0.1362 ± 0.0000	0.3468 ± 0.0076	0.3181 ± 0.0048
		DBLP3	0.0635 ± 0.0450	0.4754 ± 0.3366	0.1120 ± 0.0793	0.1362 ± 0.0000	0.3468 ± 0.0076	0.3181 ± 0.0048

Table 35: MAGNN: link prediction degree of invariance (IMDb (Movie–Director), DBLP, WN18RR). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ	Kendall-τ@1	Kendall-τ@3
IMDb (Movie–Director)	Original	IMDb1 vs. IMDb3	0.7646 ± 0.0227	0.4620 ± 0.0306	0.5805 ± 0.0507
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	IMDb1 vs. IMDb3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	IMDb1 vs. IMDb3	0.9972 ± 0.0039	1.0000 ± 0.0000	0.9940 ± 0.0085
	Invariant + Union	IMDb1 vs. IMDb3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
	Invariant + Intersect	IMDb1 vs. IMDb3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
DBLP	Original	DBLP1 vs. DBLP2	0.6715 ± 0.0053	0.6143 ± 0.5320	0.6911 ± 0.2185
		DBLP1 vs. DBLP3	0.3992 ± 0.1518	0.3266 ± 0.3439	0.2758 ± 0.1901
		DBLP2 vs. DBLP3	0.3865 ± 0.1025	0.0569 ± 0.3326	0.2398 ± 0.2053
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	DBLP1 vs. DBLP2 DBLP1 vs. DBLP3 DBLP2 vs. DBLP3	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)	1.000 ± 0.0000 (trivial)
	Augmentation	DBLP1 vs. DBLP2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP1 vs. DBLP3	0.9985 ± 0.0021	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP2 vs. DBLP3	0.9985 ± 0.0021	1.0000 ± 0.0000	1.0000 ± 0.0000
	Invariant + Union	DBLP1 vs. DBLP2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP1 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP2 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
	Invariant + Intersect	DBLP1 vs. DBLP2	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP1 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
		DBLP2 vs. DBLP3	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000

Table 36: MAGNN: link prediction efficiency (IMDb (Movie–Director), DBLP, WN18RR). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb (Movie–Director)	Original	IMDb1	1026.61 ± 117.68	26.67 ± 3.30	6.00 ± 0.00	2627.84 ± 29.70	1260.06 ± 0.35
		IMDb3	762.92 ± 124.22	19.00 ± 3.27	6.00 ± 0.00	2595.40 ± 25.85	1258.39 ± 0.76
	Universal Graph $\bigcup_{i\in\{1,3\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	1351.86 ± 383.20	28.33 ± 7.76	6.00 ± 0.00	2963.13 ± 29.24	1350.82 ± 0.54
	Augmentation	IMDb1,3	817.43 ± 181.02	19.33 ± 4.11	6.00 ± 0.00	2597.52 ± 9.79	1261.78 ± 0.89
	Invariant + Union	IMDb1	1008.42 ± 49.78	12.33 ± 0.47	6.02 ± 0.00	4599.08 ± 12.44	1685.47 ± 0.43
		IMDb3	976.46 ± 28.64	12.33 ± 0.47	6.02 ± 0.00	4599.08 ± 12.44	1685.47 ± 0.43
	Invariant + Intersect	IMDb1	633.68 ± 102.51	14.67 ± 2.49	6.00 ± 0.00	2638.20 ± 9.90	1267.58 ± 0.13
		IMDb3	638.51 ± 99.69	14.67 ± 2.49	6.00 ± 0.00	2638.20 ± 9.90	1267.58 ± 0.13
DBLP	Original	DBLP1	5691.53 ± 686.30	22.67 ± 2.87	8.70 ± 0.00	5192.90 ± 13.31	1695.21 ± 0.23
		DBLP2	3714.62 ± 630.10	14.00 ± 2.45	8.70 ± 0.00	5191.11 ± 11.68	1696.31 ± 0.61
		DBLP3	6896.63 ± 3443.19	21.00 ± 10.80	8.70 ± 0.00	6744.16 ± 9.87	1975.12 ± 0.36
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ DBLP$_i$	$\bigcup_i$ DBLP$_i$	6638.88 ± 2011.11	18.00 ± 5.35	8.71 ± 0.00	6793.44 ± 11.77	1978.79 ± 0.37
	Augmentation	DBLP1-3	5749.33 ± 380.38	19.00 ± 1.41	8.71 ± 0.00	6783.26 ± 14.30	1979.49 ± 0.84
	Invariant + Union	DBLP1	6670.62 ± 2020.68	18.00 ± 5.35	8.71 ± 0.00	6793.44 ± 11.77	1978.79 ± 0.37
		DBLP2	6651.70 ± 1991.57	18.00 ± 5.35	8.71 ± 0.00	6793.44 ± 11.77	1978.79 ± 0.37
		DBLP3	6744.52 ± 2025.72	18.00 ± 5.35	8.71 ± 0.00	6793.44 ± 11.77	1978.79 ± 0.37
	Invariant + Intersect	DBLP1	5768.56 ± 835.67	22.67 ± 2.87	8.70 ± 0.00	5192.90 ± 13.31	1695.21 ± 0.23
		DBLP2	5801.41 ± 791.96	22.67 ± 2.87	8.70 ± 0.00	5192.90 ± 13.31	1695.21 ± 0.23
		DBLP3	5741.51 ± 750.86	22.67 ± 2.87	8.70 ± 0.00	5192.90 ± 13.31	1695.21 ± 0.23

Table 37: SeHGNN: node classification results (IMDb, Freebase). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Accuracy	Precision	Recall	Micro-F1	Macro-F1
IMDb	Original	IMDb1	0.6336 ± 0.0039	0.6335 ± 0.0038	0.6352 ± 0.0028	0.6336 ± 0.0039	0.6339 ± 0.0031
		IMDb2	0.6388 ± 0.0157	0.6453 ± 0.0143	0.6369 ± 0.0170	0.6388 ± 0.0157	0.6380 ± 0.0172
		IMDb3	0.6356 ± 0.0187	0.6417 ± 0.0165	0.6341 ± 0.0201	0.6356 ± 0.0187	0.6358 ± 0.0204
		IMDb4	0.6455 ± 0.0189	0.6473 ± 0.0206	0.6465 ± 0.0182	0.6455 ± 0.0189	0.6461 ± 0.0197
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.6431 ± 0.0039	0.6456 ± 0.0052	0.6433 ± 0.0026	0.6431 ± 0.0039	0.6435 ± 0.0037
	Augmentation	IMDb1	0.6256 ± 0.0134	0.6268 ± 0.0127	0.6263 ± 0.0144	0.6256 ± 0.0134	0.6255 ± 0.0150
		IMDb2	0.6308 ± 0.0113	0.6319 ± 0.0101	0.6318 ± 0.0122	0.6308 ± 0.0113	0.6309 ± 0.0119
		IMDb3	0.6256 ± 0.0147	0.6270 ± 0.0141	0.6258 ± 0.0156	0.6256 ± 0.0147	0.6251 ± 0.0162
		IMDb4	0.6284 ± 0.0123	0.6295 ± 0.0121	0.6299 ± 0.0128	0.6284 ± 0.0123	0.6288 ± 0.0135
	Invariant	IMDb1	0.6535 ± 0.0064	0.6559 ± 0.0068	0.6544 ± 0.0070	0.6535 ± 0.0064	0.6549 ± 0.0069
		IMDb2	0.6535 ± 0.0064	0.6559 ± 0.0068	0.6544 ± 0.0070	0.6535 ± 0.0064	0.6549 ± 0.0069
		IMDb3	0.6535 ± 0.0064	0.6559 ± 0.0068	0.6544 ± 0.0070	0.6535 ± 0.0064	0.6549 ± 0.0069
		IMDb4	0.6535 ± 0.0064	0.6559 ± 0.0068	0.6544 ± 0.0070	0.6535 ± 0.0064	0.6549 ± 0.0069
Freebase	Original	Freebase1	0.6482 ± 0.0117	0.5866 ± 0.0644	0.4581 ± 0.0170	0.6482 ± 0.0117	0.4735 ± 0.0176
		Freebase2	0.6394 ± 0.0054	0.5826 ± 0.0519	0.4577 ± 0.0035	0.6394 ± 0.0054	0.4620 ± 0.0119
		Freebase3	0.6336 ± 0.0072	0.5785 ± 0.0471	0.4542 ± 0.0044	0.6336 ± 0.0072	0.4607 ± 0.0108
	Original + relation-aware	Freebase1	0.6681 ± 0.0000	0.5972 ± 0.0472	0.4646 ± 0.0238	0.6681 ± 0.0000	0.4917 ± 0.0209
		Freebase2	0.6759 ± 0.0063	0.5762 ± 0.0176	0.4663 ± 0.0095	0.6759 ± 0.0063	0.4999 ± 0.0066
		Freebase3	0.6671 ± 0.0103	0.5887 ± 0.0588	0.4628 ± 0.0103	0.6671 ± 0.0103	0.4918 ± 0.0130
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	0.6359 ± 0.0085	0.5756 ± 0.0463	0.4512 ± 0.0079	0.6359 ± 0.0085	0.4599 ± 0.0122
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$ + relation-aware	$\bigcup_i$ Freebase$_i$	0.6587 ± 0.0041	0.6524 ± 0.0958	0.4338 ± 0.0069	0.6587 ± 0.0041	0.4662 ± 0.0063
	Augmentation	Freebase1	0.6388 ± 0.0040	0.5292 ± 0.0447	0.4206 ± 0.0196	0.6388 ± 0.0040	0.4350 ± 0.0128
		Freebase2	0.6407 ± 0.0047	0.5342 ± 0.0333	0.4184 ± 0.0249	0.6407 ± 0.0047	0.4344 ± 0.0208
		Freebase3	0.6361 ± 0.0076	0.5291 ± 0.0310	0.4085 ± 0.0201	0.6361 ± 0.0076	0.4267 ± 0.0199
	Augmentation + relation-aware	Freebase1	0.6537 ± 0.0147	0.5254 ± 0.0386	0.4849 ± 0.0091	0.6537 ± 0.0147	0.4911 ± 0.0117
		Freebase2	0.6541 ± 0.0192	0.5589 ± 0.0688	0.4581 ± 0.0249	0.6541 ± 0.0192	0.4774 ± 0.0175
		Freebase3	0.6537 ± 0.0099	0.5312 ± 0.0063	0.4839 ± 0.0162	0.6537 ± 0.0099	0.4957 ± 0.0135
	Invariant	Freebase1	0.6359 ± 0.0085	0.5756 ± 0.0463	0.4512 ± 0.0079	0.6359 ± 0.0085	0.4599 ± 0.0122
		Freebase2	0.6359 ± 0.0085	0.5756 ± 0.0463	0.4512 ± 0.0079	0.6359 ± 0.0085	0.4599 ± 0.0122
		Freebase3	0.6359 ± 0.0085	0.5756 ± 0.0463	0.4512 ± 0.0079	0.6359 ± 0.0085	0.4599 ± 0.0122
	Invariant + relation-aware	Freebase1	0.6637 ± 0.0080	0.5822 ± 0.0480	0.4480 ± 0.0277	0.6637 ± 0.0080	0.4745 ± 0.0169
		Freebase2	0.6637 ± 0.0080	0.5822 ± 0.0480	0.4480 ± 0.0277	0.6637 ± 0.0080	0.4745 ± 0.0169
		Freebase3	0.6637 ± 0.0080	0.5822 ± 0.0480	0.4480 ± 0.0277	0.6637 ± 0.0080	0.4745 ± 0.0169

Table 38: SeHGNN: node classification degree of invariance (IMDb, Freebase). Mean $\pm$ standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ
IMDb	Original	IMDb1 vs. IMDb2	0.6653 ± 0.0496
		IMDb1 vs. IMDb3	0.6334 ± 0.0473
		IMDb1 vs. IMDb4	0.6433 ± 0.0395
		IMDb2 vs. IMDb3	0.6007 ± 0.0368
		IMDb2 vs. IMDb4	0.5978 ± 0.0304
		IMDb3 vs. IMDb4	0.7382 ± 0.0301
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	IMDb1 vs. IMDb2	1.000 ± 0.0000 (trivial)
		IMDb1 vs. IMDb3	
		IMDb1 vs. IMDb4	
		IMDb2 vs. IMDb3	
		IMDb2 vs. IMDb4	
		IMDb3 vs. IMDb4	
	Augmentation	IMDb1 vs. IMDb2	0.9256 ± 0.0197
		IMDb1 vs. IMDb3	0.9567 ± 0.0030
		IMDb1 vs. IMDb4	0.9375 ± 0.0139
		IMDb2 vs. IMDb3	0.9258 ± 0.0260
		IMDb2 vs. IMDb4	0.9556 ± 0.0053
		IMDb3 vs. IMDb4	0.9181 ± 0.0075
	Invariant	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
Freebase	Original	Freebase1 vs. Freebase2	0.7877 ± 0.0510
		Freebase1 vs. Freebase3	0.7058 ± 0.0321
		Freebase2 vs. Freebase3	0.6986 ± 0.0450
	Original + relation-aware	Freebase1 vs. Freebase2	0.5410 ± 0.1116
		Freebase1 vs. Freebase3	0.4948 ± 0.0288
		Freebase2 vs. Freebase3	0.5348 ± 0.0393
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$ + relation-aware	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Augmentation	Freebase1 vs. Freebase2	0.9432 ± 0.0084
		Freebase1 vs. Freebase3	0.8645 ± 0.0202
		Freebase2 vs. Freebase3	0.8752 ± 0.0213
	Augmentation + relation-aware	Freebase1 vs. Freebase2	0.8310 ± 0.0710
		Freebase1 vs. Freebase3	0.7893 ± 0.0202
		Freebase2 vs. Freebase3	0.6839 ± 0.0275
	Invariant	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000
	Invariant + relation-aware	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000

Table 39: SeHGNN: node classification efficiency (IMDb, Freebase). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb	Original	IMDb1	111.34 ± 8.53	95.67 ± 7.59	217.72 ± 0.00	7560.26 ± 0.00	4054.34 ± 0.38
		IMDb2	74.05 ± 17.27	101.67 ± 25.49	139.35 ± 0.00	4587.53 ± 0.00	2627.56 ± 0.46
		IMDb3	75.57 ± 6.09	84.00 ± 7.48	169.50 ± 0.00	5718.25 ± 0.00	3150.51 ± 0.00
		IMDb4	76.55 ± 4.10	86.67 ± 5.56	169.50 ± 0.00	5716.87 ± 0.00	3151.04 ± 0.41
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	210.09 ± 18.07	87.00 ± 7.79	428.60 ± 0.00	16839.21 ± 0.00	9336.17 ± 0.74
	Augmentation	IMDb1-4	201.88 ± 9.69	101.33 ± 4.99	271.95 ± 0.00	13070.23 ± 0.00	6951.17 ± 0.00
	Invariant	IMDb1	64.60 ± 1.84	88.33 ± 2.62	139.52 ± 0.00	4638.15 ± 0.00	2518.54 ± 0.61
		IMDb2	64.10 ± 2.33	88.33 ± 2.62	139.52 ± 0.00	4638.15 ± 0.00	2518.54 ± 0.61
		IMDb3	64.85 ± 1.77	88.33 ± 2.62	139.52 ± 0.00	4638.15 ± 0.00	2518.54 ± 0.61
		IMDb4	64.26 ± 2.17	88.33 ± 2.62	139.52 ± 0.00	4638.15 ± 0.00	2518.54 ± 0.61
Freebase	Original	Freebase1	69.37 ± 0.42	55.00 ± 0.82	232.19 ± 0.00	5951.31 ± 0.00	1827.55 ± 1.02
		Freebase2	72.84 ± 3.68	57.00 ± 2.94	232.19 ± 0.00	6137.87 ± 0.00	2031.25 ± 1.01
		Freebase3	72.16 ± 1.81	55.00 ± 1.63	232.19 ± 0.00	6737.91 ± 0.00	2691.52 ± 1.53
	Original + relation-aware	Freebase1	101.66 ± 22.81	39.00 ± 1.41	245.80 ± 0.00	2593.44 ± 0.57	1951.28 ± 0.88
		Freebase2	146.79 ± 5.64	38.33 ± 0.47	297.20 ± 0.00	4072.20 ± 0.50	2961.73 ± 2.57
		Freebase3	332.38 ± 53.38	41.00 ± 0.00	511.86 ± 0.00	11717.90 ± 1.34	8734.57 ± 6.14
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	73.90 ± 1.55	56.33 ± 1.25	232.19 ± 0.00	6736.06 ± 0.00	2692.27 ± 0.28
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$ + relation-aware	$\bigcup_i$ Freebase$_i$	370.31 ± 30.61	40.67 ± 0.47	628.26 ± 0.00	16659.01 ± 2.89	12126.27 ± 6.14
	Augmentation	Freebase1-3	43.00 ± 2.13	55.00 ± 3.74	232.19 ± 0.00	9551.41 ± 2.74	2894.12 ± 163.65
	Augmentation + relation-aware	Freebase1-3	274.06 ± 54.26	43.00 ± 3.74	563.26 ± 0.00	17656.68 ± 3.27	10540.74 ± 89.76
	Invariant	Freebase1	73.56 ± 1.19	56.33 ± 1.25	232.19 ± 0.00	6736.06 ± 0.00	2692.27 ± 0.28
		Freebase2	73.86 ± 0.53	56.33 ± 1.25	232.19 ± 0.00	6736.06 ± 0.00	2692.27 ± 0.28
		Freebase3	73.29 ± 1.55	56.33 ± 1.25	232.19 ± 0.00	6736.06 ± 0.00	2692.27 ± 0.28
	Invariant + relation-aware	Freebase1	312.77 ± 14.03	41.67 ± 1.70	563.26 ± 0.00	13641.85 ± 2.38	10119.25 ± 2.88
		Freebase2	312.97 ± 14.41	41.67 ± 1.70	563.26 ± 0.00	13641.85 ± 2.38	10119.25 ± 2.88
		Freebase3	311.82 ± 12.67	41.67 ± 1.70	563.26 ± 0.00	13641.85 ± 2.38	10119.25 ± 2.88

Table 40: PAIN: node classification results (IMDb, Freebase). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Accuracy	Precision	Recall	Micro-F1	Macro-F1
IMDb	Original	IMDb1	0.5606 ± 0.0071	0.5689 ± 0.0077	0.5554 ± 0.0060	0.5606 ± 0.0071	0.5581 ± 0.0057
		IMDb2	0.5881 ± 0.0193	0.5946 ± 0.0148	0.5855 ± 0.0235	0.5881 ± 0.0193	0.5872 ± 0.0221
		IMDb3	0.5913 ± 0.0100	0.6000 ± 0.0045	0.5887 ± 0.0118	0.5913 ± 0.0100	0.5911 ± 0.0098
		IMDb4	0.5550 ± 0.0068	0.5583 ± 0.0064	0.5503 ± 0.0084	0.5550 ± 0.0068	0.5514 ± 0.0076
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.5670 ± 0.0054	0.5852 ± 0.0062	0.5602 ± 0.0077	0.5670 ± 0.0054	0.5644 ± 0.0068
	Augmentation	IMDb1	0.4043 ± 0.0277	0.4354 ± 0.0260	0.4058 ± 0.0200	0.4043 ± 0.0277	0.3745 ± 0.0382
		IMDb2	0.4725 ± 0.0020	0.4903 ± 0.0171	0.4752 ± 0.0075	0.4725 ± 0.0020	0.4613 ± 0.0112
		IMDb3	0.4892 ± 0.0138	0.4993 ± 0.0199	0.4902 ± 0.0166	0.4892 ± 0.0138	0.4831 ± 0.0143
		IMDb4	0.3824 ± 0.0189	0.4545 ± 0.0662	0.3784 ± 0.0090	0.3824 ± 0.0189	0.3381 ± 0.0210
	Invariant	IMDb1	0.5670 ± 0.0054	0.5852 ± 0.0062	0.5602 ± 0.0077	0.5670 ± 0.0054	0.5644 ± 0.0068
		IMDb2	0.5670 ± 0.0054	0.5852 ± 0.0062	0.5602 ± 0.0077	0.5670 ± 0.0054	0.5644 ± 0.0068
		IMDb3	0.5670 ± 0.0054	0.5852 ± 0.0062	0.5602 ± 0.0077	0.5670 ± 0.0054	0.5644 ± 0.0068
		IMDb4	0.5670 ± 0.0054	0.5852 ± 0.0062	0.5602 ± 0.0077	0.5670 ± 0.0054	0.5644 ± 0.0068
Freebase	Original	Freebase1	0.6285 ± 0.0098	0.5351 ± 0.0305	0.4730 ± 0.0328	0.6285 ± 0.0098	0.4866 ± 0.0241
		Freebase2	0.5918 ± 0.0118	0.4792 ± 0.0354	0.4327 ± 0.0168	0.5918 ± 0.0118	0.4380 ± 0.0196
		Freebase3	0.5563 ± 0.0138	0.4253 ± 0.0402	0.3938 ± 0.0014	0.5563 ± 0.0138	0.3910 ± 0.0151
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	0.5929 ± 0.0131	0.4800 ± 0.0269	0.4281 ± 0.0146	0.5929 ± 0.0131	0.4479 ± 0.0199
	Augmentation	Freebase1	0.6151 ± 0.0090	0.5186 ± 0.0274	0.4560 ± 0.0297	0.6151 ± 0.0090	0.4791 ± 0.0290
		Freebase2	0.5984 ± 0.0101	0.4849 ± 0.0280	0.4344 ± 0.0106	0.5984 ± 0.0101	0.4523 ± 0.0165
		Freebase3	0.5655 ± 0.0157	0.4358 ± 0.0381	0.3944 ± 0.0040	0.5655 ± 0.0157	0.4087 ± 0.0156
	Invariant	Freebase1	0.5929 ± 0.0131	0.4800 ± 0.0269	0.4281 ± 0.0146	0.5929 ± 0.0131	0.4479 ± 0.0199
		Freebase2	0.5929 ± 0.0131	0.4800 ± 0.0269	0.4281 ± 0.0146	0.5929 ± 0.0131	0.4479 ± 0.0199
		Freebase3	0.5929 ± 0.0131	0.4800 ± 0.0269	0.4281 ± 0.0146	0.5929 ± 0.0131	0.4479 ± 0.0199

Table 41: PAIN: node classification degree of invariance (IMDb, Freebase). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ
IMDb	Original	IMDb1 vs. IMDb2	0.4540 ± 0.0197
		IMDb1 vs. IMDb3	0.4474 ± 0.0482
		IMDb1 vs. IMDb4	0.4561 ± 0.0382
		IMDb2 vs. IMDb3	0.6037 ± 0.0041
		IMDb2 vs. IMDb4	0.5237 ± 0.0142
		IMDb3 vs. IMDb4	0.5053 ± 0.0094
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	IMDb1 vs. IMDb2	1.000 ± 0.0000 (trivial)
		IMDb1 vs. IMDb3	
		IMDb1 vs. IMDb4	
		IMDb2 vs. IMDb3	
		IMDb2 vs. IMDb4	
		IMDb3 vs. IMDb4	
	Augmentation	IMDb1 vs. IMDb2	0.2520 ± 0.0723
		IMDb1 vs. IMDb3	0.2961 ± 0.0773
		IMDb1 vs. IMDb4	0.5473 ± 0.2127
		IMDb2 vs. IMDb3	0.7065 ± 0.0691
		IMDb2 vs. IMDb4	0.3230 ± 0.0859
		IMDb3 vs. IMDb4	0.3044 ± 0.0647
	Invariant	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
Freebase	Original	Freebase1 vs. Freebase2	0.6565 ± 0.0386
		Freebase1 vs. Freebase3	0.5849 ± 0.0538
		Freebase2 vs. Freebase3	0.6639 ± 0.0288
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Augmentation	Freebase1 vs. Freebase2	0.6294 ± 0.0122
		Freebase1 vs. Freebase3	0.6266 ± 0.0170
		Freebase2 vs. Freebase3	0.6162 ± 0.0164
	Invariant	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000

Table 42: PAIN: node classification efficiency (IMDb, Freebase). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb	Original	IMDb1	3298.92 ± 1085.71	224.00 ± 73.79	3.71 ± 0.00	6128.08 ± 0.00	3896.62 ± 0.00
		IMDb2	3525.86 ± 617.47	239.67 ± 41.96	3.71 ± 0.00	6127.26 ± 0.00	3893.57 ± 0.00
		IMDb3	2261.01 ± 267.02	201.00 ± 23.76	3.71 ± 0.00	6036.11 ± 0.00	3837.80 ± 0.00
		IMDb4	3447.11 ± 895.16	307.33 ± 79.79	3.71 ± 0.00	6039.06 ± 0.00	3840.44 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	11992.93 ± 1472.07	204.33 ± 24.69	3.71 ± 0.00	6727.59 ± 0.00	4323.97 ± 0.00
	Augmentation	IMDb1-4	7208.65 ± 4464.36	461.33 ± 270.35	3.71 ± 0.00	6439.07 ± 0.16	4215.81 ± 0.25
	Invariant	IMDb1	11920.41 ± 1435.58	204.33 ± 24.69	3.71 ± 0.00	6727.59 ± 0.00	4323.97 ± 0.00
		IMDb2	11913.77 ± 1429.89	204.33 ± 24.69	3.71 ± 0.00	6727.59 ± 0.00	4323.97 ± 0.00
		IMDb3	11955.97 ± 1441.17	204.33 ± 24.69	3.71 ± 0.00	6727.59 ± 0.00	4323.97 ± 0.00
		IMDb4	12106.32 ± 1469.02	204.33 ± 24.69	3.71 ± 0.00	6727.59 ± 0.00	4323.97 ± 0.00
Freebase	Original	Freebase1	38840.98 ± 12256.71	110.00 ± 35.39	90.49 ± 0.00	29538.53 ± 0.00	2607.21 ± 0.09
		Freebase2	45431.78 ± 3746.42	121.67 ± 12.26	90.49 ± 0.00	31504.03 ± 0.00	2609.28 ± 0.05
		Freebase3	49523.06 ± 2410.78	126.50 ± 12.50	90.49 ± 0.00	33403.08 ± 0.00	2611.30 ± 0.03
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	48780.05 ± 10599.17	122.00 ± 28.33	90.49 ± 0.00	33434.42 ± 0.00	2611.27 ± 0.02
	Augmentation	Freebase1-3	59703.28 ± 8762.62	162.00 ± 23.37	90.49 ± 0.00	33403.08 ± 0.00	2611.27 ± 0.04
	Invariant	Freebase1	48340.40 ± 10964.39	122.00 ± 28.33	90.49 ± 0.00	33434.42 ± 0.00	2611.27 ± 0.02
		Freebase2	48705.95 ± 10546.50	122.00 ± 28.33	90.49 ± 0.00	33434.42 ± 0.00	2611.27 ± 0.02
		Freebase3	48588.25 ± 10537.68	122.00 ± 28.33	90.49 ± 0.00	33434.42 ± 0.00	2611.27 ± 0.02

Table 43: PathNet: node classification results (IMDb, Freebase). Mean ± standard deviation over three seeds.

Dataset	Method	Variant	Accuracy	Precision	Recall	Micro-F1	Macro-F1
IMDb	Original	IMDb1	0.5901 ± 0.0113	0.5969 ± 0.0123	0.5879 ± 0.0114	0.5901 ± 0.0113	0.5895 ± 0.0124
		IMDb2	0.5941 ± 0.0117	0.6010 ± 0.0143	0.5926 ± 0.0113	0.5941 ± 0.0117	0.5953 ± 0.0122
		IMDb3	0.5953 ± 0.0293	0.6011 ± 0.0299	0.5959 ± 0.0307	0.5953 ± 0.0293	0.5967 ± 0.0302
		IMDb4	0.5949 ± 0.0064	0.6006 ± 0.0079	0.5958 ± 0.0067	0.5949 ± 0.0064	0.5972 ± 0.0072
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	$\bigcup_i$ IMDb$_i$	0.5381 ± 0.0612	0.6142 ± 0.0193	0.5359 ± 0.0695	0.5381 ± 0.0612	0.5242 ± 0.0745
	Augmentation	IMDb1	0.5419 ± 0.0614	0.6219 ± 0.0222	0.5373 ± 0.0672	0.5419 ± 0.0614	0.5285 ± 0.0727
		IMDb2	0.5415 ± 0.0638	0.6251 ± 0.0203	0.5367 ± 0.0705	0.5415 ± 0.0638	0.5244 ± 0.0781
		IMDb3	0.5403 ± 0.0631	0.6226 ± 0.0188	0.5360 ± 0.0699	0.5403 ± 0.0631	0.5252 ± 0.0763
		IMDb4	0.5407 ± 0.0569	0.6157 ± 0.0217	0.5367 ± 0.0634	0.5407 ± 0.0569	0.5279 ± 0.0687
	Invariant	IMDb1	0.5985 ± 0.0114	0.6107 ± 0.0086	0.5961 ± 0.0103	0.5985 ± 0.0114	0.5991 ± 0.0107
		IMDb2	0.5985 ± 0.0114	0.6107 ± 0.0086	0.5961 ± 0.0103	0.5985 ± 0.0114	0.5991 ± 0.0107
		IMDb3	0.5985 ± 0.0114	0.6107 ± 0.0086	0.5961 ± 0.0103	0.5985 ± 0.0114	0.5991 ± 0.0107
		IMDb4	0.5985 ± 0.0114	0.6107 ± 0.0086	0.5961 ± 0.0103	0.5985 ± 0.0114	0.5991 ± 0.0107
Freebase	Original	Freebase1	0.5508 ± 0.0062	0.3347 ± 0.0356	0.3324 ± 0.0260	0.5508 ± 0.0062	0.2973 ± 0.0044
		Freebase2	0.5414 ± 0.0083	0.3953 ± 0.0697	0.3071 ± 0.0275	0.5414 ± 0.0083	0.2877 ± 0.0112
		Freebase3	0.4683 ± 0.0359	0.2183 ± 0.0118	0.2564 ± 0.0332	0.4683 ± 0.0359	0.2115 ± 0.0132
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	$\bigcup_i$ Freebase$_i$	0.4768 ± 0.0269	0.2350 ± 0.0205	0.2513 ± 0.0329	0.4768 ± 0.0269	0.2153 ± 0.0113
	Augmentation	Freebase1	0.4844 ± 0.0061	0.1001 ± 0.0215	0.1478 ± 0.0067	0.4844 ± 0.0061	0.1058 ± 0.0164
		Freebase2	0.4806 ± 0.0114	0.1062 ± 0.0292	0.1453 ± 0.0032	0.4806 ± 0.0114	0.1031 ± 0.0126
		Freebase3	0.4785 ± 0.0135	0.0806 ± 0.0154	0.1435 ± 0.0010	0.4785 ± 0.0135	0.1007 ± 0.0098
	Invariant	Freebase1	0.5001 ± 0.0115	0.2699 ± 0.0558	0.2275 ± 0.0068	0.5001 ± 0.0115	0.2025 ± 0.0083
		Freebase2	0.5001 ± 0.0115	0.2699 ± 0.0558	0.2275 ± 0.0068	0.5001 ± 0.0115	0.2025 ± 0.0083
		Freebase3	0.5001 ± 0.0115	0.2699 ± 0.0558	0.2275 ± 0.0068	0.5001 ± 0.0115	0.2025 ± 0.0083

Table 44: PathNet: node classification degree of invariance (IMDb, Freebase). Mean ± standard deviation over seeds of the pairwise Kendall-τ for each pair of variants; the Universal Graph is a single model, so its invariance is trivial.

Dataset	Method	Comparison	Kendall-τ
IMDb	Original	IMDb1 vs. IMDb2	0.7908 ± 0.1322
		IMDb1 vs. IMDb3	0.7650 ± 0.0951
		IMDb1 vs. IMDb4	0.7780 ± 0.0917
		IMDb2 vs. IMDb3	0.7892 ± 0.0957
		IMDb2 vs. IMDb4	0.7448 ± 0.1908
		IMDb3 vs. IMDb4	0.7238 ± 0.1664
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}}$ IMDb$_i$	IMDb1 vs. IMDb2	1.000 ± 0.0000 (trivial)
		IMDb1 vs. IMDb3	
		IMDb1 vs. IMDb4	
		IMDb2 vs. IMDb3	
		IMDb2 vs. IMDb4	
		IMDb3 vs. IMDb4	
	Augmentation	IMDb1 vs. IMDb2	0.9375 ± 0.0057
		IMDb1 vs. IMDb3	0.9535 ± 0.0055
		IMDb1 vs. IMDb4	0.9540 ± 0.0087
		IMDb2 vs. IMDb3	0.9415 ± 0.0016
		IMDb2 vs. IMDb4	0.9378 ± 0.0046
		IMDb3 vs. IMDb4	0.9426 ± 0.0105
	Invariant	IMDb1 vs. IMDb2	1.0000 ± 0.0000
		IMDb1 vs. IMDb3	1.0000 ± 0.0000
		IMDb1 vs. IMDb4	1.0000 ± 0.0000
		IMDb2 vs. IMDb3	1.0000 ± 0.0000
		IMDb2 vs. IMDb4	1.0000 ± 0.0000
		IMDb3 vs. IMDb4	1.0000 ± 0.0000
Freebase	Original	Freebase1 vs. Freebase2	0.9102 ± 0.0146
		Freebase1 vs. Freebase3	0.7344 ± 0.0618
		Freebase2 vs. Freebase3	0.7644 ± 0.0552
	Universal Graph $\bigcup_{i\in\{1,2,3\}}$ Freebase$_i$	Freebase1 vs. Freebase2	1.000 ± 0.0000 (trivial)
		Freebase1 vs. Freebase3	
		Freebase2 vs. Freebase3	
	Augmentation	Freebase1 vs. Freebase2	0.9823 ± 0.0124
		Freebase1 vs. Freebase3	0.9538 ± 0.0342
		Freebase2 vs. Freebase3	0.9538 ± 0.0332
	Invariant	Freebase1 vs. Freebase2	1.0000 ± 0.0000
		Freebase1 vs. Freebase3	1.0000 ± 0.0000
		Freebase2 vs. Freebase3	1.0000 ± 0.0000

Table 45: PathNet: node classification efficiency (IMDb, Freebase). One row per trained model; mean ± standard deviation over three seeds.

Dataset	Method	Variant	Train Time (s)	Epochs	Parameter (MiB)	Peak GPU (MiB)	
						Train	Infer.
IMDb	Original	IMDb1	17.9 ± 0.8	110.7 ± 5.2	0.93 ± 0.00	4703.03 ± 0.06	1286.78 ± 0.00
		IMDb2	19.9 ± 5.3	132.7 ± 35.6	0.93 ± 0.00	4703.03 ± 0.06	1286.78 ± 0.00
		IMDb3	19.5 ± 5.3	130.7 ± 34.9	0.93 ± 0.00	4703.03 ± 0.06	1286.78 ± 0.00
		IMDb4	16.2 ± 0.2	108.3 ± 1.2	0.93 ± 0.00	4703.03 ± 0.06	1286.78 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3,4\}} \text{IMDb}_i$	$\bigcup_i \text{IMDb}_i$	31.93 ± 0.37	124.00 ± 0.00	0.93 ± 0.00	6658.25 ± 0.00	5636.09 ± 0.00
	Augmentation	IMDb1-4	11.96 ± 0.43	31.00 ± 0.00	0.93 ± 0.00	4735.92 ± 0.00	1318.56 ± 0.00
	Invariant	IMDb1	25.9 ± 6.0	129.7 ± 30.6	0.93 ± 0.00	6701.21 ± 0.00	5679.98 ± 0.00
		IMDb2	25.7 ± 6.1	129.7 ± 30.6	0.93 ± 0.00	6701.21 ± 0.00	5679.98 ± 0.00
		IMDb3	25.7 ± 6.1	129.7 ± 30.6	0.93 ± 0.00	6701.21 ± 0.00	5679.98 ± 0.00
		IMDb4	25.7 ± 6.1	129.7 ± 30.6	0.93 ± 0.00	6701.21 ± 0.00	5679.98 ± 0.00
Freebase	Original	Freebase1	11.6 ± 1.2	46.3 ± 2.6	44.18 ± 0.00	7585.54 ± 0.30	6464.69 ± 0.30
		Freebase2	10.5 ± 0.5	47.3 ± 2.1	44.18 ± 0.00	7585.49 ± 0.22	6464.63 ± 0.22
		Freebase3	10.3 ± 0.9	45.0 ± 4.3	44.18 ± 0.00	7585.33 ± 0.00	6464.47 ± 0.00
	Universal Graph $\bigcup_{i\in\{1,2,3\}} \text{Freebase}_i$	$\bigcup_i \text{Freebase}_i$	9.7 ± 1.5	42.0 ± 5.7	44.18 ± 0.00	7585.70 ± 0.27	6464.84 ± 0.27
	Augmentation	Freebase1-3	19.04 ± 0.16	32.67 ± 0.47	44.18 ± 0.00	7581.91 ± 0.05	2258.63 ± 0.48
	Invariant	Freebase1	16.8 ± 0.9	44.0 ± 2.8	44.18 ± 0.00	12549.30 ± 0.27	10651.10 ± 0.27
		Freebase2	16.5 ± 1.1	44.0 ± 2.8	44.18 ± 0.00	12549.49 ± 0.00	10651.30 ± 0.00
		Freebase3	16.5 ± 1.1	44.0 ± 2.8	44.18 ± 0.00	12549.49 ± 0.00	10651.30 ± 0.00

Tables 16–45 report, for every GNN architecture and task type, the effectiveness, the degree of invariance, and the efficiency of the three baselines and our invariant approach (three tables per architecture and task type); e.g., RGCN's node classification results, invariance, and efficiency are in Tables 16, 17, and 18, and its link prediction counterparts in Tables 19–21. This section discusses them; unless stated otherwise, the numbers quoted are means over the variant rows (Micro-F1 for node classification, MRR for link prediction), "original" refers to the mean over the per-variant original models, and differences are given in points of the corresponding metric.

Degree of invariance. The original GNNs are far from invariant, and how far depends on the architecture, the task, and the mapping. The pairwise Kendall-τ of the originals runs from 0.50 (PAIN) to 0.80 (SlotGAT) on IMDb node classification, from 0.44 (SlotGAT) to 0.78 (CMPNN) on Movie–Director, from 0.47 (RGCN) to 0.74 (CMPNN) on DBLP, and from 0.55 (SlotGAT) to 0.80 (PathNet) on Freebase. The mined mappings of WN18RR change the outputs the most: there the originals agree at $\tau = 0.03$ (CMPNN) to 0.27 (SlotGAT), i.e., the ranking of the candidates of a test query is close to independent across variants. The spread of the effectiveness metrics across variants tells the same story from the other side: the original RGCN moves by 4.8 Micro-F1 points across the IMDb variants (Table 16), the original PathNet by 8.2 and the original PAIN by 7.2 points across the Freebase variants, the original MAGNN by 11.0 MRR points across the DBLP variants (Table 34), and the original CMPNN by 13.1 MRR points across the WN18RR variants (Table 28). Training on all variants raises the agreement substantially but never removes the dependence: augmentation reaches $\tau = 0.41$–0.95 on IMDb node classification, 0.62–0.99 on Freebase, 0.84–1.00 on DBLP, and only 0.34–0.82 on WN18RR, where the jointly trained model still ranks the candidates of the same query differently depending on which variant it is given. PAIN is the one architecture for which joint training does not even help: its augmented τ falls below its original on both datasets (0.40 against 0.50 on IMDb, 0.62 against 0.64 on Freebase). Its augmentation arm is also the least stable of any we trained, running for 461 ± 270 super-epochs across seeds without settling, which is what the drop in its IMDb effectiveness reflects. The invariant versions reach $\tau = 1.000 \pm 0.000$ on every pair of variants of every dataset, for all eight architectures, on the full ranking as well as on the top-1 and top-3 indicators, and their effectiveness spread across variants is zero. As discussed in Section H.3, this is exact rather than approximate: the models trained on different variants have identical parameters. The universal-graph baseline is also invariant, but trivially so, since it is a single model trained on a single graph that requires every variant to be available.

Effectiveness. Invariance does not come at the cost of effectiveness, and it usually improves it: the invariant version matches the original within 0.2 points or improves on it in 19 of the 23 model–task cells. The gains are largest where the original's structural bias is most sensitive to how a relation is encoded: CMPNN improves by 8.3 MRR points on WN18RR and 4.4 on Movie–Director, RGCN by 5.9 MRR points on DBLP and 2.3 Micro-F1 points on IMDb, MAGNN with the Union metapaths by 5.0 MRR points on DBLP, SeHGNN by 1.5 Micro-F1 points on IMDb, and MAGNN by 0.9 on IMDb. A two-layer RGCN on IMDb2, where director and actor are attached to the link node, reaches them only through two hops and drops to 0.603, whereas the compiled bias exposes both attributes to every movie on every variant. Compared with the universal graph, the invariant version matches it wherever the two compute the same function (Section H.3: RGCN, CMPNN, SlotGAT, and MAGNN Union on DBLP, SlotGAT on Freebase, PAIN on both of its datasets) and exceeds it elsewhere, by 7.1 MRR points for CMPNN on WN18RR, 6.0 Micro-F1 points for PathNet on IMDb, and 2.4 MRR points for RGCN on Movie–Director — without needing every variant at training time.

The cases in which the invariant version is less effective are informative, because each traces to a property of the architecture rather than to a loss of information in the compiled bias. *MAGNN with the Intersect metapaths* is a deliberate trade-off rather than a failure. Under the query-edge protocol the query paper's own area edges are masked, so on DBLP the area can reach the paper only through its co-authors, P-A-Ar-A-P; that metapath is in the Union set but not in the Intersect set, which contains exactly the metapaths of DBLP1 and therefore reproduces the DBLP1 model (0.318 against 0.401 for Union). What the minimal set buys instead is cost: it is $3.0\times$ cheaper than Union on Freebase (29 against 86 minutes) and $2.5\times$ cheaper on IMDb node classification (106 against 260 seconds), where it is also faster than the original, and it reaches $\tau = 1$ just as Union does. The smallest invariant bias is thus the cheapest and the least effective, and the two sets bracket the trade-off between the amount of information the bias carries and what it costs to carry it. *SlotGAT on Freebase* (0.640 against 0.655) normalizes one softmax over all incoming messages of a node regardless of relation.

On the compiled bias every book receives the messages of all shortcut compositions of the family, which are dense (99.8% of the 493 million edges are two- and three-hop compositions; a book has about 12 direct neighbors but about 9,000 endpoints through organization and music nodes), and the attention does not learn to separate them: the augmentation model scores 0.650, 0.649, and 0.599 on Freebase1–3, i.e., accuracy falls with the number of messages a node aggregates. That the universal graph scores exactly the same 0.640 shows the limit to be the architecture's rather than the compiled bias's. *SeHGNN on Freebase* (0.636 against 0.640 with the published type-collapsed channels, 0.664 against 0.670 with relation-aware channels) pays a related price: it has no mechanism to discount weak channels, and the multi-hop composition channels, estimated from 500 sampled walks each, are weak, so every additional composition channel costs a little. *PathNet on Freebase* (0.500 against 0.520) pays the same price for the same reason: it aggregates a fixed budget of 40 sampled walks per target, so on the denser compiled bias each walk is less likely to reach an informative path. The original degrades in the same way as shortcuts are added (0.551, 0.541, and 0.468 on Freebase1–3), and the universal graph, which carries the most compositions of all, is lower still at 0.477 — below the invariant version, so here too the ceiling is the sampling budget rather than the bias. *PAIN on IMDb* (0.567 against 0.574) is the smallest of these gaps and again equals the universal graph exactly. *CMPNN on WN18RR* is the opposite case: the invariant version (0.424) is far above the original mean (0.341), because the originals on WN18RR2 and WN18RR3 (0.308 and 0.293) are much worse than on WN18RR1 (0.424). On those variants the graph contains named inverse and shortcut relations that are copies of the training triples; masking the training query's edge leaves its copies in the message graph, so the model fits the training queries through them and generalizes worse. The invariant version propagates over the base relations only and reproduces the WN18RR1 model exactly on every variant, while the universal graph (0.352) and augmentation (0.317) train on the copies and inherit the problem.

Efficiency. The invariant versions have the same number of parameters as the originals, except for SeHGNN, whose parameter count depends on the number of channels (Table 13): on IMDb the invariant channel set is smaller than the physical ones (140 against 174 MiB on average, 429 for the universal graph), and on Freebase the relation-aware invariant version has 511 channels against 91–443 (563 against 352 MiB). Training time is governed by the size of the structural input that each architecture consumes, and on IMDb, DBLP, and WN18RR the compiled bias is small enough that invariance is free or better than free. Several invariant versions are *faster* than their originals: CMPNN on WN18RR (2.7 against 6.7 hours, because it propagates over the 86,835 base edges only while the originals on WN18RR2 and WN18RR3 propagate over 2–10× more), SeHGNN on IMDb (64 against 84 seconds, since it has fewer channels than any physical variant), and MAGNN with the Intersect set on both IMDb tasks. Others are within a few per cent of the original: RGCN on IMDb node classification (42 against 44 seconds), on Movie–Director (1.01×), and on WN18RR (1.06×); CMPNN on Movie–Director (1.04×) and DBLP (1.11×); SlotGAT on DBLP (0.94×); and PAIN on Freebase (1.09×). Freebase is the one dataset where invariance is consistently expensive for the message-passing architectures, because the compiled bias enumerates every shortcut composition of the family for every book: RGCN trains in 25 against 12 minutes (2.2×; peak GPU memory 0.6 GB in both), SlotGAT in 10.1 against 4.4 hours (2.3×; 10.2 against 6.6 GB), and MAGNN with the Union set in 1.4 hours against 21 minutes (4.2×), while MAGNN with the Intersect set (1.4×), PathNet (1.5×), and SeHGNN remain close to the original. The two remaining costs are PAIN on IMDb, whose exact path enumeration over the compiled bias takes 3.8× the original, and SlotGAT on WN18RR and Movie–Director (1.5×). In every one of these cases the invariant version costs about the same as, or less than, the universal-graph baseline, which consumes the union of the variants. The augmentation baseline trains one model on all variants, and since every super-epoch iterates over every variant's graph, its training time relative to a single invariant run ranges from a third to eight times (e.g., RGCN on WN18RR: 5.4 hours against 42 minutes).

Summary. Across seven architectures, four datasets, and five tasks, the original GNNs change their outputs substantially under information-preserving mappings (pairwise Kendall-τ between 0.03 and 0.80), training on all variants reduces but does not remove this dependence, and the invariant versions remove it exactly ($\tau = 1$, zero spread) while matching (within 0.2 points) or improving the effectiveness of the originals in 19 of 23 model–task cells, by up to 8.3 MRR points (CMPNN on WN18RR) and 5.9 (RGCN on DBLP). The four exceptions are explained by the architectures rather than by the mapping, and three of them are the same story: on Freebase, SlotGAT's single softmax over all incoming messages, SeHGNN's fixed set of channels, and PathNet's fixed budget of sampled

walks are all diluted by the dense shortcut compositions that the invariant bias must include — a cost that RGCN, which keeps relations separate, does not pay on the same bias — and in each case the universal graph, which carries the same information or more, does no better. MAGNN's two metapath sets bracket the remaining trade-off: the minimal set is invariant at the lowest cost and the lowest effectiveness, the Union set recovers the effectiveness at up to three times the cost, and both are exactly invariant. The universal-graph baseline reaches the same effectiveness only when the compiled bias coincides with the union of the variants, and it requires every variant at training time; the invariant version recovers the same model from each variant alone, and for CMPNN and PathNet it is more effective than the universal graph as well, by 6 to 7 points. The cost of invariance is the size of the compiled bias: free or negative on IMDb, DBLP, and WN18RR, and up to a factor of four in training time on Freebase, where the mined mappings materialize hundreds of millions of shortcut edges.