
Bridging Theory and Practice in Link Representation with Graph Neural Networks

Veronica Lachi*

Fondazione Bruno Kessler
Trento, Italy
vlachi@fbk.eu

Francesco Ferrini*

University of Trento
Trento, Italy
francesco.ferrini@unitn.it

Antonio Longa

University of Trento
Trento, Italy
antonio.longa@unitn.it

Bruno Lepri

Fondazione Bruno Kessler
Trento, Italy
lepri@fbk.eu

Andrea Passerini

University of Trento
Trento, Italy
andrea.passerini@unitn.it

Manfred Jaeger

University of Aalborg
Aalborg, Denmark
jaeger@cs.aau.dk

Abstract

Graph Neural Networks (GNNs) are widely used to compute representations of node pairs for downstream tasks such as link prediction. Yet, theoretical understanding of their expressive power has focused almost entirely on graph-level representations. In this work, we shift the focus to links and provide the first comprehensive study of GNN expressiveness in link representation. We introduce a unifying framework, the k_ϕ - k_ρ - m framework, that subsumes existing message-passing link models and enables formal expressiveness comparisons. Using this framework, we derive a hierarchy of state-of-the-art methods and offer theoretical tools to analyze future architectures. To complement our analysis, we propose a synthetic evaluation protocol comprising the first benchmark specifically designed to assess link-level expressiveness. Finally, we ask: does expressiveness matter in practice? We use a graph symmetry metric that quantifies the difficulty of distinguishing links and show that while expressive models may underperform on standard benchmarks, they significantly outperform simpler ones as symmetry increases, highlighting the need for dataset-aware model selection.

1 Introduction

Graph Neural Networks (GNNs) have achieved remarkable success across a wide range of tasks involving structured data, including node-level [19, 51, 36, 28], graph-level [18, 52], and link-level tasks [44, 58, 56, 13]. Despite the growing importance of link representation learning, the theoretical analysis of GNNs has so far focused almost exclusively on their expressiveness for graph-level representation [53, 37, 6]. In particular, it is well-established that message-passing GNNs are at most as powerful as the 1-dimensional Weisfeiler-Lehman (1-WL) test in distinguishing non-isomorphic graphs [53, 37]. However, their ability to distinguish *links* – that is, to generate discriminative representations of node pairs – remains far less understood.

It is well known that standard GNNs struggle to distinguish structurally different links, as they typically compute link embeddings by aggregating the representations of the two endpoint nodes. This node-centric strategy introduces a significant expressiveness bottleneck: links whose endpoints are automorphic may be mapped to identical representations, even if their structural roles in the graph differ [44, 58, 13]. To mitigate this issue, several methods have extended GNNs with structural

*Equal contribution

features (SFs) [58, 56, 46, 13]. While such approaches are more expressive than standard GNNs, there is still no unifying framework to systematically characterize their discriminative power or organize them into a principled expressiveness hierarchy—unlike the case of graph-level representation, where the k -Weisfeiler-Lehman hierarchy provides a widely adopted standard [37, 53].

In this work, we bridge this gap by conducting a comprehensive investigation into the expressive power of GNN-based models for *link representation*. We observe that most existing methods for link representation combine two ingredients: a message-passing GNN for node encoding and a pairwise function for aggregating information about the target link. Building on this insight, we propose a *general theoretical framework* that captures a broad class of link representation methods. The framework characterizes models based on two key dimensions: the expressiveness of the functions involved and the radius of the neighborhood around the link that they access. It subsumes many state-of-the-art models and allows us to formally reason about their ability to distinguish links.

We explore link-level expressiveness along three axes. **First**, grounded in the proposed framework, we develop a theoretical foundation to analyze and *compare the expressiveness of link representation models*. Specifically, we derive criteria to assess a model’s capacity to discriminate between different link structures, enabling us to organize existing methods into an expressiveness hierarchy and to position novel models within it. **Second**, we introduce a *synthetic evaluation protocol* designed to probe the expressiveness of link representation methods. This protocol comprises both a synthetic dataset and an accompanying evaluation procedure. While previous work has introduced synthetic tools for assessing graph-level expressiveness [7, 48, 1, 40, 3, 8], to the best of our knowledge, no analogous setup exists for links. Our procedure enables systematic, reproducible, and scalable comparisons across models. **Third**, we investigate the *practical relevance* of expressiveness on link representation by studying whether more expressive models lead to better performance in real-world scenarios. We propose a graph symmetry metric to quantify structural ambiguity among links, and we empirically show that while simple models suffice in low-symmetry settings, more expressive models significantly outperform them as symmetry increases.

2 Preliminaries

Definition 2.1 (graph). A **graph** is a tuple $G = (V_G, E_G)$ where $V_G = \{1, \dots, n\}$ is a set of nodes, $E_G \subseteq V_G \times V_G$ is a set of edges. To each graph is associated a node features matrix $\mathbf{X}^0 \in \mathbb{R}^{n \times f}$ and an adjacency matrix $\mathbf{A}_G \in \{0, 1\}^{n \times n}$ with $\mathbf{A}_{G_{u,v}} = 1$ if and only if $(u, v) \in E_G$. In our analysis, we consider simple, finite and undirected graphs. We will call $\mathcal{G}$ the set of all graphs.

Definition 2.2 (neighborhood). Let $G = (V_G, E_G)$ be a graph. Given two nodes $u, v \in V_G$, u is said to be m -far from v if there exists a path of m edges connecting u and v . Given a node $v \in V_G$, we denote by $N^m(v)$ the set of nodes m -far away from v . In particular, the 1-hop neighborhood of v , denoted $N(v)$, corresponds to $N^1(v)$ and includes all nodes connected to v , i.e., $N(v) = \{u \in V_G \mid (v, u) \in E_G\}$. Finally, given two nodes $u, v \in V_G$, their joint m -hop neighborhood is defined as $N^m(u, v) = N^m(u) \cup N^m(v)$.

Definition 2.3 (link representation model). A **link representation model** M is a class of functions F mapping node pairs in graphs to vector representations:

$$F : ((u, v), G, \mathbf{X}^0) \mapsto \mathbf{x}_{(u,v)} \in \mathbb{R}^d \quad (1)$$

with $G \in \mathcal{G}$, $\mathbf{X}^0 \in \mathbb{R}^{n \times f}$ node feature matrix and $u, v \in V_G$.

In this work, we adopt a broad notion of *link*, referring to any pair of nodes $(u, v) \in V_G \times V_G$, regardless of whether an edge between them actually exists in G , i.e., whether $(u, v) \in E_G$. Accordingly, the link representation model computes a vector representation for arbitrary node pairs, not only for existing edges. Throughout the paper, the term *link* will always refer to a generic node pair.

Models that compute link representations can be applied to a variety of downstream tasks. For instance, they can be used for *link prediction* [57, 35, 60], where the goal is to predict the existence of an edge between two nodes, for *link classification* [43, 47, 15], where different types of relationships are inferred and for *link regression* [33, 17], where real-valued properties associated with node pairs are estimated. In this work, we focus on message-passing (MP) link representation models, which have demonstrated superior performance and robustness across a wide range of benchmarks [32].

Definition 2.4 (GNN). Let $G = (V_G, E_G)$ be a graph with feature matrix $\mathbf{X}^0$. A GNN iteratively updates the representation of nodes $v \in V_G$ following the propagation scheme:

$$\mathbf{x}_v^0 = \mathbf{X}_{[v,:]}^0 \quad \mathbf{x}_v^l = \text{UPDATE}(\mathbf{x}_v^{l-1}, \text{AGGREGATE}(\{\mathbf{x}_u^{l-1} \mid u \in N(v)\})) . \quad (2)$$

We denote the final representation at the L -th layer as $\mathbf{x}_v^L = \rho(v, G, \mathbf{X}^0)$.

In the rest of this paper, whenever we refer to MP models, we refer to models that employ the iterative mechanism defined in Definition 2.4 to compute node representations.

A key theoretical question concerns GNNs expressive power, their ability to distinguish non-isomorphic inputs. Standard GNNs have been shown to be at most as powerful as the 1-WL test [37, 54], a heuristic for graph isomorphism based on iterative multiset aggregation [50]. To go beyond this limit, higher-order GNNs [37] aggregate over k -tuples of nodes, reaching the expressivity of the k -WL test; we denote by k_ρ the smallest such k such that ρ is as powerful as k -WL, i.e., k_ρ expresses the representational power of ρ within the k -WL paradigm. In contrast to graph-level tasks, the expressiveness of message-passing models for link representation remains less explored [58, 44, 24]. We now provide formal definitions on what does it mean for two links to be different in a graph.

Definition 2.5 (node permutation). A **node permutation** $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ is a bijective function that assigns a new index to each node of the graph. All the $n!$ possible node permutations constitute the permutation group Π_n . Given a subset of nodes $S \subseteq V_G$, we define the permutation π on S as $\pi(S) := \{\pi(i) \mid i \in S\}$. Additionally, we define $\pi(\mathbf{A}_G)$ as the matrix $\mathbf{A}_G$ with rows and columns permuted based on π , i.e., $\pi(\mathbf{A}_G)_{\pi(i), \pi(j)} = (\mathbf{A}_G)_{i,j}$.

Definition 2.6 (automorphism). An **automorphism** on the graph $G = (V_G, E_G)$ is a permutation $\sigma \in \Pi_n$ such that $\sigma(\mathbf{A}_G) = \mathbf{A}_G$. All the possible automorphisms on a graph G constitute the automorphism group Σ_n^G .

Definition 2.7 (automorphic links). Let $G = (V_G, E_G)$ be a graph and Σ_n^G its automorphism group. Two pairs of nodes $(u, v), (u', v') \in V_G \times V_G$ are said to be **automorphic links** ($(u, v) \simeq (u', v')$) if exists $\sigma \in \Sigma_n^G$ such that $\sigma(\{u, v\}) = \{u', v'\}$.

The notion of automorphic links provides the foundation for evaluating models expressivity, as an expressive model should be able to distinguish between non automorphic links.

Definition 2.8 (more expressive). Let M_1 and M_2 be two link representation models (Def. 2.3). M_2 is **more expressive** than M_1 ($M_1 \leq M_2$) if, for any graph $G = (V_G, E_G)$ and any pair $(u, v), (u', v') \in V_G \times V_G$ with $(u, v) \not\simeq (u', v')$:

$$\exists F_1 \in M_1 : F_1((u, v), G, \mathbf{X}^0) \neq F_1((u', v'), G, \mathbf{X}^0) \Rightarrow \exists F_2 \in M_2 : F_2((u, v), G, \mathbf{X}^0) \neq F_2((u', v'), G, \mathbf{X}^0).$$

3 The Expressive Power of MP-based Link Representation Methods

We propose a general framework for message-passing models that compute link representations, encompassing a broad class of existing methods as specific instances. By formalizing these models under a unified perspective, we enable a principled analysis of their expressive power. Leveraging this framework, we further establish an expressiveness-based hierarchy of existing approaches.

3.1 MP-based methods for link representation

Before defining the framework, we first review representative MP-based models, including Pure GNNs [30, 29, 45, 22, 53], NCN [46], NCNC [46], ELPH [13], BUDDY [13], Neo-GNN [56] and SEAL [58].

Pure GNN Methods. Pure GNN Methods learn representation $\mathbf{x}_{(u,v)} \in \mathbb{R}^d$ for each pair of node (u, v) with $u, v \in V_G$ as $\mathbf{x}_{(u,v)} = g(\mathbf{x}_u^L, \mathbf{x}_v^L)$, where g is an aggregation function and $\mathbf{x}_u^L, \mathbf{x}_v^L$ are the node representation of u and v learned by the GNN at the final layer L . In principle, any type of GNN can be used, and the function g can be modeled using an MLP over any aggregation function over the feature vectors. In practice, the most commonly used pure GNN model is GAE [29].

Neo-GNN. Given a graph $G = (V_G, E_G)$, Neo-GNN computes a node representation matrix $\mathbf{Z}$ as:

$$\mathbf{Z} = \sum_{l=1}^L \beta^{l-1} \mathbf{A}_G^l \mathbf{X}^{\text{struct}} \quad \text{with} \quad \mathbf{X}^{\text{struct}} = \text{diag}(\mathbf{x}^{\text{struct}}), \quad \mathbf{x}_v^{\text{struct}} = \mathcal{F}_\Theta((\mathbf{A}_G)_v) \quad (3)$$

where $\mathcal{F}_\Theta$ is a learnable function over the adjacency matrix A_G , L is the maximum hop considered, and $\beta \in (0, 1)$ controls the weight of distant neighbors. The final link representation is computed as an aggregation of the node representation obtained through standard GNN and the one in $\mathbf{Z}$, namely $\mathbf{x}_{(u,v)} = g((\mathbf{z}_u, \mathbf{z}_v), (\mathbf{x}_u^L, \mathbf{x}_v^L))$.

NCN. NCN computes the representation of node pairs using the representations of their common neighbors. NCN defines the link representation as:

$$\mathbf{x}_{(u,v)} = \mathbf{x}_u^L \odot \mathbf{x}_v^L \parallel \sum_{i \in N(u) \cap N(v)} \mathbf{x}_i^L, \quad (4)$$

where $\mathbf{x}_u^L$ is the node representation at the final L -th layer of a GNN. NCNC extends NCN to deal with settings with incomplete topological information (e.g., for link prediction or graph completion), by considering not only nodes in $N(u) \cap N(v)$ but all nodes in $N(u) \cup N(v)$ and calculating for them the probability of being common neighbors with NCN.

SEAL. Given a graph $G = (V_G, E_G)$, for each link (u, v) , SEAL constructs an h -hop enclosing subgraph $G_{uv}^{(h)}$, which contains all nodes within h hops of u and v . Nodes in $G_{uv}^{(h)}$ are labeled using the Double-Radius Node Labeling (DRNL)[58], which assigns different labels based on distances to u and v . The representation of link (u, v) is obtained processing the graph $G_{uv}^{(h)}$ with a GNN.

ELPH. In ELPH, two types of count-based features are computed: (i) $A_{uv}[d_u, d_v]$, i.e., the number of nodes at distance d_u from node u and d_v from node v ; and (ii) $B_{uv}[d]$, the number of nodes at distance d from u and more distant than d from v . These counts are estimated using *MinHash* (for Jaccard similarity) and *HyperLogLog* (for set cardinality). The link representation is obtained as:

$$\mathbf{x}_{(u,v)} = g(\mathbf{x}_u^L, \mathbf{x}_v^L, \{A_{uv}[d_u, d_v], B_{uv}[d] \mid d, d_u, d_v < [L]\}), \quad (5)$$

where g is an aggregation function and L is the final layer. To overcome ELPH's memory limitations, BUDDY precomputes structural features via offline graph traversals and sketching.

3.2 A General Framework for MP-based Link Representation Methods

Building on the definitions introduced in Section 3.1, we observe that MP-based approaches for link representation generally follow a common paradigm: they combine the representations of the two endpoint nodes with additional pairwise information extracted from their local neighborhoods, except for pure GNN methods, which rely solely on node representations. Although specific mechanisms and architectural designs vary, the core structure of these models can be systematically characterized by two key factors: (i) the link neighborhood radius, which defines the portion of the graph considered when extracting pairwise information for the link, and (ii) the expressiveness of the message-passing (MP) functions used to compute the node representations. Importantly, the link neighborhood radius

Table 1: Model formulations from Section 3.1 expressed within the k_ϕ - k_ρ - m framework. A ‘/’ indicates that the corresponding component is not included in the model.

Model	COMB	g	k_ϕ	AGG	ψ	k_ρ	m	h
Pure GNN	/	$\odot$	1-WL	/	/	/	/	/
NCN	$\parallel$	$\odot$	1-WL	$\sum$	$\rho(i; G, \mathbf{X}^0)$	1-WL	1	$\mathbf{X}^0$
ELPH	$\parallel$	$\parallel$	1-WL	$\sum$	$\rho(i; G, \mathbf{X}^1) \cdot \prod_{r=1}^m \prod_{d=1}^m \mathbb{1}_{dr}(i)$	1-WL	m	$\mathbf{x}_i^1 = 1$
Neo-GNN	$\parallel$	$\parallel$	1-WL	$\sum$	$b \cdot \rho(i; G, \mathbf{X}^0)$ with $b = \sum_{r=1}^m \sum_{d=1}^m A_{uv}^r \cdot A_{uv}^d$	1-WL	m	$\mathbf{X}^0$
SEAL	/	/	/	$\sum$	$\rho(i; G, \mathbf{X}^D)$	$1 - N^m(u, v) $ -WL	m	$\mathbf{x}_i^D = \mathbf{x}_i^0 \parallel \min_{u,v}(\delta(i, u), \delta(i, v)) + 1$

is distinct from the radius induced by the depth of the MP function itself. Motivated by these common factors, we introduce a general framework that unifies existing approaches under a common formalism. In this framework, the link representation is constructed by combining: (1) the representations of the two endpoint nodes, and (2) a representation of the link’s m -order neighborhood, each computed via MP functions that may have different levels of expressive power and different number of layers.

Definition 3.1 (k_ϕ - k_ρ - m framework). Given a graph $G = (V_G, E_G)$ with feature matrix $\mathbf{X}^0$, a MP link representation model M belongs to the k_ϕ - k_ρ - m framework if its functions can be expressed as:

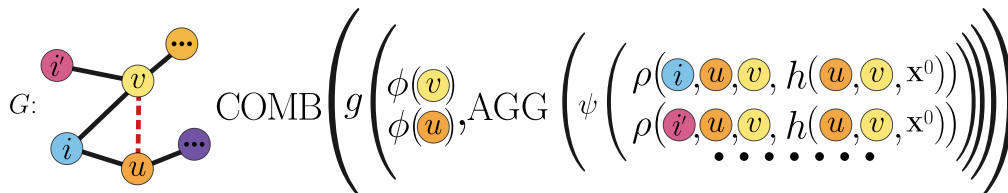
$$f(i, u, v, \mathbf{X}^0) = \psi(\rho(i, h(u, v, \mathbf{X}^0)), u, v) \quad (7)$$

Table 1 provides the formulation of all the models presented in Section 3.1 using the k_ϕ - k_ρ - m framework, while Figure 1 provides a visualization of the framework components. Through the k_ϕ - k_ρ - m framework, we can systematically study the expressive power of MP-based link representation methods. Let $\mathcal{M}$ be the set of link representation models expressible by the k_ϕ - k_ρ - m framework. Let $M \in \mathcal{M}$, we indicate with k_ϕ^M - k_ρ^M - m^M the specific value that k_ϕ - k_ρ - m assume in the model M .

1. If $m^M = 0$, then, regardless of k_ϕ^M , M is not able to distinguish between links whose endpoints are automorphic, i.e.,

$$F((u', v'), \mathbf{X}^0) = F((u, v), \mathbf{X}^0) \quad (9)$$

2. Let $M_1, M_2 \in \mathcal{M}$ with $k_\rho^{M_1} = k_\rho^{M_2} = I$ -WL and let l^{M_1}, l^{M_2} be the number of layers used respectively by M_1, M_2 . If $m^{M_1} + l^{M_1} \leq m^{M_2} + l^{M_2}$, then $M_1 \leq M_2$.
3. Let $M_1, M_2 \in \mathcal{M}$ with $m^{M_1} = m^{M_2}$. If $k_\rho^{M_1} \leq k_\rho^{M_2}$, then $M_1 \leq M_2$.



Proof is provided in Appendix A. Intuitively, the theorem highlights that relying solely on the representations of the endpoints through g and ϕ is fundamentally limited: even if ϕ is highly expressive, such models fail to distinguish links whose endpoints are automorphic. This limitation arises from the fact that the k -WL algorithm, preserves graph automorphisms for every k [34, 16, 11]. Instead, a more efficient and expressive modeling strategy is to compute ρ over a suitable m -order neighborhood using an appropriate number of MP layers. In many models, the number of such layers is a design choice: for instance, *ELPH* encodes link information by counting nodes at various distances from the link, which effectively corresponds to zero message-passing layers. Conversely, in *Neo-GNN*, node representations are computed through a function on the adjacency matrix, effectively implementing a single round of message passing. This distinction is crucial: even models with large neighborhoods m but few message-passing layers can be outperformed—both in expressiveness and efficiency—by models with smaller m but deeper architectures. Thus, controlling expressiveness via the number of layers offers a more principled and computationally efficient path. Finally, increasing the expressiveness of the base model ρ on the m -order neighborhood leads to more expressive representations. One example of model with more expressive ρ is *SEAL* which increases expressiveness by adding link-aware positional encodings to node features, effectively simulating the power of $1-|N^m(u, v)|$ -WL, which is strictly more expressive than 1-WL [59]. However, increasing expressivity beyond 1-WL comes at a computational cost: for example, the complexity of k -WL is $O(n^{k+1} \log n)$ [25]. Details on the computational costs of the models are provided in Appendix D.

Using Theorem 3.2 it is possible to study the expressiveness of existing methods. The following Theorem introduces a hierarchy of existing methods.

Theorem 3.3. *The following hold:*

1. *For any radius and number of layers, NCN, Neo-GNN, ELPH, SEAL are more expressive than Pure GNNs;*
2. *if $l^{NCN} \geq m^{Neo-GNN}$, NCN is more expressive than Neo-GNN;*
3. *if $m^{Neo-GNN} \geq m^{ELPH} - 1$, Neo-GNN is more expressive than ELPH;*
4. *if $l^{NCN} \geq m^{ELPH} - 1$, NCN is more expressive than ELPH;*
5. *SEAL is more expressive than NCNC, NCN, ELPH, Neo-GNN.*

Proof is provided in Appendix A and rely on Theorem 3.2 and the formal expression of models within the theoretical framework described in Table 1.

The k_ϕ - k_ρ - m framework serves as a foundation for analyzing the expressiveness of link representation models in a principled and structured way. In the next section, we complement this theoretical perspective by introducing a synthetic evaluation protocol designed to empirically assess the expressiveness of link representation models.

4 A Synthetic Evaluation Procedure for Link Representation Expressiveness

We introduce a synthetic protocol to empirically assess a MP model’s ability to assign distinct representations to structurally different links. The protocol is designed to help evaluate new link representation models in a simple and controlled setting. It consists of a synthetic dataset and an evaluation framework. The synthetic dataset and the code are publicly available².

4.1 LR-EXP: A Synthetic Benchmark for Link-Level Expressiveness

While several synthetic datasets have been proposed to evaluate the expressive power of GNNs in distinguishing non-isomorphic graphs [7, 48, 1, 40, 3, 8], a comparable benchmark for assessing link-level expressiveness is still missing, despite its potential utility as a fast and controlled way to empirically assess the expressivity of both existing and new models. To address this gap, we propose a novel synthetic dataset, LR-EXP, aimed at measuring a model’s ability to assign different outputs to non-automorphic links. To make the distinction task challenging, the dataset includes non-automorphic links whose endpoint nodes share the same 1-WL colors.

²<https://anonymous.4open.science/r/link-representation-gnn-8124/README.md>

This is a critical design choice: non-automorphic links with differently colored nodes are trivially distinguishable by any pure GNN with an injective aggregation function g . LR-EXP includes 1400 graphs, each containing at least one such pair of non-automorphic links. Each graph is generated in two steps: (1) sample an Erdős–Rényi graph with $n \in \{5, \dots, 17\}$ nodes and edge probability $p \sim \mathcal{U}(0, 1)$; (2) duplicate the graph and add inter-block edges with probability $p' \sim \mathcal{U}(0, 1)$. This construction introduces rich symmetries and increases the probability of obtaining node pairs that are non-automorphic yet indistinguishable under the WL test. We retain only those graphs where such link pairs exist, and use them to define training, validation, and test sets. Figure 2 shows examples of these link pairs. Importantly, while LR-EXP is relatively small in scale, this is by design: our goal is not to evaluate scalability, but to isolate and probe expressive. Since expressiveness concerns a model’s ability to produce different outputs for structurally distinct inputs, rather than generalization over large data volumes, dataset size is not a confounding factor for the phenomenon we aim to study.

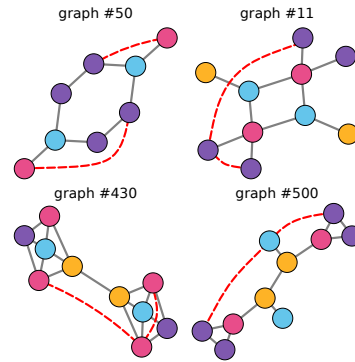


Figure 2: Examples of LR-EXP graphs. Node colors show WL colors; dashed lines mark test links that are non-automorphic but indistinguishable for standard GNNs.

4.2 Evaluation Framework

Our training and evaluation approach is inspired by the method proposed in Wang and Zhang [49]. In particular, we employ a siamese network design [9], with central component consisting of two identical models with identical parameters, each processing one link independently to generate separate embeddings. For a pair of links $(u, v) \not\sim (u', v') \in V_G \times V_G$, and a model F , the model generates a corresponding pair of embeddings. The model is trained with the goal of encouraging separation of links embeddings using the contrastive loss:

$$L(F, (u, v), (u', v'), \mathbf{X}^0) = \max \left(0, \frac{F((u, v), \mathbf{X}^0) \cdot F((u', v'), \mathbf{X}^0)}{\|F((u, v), \mathbf{X}^0)\| \|F((u', v'), \mathbf{X}^0)\|} \right), \quad (10)$$

Contrastive loss functions are often defined with an explicit margin term which controls the degree of dissimilarity enforced between negative pairs. In our case, the goal is to fully minimize similarity between representations of non-automorphic links. Following the approach in [48], we deliberately adopted a zero-margin formulation to push the model toward maximal dissimilarity between such pairs. This choice aligns with our objective of measuring and testing expressiveness, where even small overlaps in representation between structurally different links would be undesirable.

To evaluate models, we compare their outputs on pairs of non-automorphic links. If the resulting representations differ sufficiently, this indicates successful discrimination. However, selecting an appropriate threshold is challenging: a high threshold may cause false negatives, where true distinctions fall below the threshold; a low threshold may cause false positives, where random noise creates apparent differences. To address this, we adopt the Reliable Paired Comparison method [49], which assesses groups of results using two components: the Major Procedure, quantifying representation differences, and the Reliability Check, measuring internal variability. A pair is considered distinguishable only if both checks are passed. As all pairs are non-automorphic, the task is framed as one-class classification, and we report precision as the fraction of correctly distinguished pairs. See Appendix B.1 for details.

4.3 Experimental results on LR-EXP

We evaluated the expressive power of all models from Section 3.1 on LR-EXP. As discussed in Section 3.2, some models treat m as a fixed design choice while keeping the number of layers l as a tunable parameter, and vice versa. Even in models where m is left as a free parameter, practical implementations impose constraints due to computational costs. For example, both ELPH and SEAL limit m to a maximum of 3. In our experiments, we adopt $m = 3$ for all models that allow it, ensuring a fair comparison. We set the number of layers $l = 3$ across all methods, which strikes a balance between expressiveness and avoiding oversmoothing effects [10]. Under this configuration, Theorem

3.3 shows that NCN is more expressive than ELPH and matches the expressiveness of Neo-GNN, while SEAL is the most expressive method.

Table 2: Test precision (mean $\pm$ std over 5 runs) for different models. Higher model expressiveness leads to higher test precision.

Model	Test Precision(%)
<i>GCN</i>	0 ($\pm$ 0)
<i>GAT</i>	0 ($\pm$ 0)
<i>GAE</i>	0 ($\pm$ 0)
<i>SAGE</i>	0 ($\pm$ 0)
<i>BUDDY</i>	45 ($\pm$ 1)
<i>ELPH</i>	62 ($\pm$ 7)
<i>Neo-GNN</i>	75 ($\pm$ 2)
<i>NCN</i>	75 ($\pm$ 1)
<i>SEAL</i>	97 ($\pm$ 0)

The experimental results on LR-EXP provide empirical validation of our theoretical framework. Table 4.3 shows that Pure GNN methods (*GCN*, *GAT*, *GAE*, *SAGE*) completely fail to distinguish non-automorphic links, achieving zero precision. This is a consequence of their fundamental limitation - the automorphic node problem we analyzed in Section 3.2, which prevents them from generating distinct representations for links composed by automorphic nodes. The remaining methods demonstrate progressively better performance as their expressive power increases. The performance gap between *BUDDY* (45%) and *ELPH* (62%) stems from *BUDDY*'s approximate counting of common neighbors (Section 3.1), which reduces its expressiveness as shown in [13]. *Neo-GNN* and *NCN*³ achieve comparable precision (75%), outperforming both *ELPH* (62%) and *BUDDY* (45%). Finally, *SEAL* demonstrates its high expressiveness, being able to correctly distinguish 97% of links. Further results as we vary the number of layers l and the radius

m are in Appendix. Appendix B.2 reports robustness checks under variations of the significance level employed in the Reliable Paired Comparison.

5 Do More Expressive Models Perform Better in Real-World Benchmarks?

An interesting and practically relevant question is whether increased expressivity in link representation translates into better performance on downstream tasks involving real-world data [27, 38]. Among the various tasks enabled by expressive link representations, we focus on link prediction due to its broad impact across domains such as recommender systems [55], knowledge graph completion [41], and biological interaction prediction [26]. Intuitively, link prediction should particularly benefit from models with high expressive power when the dataset contains many non-automorphic links whose endpoint nodes are automorphic, a setting where Pure GNNs struggle. In this section, we aim to answer the following three questions: (1) How can we formally quantify the presence of non-automorphic links with automorphic endpoints in a dataset? (2) Are there real-world link prediction benchmarks where this property is especially prevalent? (3) How do models with varying levels of expressiveness perform across datasets with different levels of structural difficulty?

We address these questions by first introducing a metric to quantify the link-level symmetry of a dataset. We then apply this metric to a collection of real-world link prediction benchmarks and analyze their structural difficulty. Finally, we evaluate the performance of several state-of-the-art models with different expressive capabilities across these datasets. Code is available at⁴.

5.1 A Metric for Measuring the Symmetry of Graphs

We introduce a metric to quantify the presence of non automorphic links with automorphic endpoints in a dataset. This occurs when the underlying graph exhibits a high number of non-trivial automorphisms that is, intuitively, when the graph is highly symmetric. The symmetry of a graph is tightly connected to the concept of *node orbits*, which we define below.

Definition 5.1 (orbit). The **orbit** of a vertex $v \in V_G$ under the action of the automorphism group Σ_n^G is defined as the set of nodes of V_G to which v can be mapped via an automorphism $\sigma \in \Sigma_n^G$. Formally, $\text{Orb}(v) = \{\sigma(v) \mid \sigma \in \Sigma_n^G\}$. We denote with $O_G = \{\text{Orb}(v) \mid v \in V_G\}$ the set of all the orbits of V with respect to the action of Σ_n^G .

Following [4], it is possible to define a metric that measures how much a graph is symmetric as follows:

³We do not report results for NCNC, as it coincides with NCN for fully observed topologies.

⁴<https://anonymous.4open.science/r/link-representation-gnn-8124/README.md>

Table 3: MRR on real-world link prediction datasets sorted by increasing $\hat{r}_G$. The top three results are highlighted using **first**, **second**, and **third**. OOM means Out Of Memory, >24h means more than 24 hours. As symmetry increases, only more expressive models maintain top performance. Results are averaged over 5 runs with different random seeds; standard deviations are reported in Appendix F.

Models	Graph Symmetry ($\rightarrow$)											
	OGBL CITATION2	CORA	CITeseer	PUBMED	OGBL DDI	OGBL PPA	OGBL COLLAB	NSC	YST	GRQ	AIFB	EDIT TSW
$\hat{r}_G$	0.01	0.02	0.02	0.08	0.10	0.12	0.13	0.16	0.23	0.30	0.41	0.67
GCN	19.98	16.61	21.09	7.13	13.46	26.94	6.09	26.79	1.17	6.89	10.85	4.12
GAT	OOM	13.84	19.58	4.95	12.92	OOM	4.18	2.47	0.40	0.63	1.07	0.40
SAGE	22.05	14.74	21.09	9.40	12.60	27.27	5.53	20.32	1.77	1.34	0.54	8.35
GAE	OOM	18.32	25.25	5.27	3.49	OOM	OOM	16.61	2.32	2.32	10.98	3.54
BUDDY	19.17	13.71	22.84	7.56	12.43	27.70	5.67	18.40	13.68	46.23	13.73	22.66
Neo-GNN	16.12	13.95	17.34	7.74	10.86	21.68	5.23	22.86	5.78	29.51	13.11	8.10
NCN	23.35	14.66	28.65	5.84	12.86	35.06	5.09	30.36	11.99	48.45	6.47	10.32
NCNC	19.61	14.98	24.10	8.58	>24h	33.52	4.73	30.20	13.54	47.60	OOM	10.30
SEAL	20.60	10.67	13.16	5.88	9.99	29.71	6.43	30.85	17.51	56.72	16.30	25.82

Definition 5.2 (graph symmetry measure r_G). Given a graph $G = (V_G, E_G)$, its **symmetry measure** is calculated as:

$$r_G = 1 - \frac{|O_G| - 1}{|V_G| - 1} \quad (11)$$

where $|O_G|$ denotes the number of orbits of the graph and $|V_G|$ is the number of nodes.

Computing all node orbits is computationally expensive. However, the number of orbits can be efficiently approximated using the number of distinct colors assigned by the WL test at convergence WL_G [31, 39], i.e.,

$$\hat{r}_G = 1 - \frac{|WL_G| - 1}{|V_G| - 1} \quad (12)$$

The approximation in Equation 12 reflects how closely the WL-induced partition (used in Equation 11) matches the true orbit partition. Although there is no complete characterization of WL failures in distinguishing nodes from different orbits, it is said that WL gives different representations to *almost all* nodes that belong to different orbits [2]. A practical example is presented in [39], where the authors explicitly compare WL partitions and orbit partitions on the Alchemy dataset [14]. Out of 202,579 graphs, only in 4 graphs the partition of the nodes induced by the WL is different wrt the orbits, highlighting that the WL approximation is extremely close in practice.

5.2 Symmetric Real-World Datasets for Link Prediction

We selected twelve datasets for link prediction, including well-known citation networks such as CORA and CITESEER, large-scale benchmarks from the Open Graph Benchmark suite like OGBL-CITATION2 and OGBL-COLLAB, and structurally complex knowledge and ontology graphs such as AIFB, EDIT-TSW. More details about datasets can be found in Appendix E.1. For each dataset, we compute the proposed symmetry metric $\hat{r}_G$ (Equation 12). The computed values are reported in the first row of Table 3, with datasets sorted in ascending order of $\hat{r}_G$. As shown, the datasets exhibit widely varying degrees of symmetry. For instance, OGBL-CITATION2, CORA, and PUBMED display very low $\hat{r}_G$ values, largely due to the presence of informative node features that yield many unique WL colors. At the other end of the spectrum, datasets such as EDIT-TSW, which lacks node features and exhibit high topological regularity, show significantly higher symmetry scores.

Table 3 reports the results, in terms of the standard link prediction metric MRR, achieved by the models described in Section 3.1. The models are listed in ascending order of theoretical expressiveness, as characterized by Theorem 3.3. Since we are now evaluating under the standard link prediction setting (i.e., graphs with missing edges), we also include the NCNC variant of NCN. For ELPH, we report results for BUDDY, its more scalable version. All evaluations are conducted under the challenging and realistic HearT setting [32]. Further details on model architectures and hyperparameter selection are provided in Appendix E, while the complete results, including standard deviations and an additional evaluation metric, are reported in Appendix F.

The results reveal a clear trend: as $\hat{r}_G$ increases, more expressive models become necessary. In the five datasets with the lowest symmetry, pure GNN-based methods consistently rank among the top three performers. However, as symmetry increases, their performance deteriorates and none of them appears in the top three on the six most symmetric datasets. Conversely, SEAL, the most expressive model in our hierarchy, ranks first across all high-symmetry datasets. This provides a key practical insight: while simple, less expressive models may suffice when node features are informative and structural ambiguity is limited, datasets with high link symmetry demand more powerful architectures.

6 Conclusion

We propose a theoretical framework that formally compares a wide range of existing models and establishes a hierarchy of them. We introduced LR-EXP, the first synthetic benchmark explicitly designed to evaluate link-level expressiveness. Finally, we demonstrated that increased expressiveness translates into performance gains on real-world link prediction tasks, especially in structurally symmetric graphs. These findings highlight that model selection should account for dataset symmetry: while simple models may suffice in low-symmetry settings, high-expressiveness architectures are essential when structural ambiguity is present.

Limitations and Future Work Our framework is tailored to MP models and does not directly account for recent advances in transformer-based architectures or spectral methods for link representation. Extending the framework to encompass these directions may provide additional insights, making this an interesting and valuable direction for future work. Furthermore, the high-symmetry datasets used in our evaluation are relatively small, highlighting the need for larger and more robust benchmarks to properly assess model generalization and advance research in this area [5].

Acknowledgments

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them. Grant Agreement no. 101120763 - TANGO. We acknowledge the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU. This work was also supported by Ministero delle Imprese e del Made in Italy (IPCEI Cloud DM 27 giugno 2022 – IPCEI-CL-0000007)

References

- [1] Ralph Abboud, Ismail Ilkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising power of graph neural networks with random node initialization. *arXiv preprint arXiv:2010.01179*, 2020.
- [2] László Babai and Ludik Kucera. Canonical labelling of graphs in linear average time. In *20th annual symposium on foundations of computer science (sfcs 1979)*, pages 39–46. IEEE, 1979.
- [3] Muhammet Balcilar, Pierre Héroux, Benoit Gauzere, Pascal Vasseur, Sébastien Adam, and Paul Honeine. Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pages 599–608. PMLR, 2021.
- [4] Fabian Ball and Andreas Geyer-Schulz. How symmetric are real-world graphs? a large-scale study. *Symmetry*, 10(1):29, 2018.
- [5] Maya Bechler-Speicher, Ben Finkelshtein, Fabrizio Frasca, Luis Müller, Jan Tönshoff, Antoine Siraudin, Viktor Zaverkin, Michael M Bronstein, Mathias Niepert, Bryan Perozzi, et al. Position: Graph learning will lose relevance due to poor benchmarks. *arXiv preprint arXiv:2502.14546*, 2025.
- [6] Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=dFbKQaRk15w>.
- [7] Filippo Maria Bianchi and Veronica Lachi. The expressive power of pooling in graph neural networks. *Advances in neural information processing systems*, 36:71603–71618, 2023.
- [8] 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, 2022.
- [9] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a " siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.
- [10] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.
- [11] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.
- [12] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5103–5113, 2021.
- [13] Benjamin Paul Chamberlain, Sergey Shirobokov, Emanuele Rossi, Fabrizio Frasca, Thomas Markovich, Nils Yannick Hammerla, Michael M. Bronstein, and Max Hansmire. Graph neural networks for link prediction with subgraph sketching. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=m1oqEOAozQU>.

- [14] Guangyong Chen, Pengfei Chen, Chang-Yu Hsieh, Chee-Kong Lee, Benben Liao, Renjie Liao, Weiwen Liu, Jiezhong Qiu, Qiming Sun, Jie Tang, et al. Alchemy: A quantum chemistry dataset for benchmarking ai models. *arXiv preprint arXiv:1906.09427*, 2019.
- [15] Xueqi Cheng, Yu Wang, Yunchao Liu, Yuying Zhao, Charu C Aggarwal, and Tyler Derr. Edge classification on graphs: New directions in topological imbalance. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 392–400, 2025.
- [16] Anuj Dawar and Danny Vagnozzi. Generalizations of k-dimensional weisfeiler–leman stabilization. *Moscow Journal of Combinatorics and Number Theory*, 9(3):229–252, 2020.
- [17] Bowen Dong, Charu C Aggarwal, and S Yu Philip. The link regression problem in graph streams. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1088–1095. IEEE, 2019.
- [18] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*, 2019.
- [19] Francesco Ferrini, Antonio Longa, Andrea Passerini, and Manfred Jaeger. Meta-path learning for multi-relational graph neural networks. In *Learning on Graphs Conference*, pages 2–1. PMLR, 2024.
- [20] R. A. Fisher. *Contributions to Mathematical Statistics*. Wiley, 1950.
- [21] Martin Grohe, Moritz Lichter, and Daniel Neuen. The iteration number of the weisfeiler-leman algorithm. *ACM Transactions on Computational Logic*, 26(1):1–31, 2025.
- [22] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9ea9-Paper.pdf.
- [23] Harold Hotelling. The generalization of student’s ratio. *The Annals of Mathematical Statistics*, 2(3):360–378, 1931. doi: 10.1214/aoms/1177732979. URL <https://projecteuclid.org/euclid.aoms/1177732979>.
- [24] Xingyue Huang, Miguel Romero, Ismail Ceylan, and Pablo Barceló. A theory of link prediction via relational weisfeiler-leman on knowledge graphs. *Advances in Neural Information Processing Systems*, 36:19714–19748, 2023.
- [25] Neil Immerman and Rik Sengupta. The k -dimensional weisfeiler-leman algorithm. *arXiv preprint arXiv:1907.09582*, 2019.
- [26] Kanchan Jha, Sriparna Saha, and Hiteshi Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.
- [27] Fabian Jogi, Pascal Welke, and Thomas Gärtner. Is expressivity essential for the predictive performance of graph neural networks? In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL <https://openreview.net/forum?id=5v7hpSy3Ir>.
- [28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [29] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
- [30] 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>.

- [31] Veronica Lachi, Francesco Ferrini, Antonio Longa, Bruno Lepri, and Andrea Passerini. A simple and expressive graph neural network based method for structural link representation. In *Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM) at ICML 2024*, pages 187–201. PMLR, 2024.
- [32] Juanhui Li, Harry Shomer, Haitao Mao, Shenglai Zeng, Yao Ma, Neil Shah, Jiliang Tang, and Dawei Yin. Evaluating graph neural networks for link prediction: Current pitfalls and new benchmarking. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL <https://openreview.net/forum?id=YdjWXrd0Th>.
- [33] Jinbi Liang, Cunlai Pu, Xiangbo Shu, Yongxiang Xia, and Chengyi Xia. Line graph neural networks for link weight prediction. *Physica A: Statistical Mechanics and its Applications*, page 130406, 2025.
- [34] Moritz Lichter, Simon Raßmann, and Pascal Schweitzer. Computational complexity of the weisfeiler-leman dimension. In *33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)*, pages 13–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.
- [35] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.
- [36] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. Simplifying approach to node classification in graph neural networks. *Journal of Computational Science*, 62:101695, 2022.
- [37] 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, pages 4602–4609, 2019.
- [38] Christopher Morris, Fabrizio Frasca, Nadav Dym, Haggai Maron, Ismail Ilkan Ceylan, Ron Levie, Derek Lim, Michael M Bronstein, Martin Grohe, and Stefanie Jegelka. Position: Future directions in the theory of graph machine learning. In *Forty-first International Conference on Machine Learning*, 2024.
- [39] Matthew Morris, Bernardo Cuenca Grau, and Ian Horrocks. Orbit-equivariant graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2023.
- [40] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673. PMLR, 2019.
- [41] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.
- [42] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*.
- [43] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49, 2021.
- [44] Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=SJxzFySKwH>.
- [45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=rJXMpikCZ>.
- [46] Xiyuan Wang, Haotong Yang, and Muhan Zhang. Neural common neighbor with completion for link prediction. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=sNFLN3itAd>.

- [47] Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 international conference on management of data*, pages 2628–2638, 2021.
- [48] Yanbo Wang and Muhan Zhang. An empirical study of realized gnn expressiveness. *arXiv preprint arXiv:2304.07702*, 2023.
- [49] Yanbo Wang and Muhan Zhang. An empirical study of realized GNN expressiveness. In *Forty-first International Conference on Machine Learning*, 2024. URL <https://openreview.net/forum?id=WIaZFk02fI>.
- [50] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16, 1968.
- [51] Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. Graph neural networks in node classification: survey and evaluation. *Machine Vision and Applications*, 33(1):4, 2022.
- [52] Yu Xie, Shengze Lv, Yuhua Qian, Chao Wen, and Jiye Liang. Active and semi-supervised graph neural networks for graph classification. *IEEE Transactions on Big Data*, 8(4):920–932, 2022.
- [53] 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>.
- [54] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.
- [55] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.
- [56] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J Kim. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. *Advances in Neural Information Processing Systems*, 34:13683–13694, 2021.
- [57] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.
- [58] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34:9061–9073, 2021.
- [59] Cai Zhou, Xiyuan Wang, and Muhan Zhang. From relational pooling to subgraph gnns: A universal framework for more expressive graph neural networks. In *International Conference on Machine Learning*, pages 42742–42768. PMLR, 2023.
- [60] Tao Zhou. Progresses and challenges in link prediction. *Iscience*, 24(11), 2021.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The main claims are clearly stated in both the abstract and introduction, with pointers to specific sections that provide theoretical or empirical support. Each claim is substantiated in the body of the paper, ensuring consistency between the stated goals and the actual contributions.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discussed the limitations of our framework in Section 6, addressing both its scope, that is, its focus on message passing models, and a broader limitation related to the scarcity of large high-symmetry datasets for evaluating link-level expressiveness.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix A provides formal proofs for all theoretical results presented in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: For synthetic experiments, we provide details on graph construction, training, and evaluation in Section 4. Additional implementation details are included in Appendix E for both synthetic and real-world experiments. To ensure full reproducibility, we also release the complete source code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release both the code and the datasets, along with usage instructions, to ensure full reproducibility of the experimental results <https://anonymous.4open.science/r/link-representation-gnn-8124/README.md>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify all relevant training and evaluation details, including hyperparameters, data splits, and optimizer settings, in Appendix E.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Result tables report standard deviation as error bars (Appendix F). Unless otherwise specified, results are averaged over 5 runs with different seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Appendix E.2 reports the compute resources used to run our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [\[Yes\]](#)

Justification: The authors have read the NeurIPS Ethics Guidelines. All datasets used are publicly available, and we believe our work poses no immediate harmful consequences.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses the practical benefits of using more expressive models in settings with high graph symmetry, where simpler models may fail (Section 5). No immediate negative societal impacts are anticipated.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our results pose no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credit the original authors of all models and datasets used in our work, and we respect the licenses and terms of use associated with them.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The new assets include our synthetic dataset and the experimental setup on real-world benchmarks, which had not been used in this way before. We release well-documented code and data under a non-restrictive license <https://anonymous.4open.science/r/link-representation-gnn-8124/README.md>.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs were not used as part of the core methods in this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Proofs

Proof of Theorem 3.2. We prove theorem by addressing each of its three components individually.

1. Let $M \in \mathcal{M}$. If $m^M = 0$, then, regardless of k_ϕ^M , M is not able to distinguish between links whose endpoints are automorphic, i.e.,

$$\forall F \in M, \forall (u, v), (u', v') \text{ s.t. } \exists \sigma_1, \sigma_2 \in \Sigma_n^G \text{ with } \sigma_1(u) = u' \wedge \sigma_2(v) = v' \quad (13)$$

$$F((u', v'), \mathbf{X}^0) = F((u, v), \mathbf{X}^0) \quad (14)$$

therefore, automorphic links whose endpoints are automorphic will be assigned to the same representation. Moreover, given $M_1, M_2 \in \mathcal{M}$, if $m^{M_1} = 0$, while $m^{M_2} > 0$ and COMB is injective, regardless of $k_\phi^{M_1}$ and $k_\phi^{M_2}$, $M_1 \leq M_2$.

Proof Suppose $m^M = 0$, i.e., M is composed only of the functions ϕ and g , i.e., its functions are:

$$F((u, v), \mathbf{X}^0) = g(\phi(u, \mathbf{X}^0), \phi(v, \mathbf{X}^0)).$$

Assume ϕ is maximally expressive at the node level (note that no GNN can achieve this, as the k -WL algorithm, preserves graph automorphisms for every k [34, 16, 11]), i.e., it assigns representations to nodes solely based on their automorphism classes. Then for any automorphisms $\sigma_1, \sigma_2 \in \Sigma_n^G$ such that $\sigma_1(u) = u'$ and $\sigma_2(v) = v'$, it follows that

$$\phi(u, \mathbf{X}^0) = \phi(u', \mathbf{X}^0), \quad \phi(v, \mathbf{X}^0) = \phi(v', \mathbf{X}^0).$$

Thus, regardless of the function g , we obtain

$$g(\phi(u, \mathbf{X}^0), \phi(v, \mathbf{X}^0)) = g(\phi(u', \mathbf{X}^0), \phi(v', \mathbf{X}^0)),$$

i.e., the model cannot distinguish between links whose endpoints are automorphic. If (u, v) and (u', v') are not themselves automorphic pairs, then the model fails to differentiate them, thus assigning the same representation to non-automorphic links.

Now consider $M_1 \in \mathcal{M}$ with all functions $F \in M_1$ composed only of ϕ and g , while $M_2 \in \mathcal{M}$ with all functions $F \in M_2$ that also include a component f (on the m -order neighborhood). Using Definition 2.8, we show that M_2 is more expressive.

Let $F_1 \in M_1, F_2 \in M_2, (u, v), (u', v') \in V_G \times V_G$ such that $(u, v) \not\sim (u', v')$ and

$$F_1((u, v), \mathbf{X}^0) = F_1((u', v'), \mathbf{X}^0),$$

but

$$F_2((u, v), \mathbf{X}^0) \neq F_2((u', v'), \mathbf{X}^0).$$

This implies that

$$g(\phi(u, \mathbf{X}^0), \phi(v, \mathbf{X}^0)) = g(\phi(u', \mathbf{X}^0), \phi(v', \mathbf{X}^0)),$$

but due to the presence of a component f and the use of an injective COMB, we get

$$F_2((u, v), \mathbf{X}^0) \neq F_2((u', v'), \mathbf{X}^0),$$

demonstrating by Definition 2.8 that M_2 is more expressive than M_1 .

2. Let $M_1, M_2 \in \mathcal{M}$ with $k_\rho^{M_1} = k_\rho^{M_2} = 1$ -WL and let l^{M_1}, l^{M_2} be the number of layers used respectively by M_1, M_2 . If $m^{M_1} + l^{M_1} \leq m^{M_2} + l^{M_2}$, then $M_1 \leq M_2$.

Proof Let $F_1 \in M_1, F_2 \in M_2$. Assume $l^{M_2} \geq l^{M_1} + m^{M_1} - m^{M_2}$ and $l^{M_1} \leq l^{M_2}$ and $m^{M_1} > m^{M_2}$.

Let $(u, v), (u', v') \in V_G \times V_G$ such that

$$F_1((u, v), G, \mathbf{X}^0) \neq F_1((u', v'), G, \mathbf{X}^0).$$

Let c_u^ℓ denote the color assigned to node u after ℓ iterations of the 1-WL algorithm. Assume $c_u^\ell = c_{u'}^\ell$ and $c_v^\ell = c_{v'}^\ell$; otherwise, F_2 alone could distinguish between (u, v) and (u', v') , and the proof would be complete.

We divide the proof into three steps. Since F_1 distinguishes (u, v) from (u', v') while $c_u^\ell = c_{u'}^\ell$ and $c_v^\ell = c_{v'}^\ell$, this implies that:

$$\left\{ \rho_1(w, \mathbf{X}) \left| w \in \bigcup_{j=0}^{m^{M_1}} N^j(u, v) \right. \right\} \neq \left\{ \rho_1(i, \mathbf{X}) \left| i \in \bigcup_{j=0}^{m^{M_1}} N^j(u', v') \right. \right\} \quad (1)$$

Let's consider the smaller neighborhood m^{M_2} :

$$\left\{ \rho_1(w, \mathbf{X}) \left| w \in \bigcup_{j=0}^{m^{M_2}} N^j(u, v) \right. \right\} \quad \text{and} \quad \left\{ \rho_1(i, \mathbf{X}) \left| i \in \bigcup_{j=0}^{m^{M_2}} N^j(u', v') \right. \right\}.$$

If these sets are already different, then F_2 (with smaller neighborhood but same ρ_1) suffices, and the proof is complete. If instead they are equal, then from (1) it follows that the difference lies in the sets:

$$\left\{ \rho_1(w, \mathbf{X}) \left| w \in \bigcup_{j=m_2+1}^{m^{M_1}} N^j(u, v) \right. \right\} \neq \left\{ \rho_1(i, \mathbf{X}) \left| i \in \bigcup_{j=m_2+1}^{m^{M_1}} N^j(u', v') \right. \right\}.$$

Since F_2 uses a number of layers $l^{M_2} \geq l^{M_1} + m^{M_1} - m^{M_2}$, it can compensate for the smaller neighborhood radius m_2 by using more layers to reach the same depth of information as F_1 . Thus, using ρ_2 in F_2 , we obtain:

$$\left\{ \rho_2(w, \mathbf{X}) \left| w \in \bigcup_{j=0}^{m^{M_2}} N^j(u, v) \right. \right\} \neq \left\{ \rho_2(w, \mathbf{X}) \left| w \in \bigcup_{j=0}^{m^{M_2}} N^j(u', v') \right. \right\},$$

which implies

$$F_2((u, v), \mathbf{X}^0) \neq F_2((u', v'), \mathbf{X}^0),$$

and by Definition 2.8, this yields $M_1 \leq M_2$.

3. Let $M_1, M_2 \in \mathcal{M}$ with $m^{M_1} = m^{M_2}$. If $k_\rho^{M_1} \leq k_\rho^{M_2}$, then $M_1 \leq M_2$.

Proof Let $m^{M_1} = m^{M_2} = m$ and $k_\rho^{M_1} \leq k_\rho^{M_2}$. Let $F_1 \in M_1$ and $F_2 \in M_2$ and $(u, v), (u', v') \in V_G \times V_G$ such that

$$F_1((u, v), G, \mathbf{X}^0) \neq F_1((u', v'), G, \mathbf{X}^0).$$

Assume that $c_u^\ell = c_{u'}^\ell$ and $c_v^\ell = c_{v'}^\ell$, i.e., u and u' , as well as v and v' , have the same WL color after ℓ iterations. Otherwise, F_2 could already distinguish the links based on $k_\rho^{M_2}$ alone, and the proof would be trivial.

Now, since F_1 can distinguish the links despite $c_u^\ell = c_{u'}^\ell$ and $c_v^\ell = c_{v'}^\ell$, this implies that:

$$\left\{ \rho_1(w, G, \mathbf{X}) \left| w \in \bigcup_{j=0}^m N^j(u, v) \right. \right\} \neq \left\{ \rho_1(i, G, \mathbf{X}) \left| i \in \bigcup_{j=0}^m N^j(u', v') \right. \right\} \quad (1)$$

Since $k_\rho^{M_1} \leq k_\rho^{M_2}$, and both models operate on the same neighborhood radius m , it follows that there exists a number of layers (possibly greater than in F_1 [21, 25]) such that F_2 can also distinguish these sets. That is,

$$\left\{ \rho_2(w, G, \mathbf{X}) \left| w \in \bigcup_{j=0}^m N^j(u, v) \right. \right\} \neq \left\{ \rho_2(i, G, \mathbf{X}) \left| i \in \bigcup_{j=0}^m N^j(u', v') \right. \right\},$$

which implies

$$F_2((u, v), G, \mathbf{X}^0) \neq F_2((u', v'), G, \mathbf{X}^0).$$

Therefore, by Definition 2.8, $F_1 \leq F_2$.

□

Proof of Theorem 3.3. 1. This follows directly from Theorem 3.2. Pure GNNs only rely on node representations ϕ and the combining function g , without any function f over the m -order neighborhood. As established in Theorem 3.2, such models cannot distinguish between links whose endpoints are automorphic. In contrast, NCN, Neo-GNN, ELPH, and SEAL incorporate neighborhood-level functions f (all with injective COMB operations, namely concatenation), which increases their expressive power.

2. According to Table 1, Neo-GNN uses a fixed number of message-passing layers $l^{\text{Neo-GNN}} = 1$, while NCN allows l^{NCN} to be freely chosen. Since both models operate with the same type of base function ρ (e.g., 1-WL), Theorem 3.2 implies that if $l^{\text{NCN}} \geq m^{\text{Neo-GNN}}$, then NCN can simulate or surpass Neo-GNN's receptive field and representational capability. Hence, $\text{Neo-GNN} \leq \text{NCN}$.

3. By design, ELPH does not use any message-passing layers ($l^{\text{ELPH}} = 0$, see Table 1), and instead aggregates counts over the m -hop neighborhood. If Neo-GNN uses a neighborhood of size at least $m^{\text{ELPH}} - 1$, then Theorem 3.2 ensures it can access all the information available to ELPH and apply additional learned transformations via its GNN layers. Thus, Neo-GNN is more expressive.

4. ELPH relies only on aggregated neighborhood counts and uses $l^{\text{ELPH}} = 0$ layers. NCN uses a fixed radius $m^{\text{NCN}} = 1$ (Table 1), but allows a configurable number of layers. If $l^{\text{NCN}} \geq m^{\text{ELPH}} - 1$, then by Theorem 3.2 NCN can effectively capture structural signals that ELPH computes with wider but shallower architectures, achieving greater expressiveness.

5. SEAL is more expressive due to both its flexible architecture and enhanced message-passing capabilities. Specifically, SEAL builds a subgraph around the link and augments the adjacency matrix with positional encodings that are link-aware, effectively allowing a learned structural encoding over the link neighborhood $N^m(u, v)$. This corresponds to the expressive power of a higher-order WL test (namely $1-|N^m(u, v)|$ -WL [59]), which is strictly more powerful than 1-WL. Moreover, SEAL allows arbitrary m^{SEAL} and l^{SEAL} , so for any other model with weaker ρ , one can always configure SEAL with the same m and a sufficient number of layers to match or exceed its expressiveness. Prior results have shown that highly expressive architectures may require deeper message passing to converge [21, 25], which SEAL supports by design. □

B Additional Information on LR-EXP

B.1 Evaluation method

This section provides a more detailed explanation of how the two components of the Reliable Paired Comparison work to evaluate different models in the synthetic setting.

Major Procedure Given a graph G , we select two structurally distinct links $e = (u, v), e' = (u', v')$. Then we create q isomorphic copies $\{G_1, G_2, \dots, G_q\}$ by randomly permuting its nodes. For each copy G_i and pair of links e_i, e'_i , we compute the embedding differences:

$$d_i = f(G_i, e_i) - f(G_i, e'_i), \quad i \in [q], \quad (15)$$

where f is the embedding function learned by the model. Assuming that the difference vectors are independent random vectors $\mathcal{N}(\mu, \Sigma)$, we consider that $f(G_i)$ follows a Gaussian distribution, so random permutations only introduce Gaussian noise into the results.

If the model is not able to distinguish links e and e' , the mean difference should be $\mu = 0$. To check this a α -level Hotelling's T-square test is conducted, comparing hypothesis $H_0 : \mu = 0$ and $H_1 : \mu \neq 0$. The T^2 -statistic for μ is calculated as:

$$T^2 = q(\bar{d} - \mu)^T S^{-1}(\bar{d} - \mu), \quad (16)$$

where

$$\bar{d} = \frac{1}{q} \sum_{i=1}^q d_i, \quad \mathbf{S} = \frac{1}{q-1} \sum_{i=1}^q (d_i - \bar{d})(d_i - \bar{d})^T. \quad (17)$$

Hotelling's T-square test establishes that T^2 follows the distribution of a random variable $\frac{(q-1)d}{q-d}F_{d,q-d}$, where $F_{d,q-d}$ represents an F -distribution with degrees of freedom d and $q-d$ [23]. This theorem provides a link between the unknown parameter μ and a specific distribution $F_{d,q-d}$, enabling us to validate the confidence interval of μ by analyzing how well it matches the assumed distribution. To evaluate the hypothesis $H_0 : \mu = 0$, we substitute $\mu = 0$ into Equation (16), leading to:

$$T_{\text{test}}^2 = q\bar{\mathbf{d}}^T S^{-1} \bar{\mathbf{d}}. \quad (18)$$

An α -level test of $H_0 : \mu = 0$ against the alternative $H_1 : \mu \neq 0$ supports H_0 (meaning the model fails to differentiate the pair) if:

$$T_{\text{test}}^2 = q\bar{\mathbf{d}}^T S^{-1} \bar{\mathbf{d}} < \frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha), \quad (19)$$

where $F_{d,q-d}(\alpha)$ represents the upper (100α) -th percentile of the F -distribution $F_{d,q-d}$ [20], with degrees of freedom d and $q-d$. Conversely, we reject H_0 (indicating that the model successfully distinguishes the pair) if:

$$T_{\text{test}}^2 = q\bar{\mathbf{d}}^T S^{-1} \bar{\mathbf{d}} > \frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha). \quad (20)$$

Reliability Check By selecting an appropriate value of α , the Main Procedure ensures a reliable confidence interval to evaluate distinguishability. However, choosing α heuristically may not be the optimal approach. Additionally, computational precision can introduce deviations in the assumed Gaussian fluctuations. To address this issue, we employ the Reliability Check, which captures both external differences between two graphs and internal variations within a single graph.

Without loss of generality (WLOG), we replace (G_i, e'_i) with a permutation of G , denoted as (G^π, e^π) . This allows us to analyze the internal fluctuations of G for the same link. We follow the same steps as in the Main Procedure.

Proceeding with (G_i, e_i) and (G_i^π, e_i^π) , we compute the T^2 -statistic as:

$$T_{\text{reliability}}^2 = q\bar{\mathbf{d}}^T S^{-1} \bar{\mathbf{d}}, \quad (21)$$

where

$$\bar{d} = \frac{1}{q} \sum_{i=1}^q d_i, \quad d_i = f(G_i, e_i) - f(G_i^\pi, e_i^\pi), \quad i \in [q], \quad (22)$$

$$S = \frac{1}{q-1} \sum_{i=1}^q (d_i - \bar{d})(d_i - \bar{d})^T. \quad (23)$$

Since G and G^π are isomorphic, the GNN should not differentiate between them, implying that $\mu = 0$. Consequently, the test is considered reliable only if

$$T_{\text{reliability}}^2 < \frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha). \quad (24)$$

By combining the reliability check with the distinguishability results, we obtain the full RPC Reliability Procedure Check.

For each pair of structurally different links e and e' , the threshold is computed as:

$$\text{Threshold} = \frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha). \quad (25)$$

Next, we perform the Main Procedure on e and e' to evaluate distinguishability, and we conduct the Reliability Check on e and e^π . The GNN is considered capable of distinguishing e and e' only if both conditions

$$T_{\text{reliability}}^2 < \text{Threshold} < T_{\text{test}}^2 \quad (26)$$

are satisfied.

B.2 Experimental Results with different significance level values

We tested the sensitivity of results of Section 4.3 to the choice of the significance level in the reliability check. Specifically, we experimented with three commonly used values in statistical testing, i.e., . The resulting performance on LR-EXP are reported in Table 4.

As expected, all standard GNN models consistently fail to recognize any links, regardless of the significance level. Among the more expressive models, the majority show stable behavior across different values of significance, with fluctuations that remain within the standard deviation. *BUDDY* and *NCN* exhibit slightly larger fluctuations in absolute value, but the key point for this experiment is the expressiveness ranking across models. This ranking remains robust across all significance levels, with one minor exception: at significance level 0.1, *Neo-GNN* slightly surpasses *NCN*. However, this difference falls within the standard deviation range.

Table 4: Test Precision (mean $\pm$ std over 5 runs) for significance level values 0.1, 0.05, 0.01.

Model	0.1	0.05	0.01
<i>GCN</i>	0 ± 0	0 ± 0	0 ± 0
<i>GAT</i>	0 ± 0	0 ± 0	0 ± 0
<i>GAE</i>	0 ± 0	0 ± 0	0 ± 0
<i>SAGE</i>	11 ± 0	8 ± 4	3 ± 0
<i>BUDDY</i>	54 ± 5	45 ± 1	59 ± 1
<i>ELPH</i>	61 ± 1	62 ± 7	62 ± 1
<i>Neo-GNN</i>	76 ± 6	75 ± 2	71 ± 2
<i>NCN</i>	70 ± 1	75 ± 1	81 ± 1
<i>SEAL</i>	92 ± 0	97 ± 0	97 ± 0

C Further Experiments on LR-EXP

In the experiment of Section 4.3, to ensure a fair comparison, we selected for each model the maximum neighborhood radius that makes it as expressive as possible while still respecting its design constraints. For example, in models like *NCN*, the radius is fixed by design and equal to 1, so it cannot be varied. In others, such as *ELPH*, the radius is tunable but is constrained to a maximum of 3 by the original authors for computational reasons, and we adopted that upper bound. As for GNN depth, we fixed the number of layers to 3 as commonly done in the literature [42]. To explore this further, we conducted two additional experiments on the LR-EXP dataset: 1) **Fixed radius** $m = 1$, varying the number of layers (see Table 5); 2) **Fixed number of layers** $l = 3$, varying the neighborhood radius (see Table 6).

Table 5: 1) Fixed Radius = 1

# Layers	<i>NCN</i>	<i>Neo-GNN</i>	<i>ELPH</i>	<i>SEAL</i>
1	74 ± 1	42 ± 4	40 ± 4	79 ± 3
2	75 ± 1	42 ± 4	55 ± 4	89 ± 1
3	75 ± 1	45 ± 3	62 ± 2	89 ± 2

These results are fully consistent with Theorem 3.2 (Point 2): when fixing the radius, increasing the number of layers improves expressiveness in all the models; when fixing the number of layers, increasing the radius also improves expressiveness in all the models.

Table 6: 2) Fixed # Layers = 3

Radius	<i>NCN</i>	<i>Neo-GNN</i>	<i>ELPH</i>	<i>SEAL</i>
1	75 ± 1	45 ± 3	62 ± 2	89 ± 2
2	—	63 ± 3	61 ± 4	90 ± 3
3	—	75 ± 2	62 ± 7	97 ± 0

D Computational Costs

The overall time complexity of models for link prediction can be expressed as $O(B + Ct)$, where t is the number of links to be predicted, B is the preprocessing cost and C per-link cost [46]. In Table 7 we report the time complexity of existing models, where n is the number of nodes, d is the maximum node degree, h is the complexity of the hash function used in *BUDDY*, f is the feature dimension and m is the radius of the neighborhood.

Table 7: Time complexity of models expressed as $O(B + Ct)$, where t is the number of links to be predicted, B is the preprocessing cost and C per-link cost.

Method	B	C
GAE	$ndf + nf^2$	f^2
Neo-GNN	$ndf + nf^2 + nd^m$	$d^m + f^2$
BUDDY	$ndf + nh$	$h + f^2$
SEAL	0	$d^{(m+1)}f + d^m f^2$
NCN	$ndf + nf^2$	$df + f^2$
NCNC	$ndf + nf^2$	$d^2 f + df^2$

Among these, *SEAL* is the most expressive but also the most computationally expensive, due to its exponential dependence on the neighborhood radius. *NCN* and *BUDDY* provide the best trade-off between expressiveness and computational cost. *Neo-GNN*, while more expensive than *NCN*, is in fact less expressive. *GAE* is the least costly, but also the least expressive.

E Experimental details

E.1 Real world datasets

Table 8 presents the statistics of the standard datasets commonly used for link prediction tasks. The Cora, Citeseer, and Pubmed datasets are well-known citation networks, often employed as benchmarks for graph-based learning methods. These datasets are relatively small in scale, both in terms of nodes and edges. In contrast, the OGB (Open Graph Benchmark) datasets including ogbl-collab, ogbl-ddi, ogbl-ppa, and ogbl-citation2 are significantly larger and more complex, providing challenging benchmarks for evaluating scalability and performance on large graphs.

For Cora, Citeseer, and Pubmed, we adopt a fixed train/validation/test split of 85/5/10%. For the OGB datasets, we use the official splits provided by the OGB benchmark.

Table 8: Statistics of datasets. The split ratio is for train/validation/test.

Dataset	Cora	Citeseer	Pubmed	ogbl-collab	ogbl-ddi	ogbl-ppa	ogbl-citation2
#Nodes	2,708	3,327	18,717	235,868	4,267	576,289	2,927,963
#Edges	5,278	4,676	44,327	1,285,465	1,334,889	30,326,273	30,561,187
Mean Degree	3.90	2.81	4.74	10.90	625.68	105.25	20.88
Split Ratio	85/5/10	85/5/10	85/5/10	92/4/4	80/10/10	70/20/10	98/1/1

Table 9 reports the statistics of additional datasets characterized by high symmetry, which are also used in our experiments. These datasets span various domains, including social networks, biological systems, and knowledge graphs, offering diverse structural properties and serving as valuable benchmarks for evaluating models under highly regular and repetitive connection patterns.

Specifically, the *AIFB*⁵ dataset models the organizational structure of the AIFB research institute, capturing relationships among its staff, research groups, and publications. *Edit-TSW*² models user interactions and content editing activities within the Wiktionary platform. Finally, the *NSC*⁶, *YST*³, and *GRQ*³ datasets, introduced in [12], focus on relational learning tasks and exhibit highly symmetric structures, making them particularly suitable for evaluating link prediction models.

Table 9: Statistics of datasets. The split ratio is for train/validation/test.

Dataset	NSC	YST	GRQ	aifb	edit-tsw
#Nodes	332	2,284	5,241	8,285	1,079
#Edges	2,126	6,646	14,484	46,042	2,756
Mean Degree	12.81	5.82	5.53	5.56	5.11
Split Ratio	80/10/10	80/10/10	80/10/10	80/10/10	80/10/10

E.2 Implementation details

This section provides hyperparameters details for all models trained and evaluated on both LR-EXP and the real-world datasets. All experiments were conducted on a workstation running Ubuntu 22.04 with an AMD Ryzen 9 7950X CPU (32 threads), 124GB of RAM, and two NVIDIA GeForce RTX 4090 GPUs (24GB each).

Synthetic Hyperparameter Settings We present the hyperparameter search space for the LR-EXP datasets in Table 10. For each model, we initially followed the hyperparameter configurations proposed in their respective papers. However, due to the small size of the synthetic graphs in LR-EXP, we significantly reduced the embedding dimensions, which led to improved performance in most cases.

Table 10: Hyperparameter Search Ranges for LR-EXP. Abbreviations: LR = Learning Rate, Drop. = Dropout, WD = Weight Decay, #L = Number of Model Layers, #P = Number of Prediction Layers, Dim = Embedding Dimension.

Dataset	LR	Drop.	WD	#L	#P	Dim
LR-EXP	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(2–256)

Real-World Hyperparameter Settings We report the hyperparameter search space for all real-world datasets used in our experiments in Table 11. For *Cora*, *Citeseer*, *Pubmed*, *ogbl-collab*, *ogbl-ddi*, *ogbl-ppa*, and *ogbl-citation2*, we adopt the hyperparameter settings declared in the paper [32]. For the remaining datasets, *NSC*, *YST*, *GRQ*, *aifb*, and *edit-tsw*, we conducted a hyperparameter search using the ranges specified in the table.

F Real-world results with standard deviations

In this section, we present the same results as in Table 3, now including standard deviations computed over 5 runs with different random seeds. Specifically, Table 12 reports the MRR results on standard small datasets, Table 13 shows the MRR results on the OGB benchmark, and Table 14 reports the MRR results for highly symmetric datasets.

Hits@k. We also report the Hits@ k metric, using $k = 10$ for the standard small datasets (Table 15) and for the new highly symmetric dataset (Table 16), and $k = 20$ for the larger datasets (Table 17), following the evaluation protocol of Li et al. [32].

⁵<https://pytorch-geometric.readthedocs.io/>

⁶<https://github.com/LeiCaiwsu/LGLP/tree/main/LGLP/Python/data>

Table 11: Hyperparameter search ranges for all real-world datasets. Abbreviations: LR = Learning Rate, Drop. = Dropout, WD = Weight Decay, #L = Number of Model Layers, #P = Number of Prediction Layers, Dim = Embedding Dimension.

Dataset	LR	Drop.	WD	#L	#P	Dim
Cora	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(128, 256)
Citeseer	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(128, 256)
Pubmed	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(128, 256)
ogbl-collab	(0.01, 0.001)	(0, 0.3, 0.5)	0	3	3	256
ogbl-ddi	(0.01, 0.001)	(0, 0.3, 0.5)	0	3	3	256
ogbl-ppa	(0.01, 0.001)	(0, 0.3, 0.5)	0	3	3	256
ogbl-citation2	(0.01, 0.001)	(0, 0.3, 0.5)	0	3	3	128
NSC	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(64, 128, 256)
YST	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(64, 128, 256)
GRQ	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(64, 128, 256)
aifb	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(64, 128, 256)
edit-tsw	(0.01, 0.001)	(0.1, 0.3, 0.5)	(1e-4, 1e-7, 0)	(1, 2, 3)	(1, 2, 3)	(64, 128, 256)

Table 12: MRR with standard deviations on real-world link prediction datasets.

Models	NSC	YST	GRQ	AIFB	EDIT-TSW
$\hat{r}_G$	0.16	0.24	0.30	0.41	0.67
GCN	26.79 (± 1.08)	1.17 (± 0.02)	6.89 (± 0.47)	10.85 (± 0.12)	4.12 (± 0.21)
GAT	2.47 (± 3.58)	0.40 (± 0.00)	0.63 (± 0.40)	1.07 (± 0.51)	0.40 (± 0.00)
SAGE	20.32 (± 0.11)	1.77 (± 1.38)	1.34 (± 1.32)	0.54 (± 0.08)	8.35 (± 7.27)
GAE	16.61 (± 10.00)	2.32 (± 0.01)	2.32 (± 0.01)	10.98 (± 0.01)	3.54 (± 0.01)
BUDDY	18.40 (± 0.07)	13.68 (± 0.61)	46.23 (± 0.42)	13.73 (± 0.11)	22.66 (± 0.46)
ELPH	26.09 (± 1.04)	13.15 (± 1.12)	38.88 (± 0.55)	13.39 (± 0.60)	10.10 (± 7.37)
Neo-GNN	22.86 (± 0.75)	5.78 (± 0.12)	29.51 (± 0.38)	13.11 (± 0.51)	8.10 (± 0.29)
NCN	30.36 (± 0.28)	11.99 (± 0.06)	48.45 (± 0.02)	6.47 (± 0.23)	10.32 (± 0.39)
NCNC	30.20 (± 0.03)	13.54 (± 0.47)	47.60 (± 0.52)	OOM	10.30 (± 0.39)
SEAL	30.85 (± 1.46)	17.51 (± 0.94)	56.72 (± 1.35)	16.30 (± 0.22)	25.82 (± 1.47)

Table 13: MRR with standard deviations on real-world link prediction datasets.

Models	CORA	CITeseer	PUBMED
$\hat{r}_G$	0.02	0.02	0.08
GCN	16.61 (± 0.30)	21.09 (± 0.88)	7.13 (± 0.27)
GAT	13.84 (± 0.68)	19.58 (± 0.84)	4.95 (± 0.14)
SAGE	14.74 (± 0.69)	21.09 (± 1.15)	9.40 (± 0.70)
GAE	18.32 (± 0.41)	25.25 (± 0.82)	5.27 (± 0.25)
BUDDY	13.71 (± 0.59)	22.84 (± 0.36)	7.56 (± 0.18)
Neo-GNN	13.95 (± 0.39)	17.34 (± 0.84)	7.74 (± 0.30)
NCN	14.66 (± 0.95)	28.65 (± 1.21)	5.84 (± 0.22)
NCNC	14.98 (± 1.00)	24.10 (± 0.65)	8.58 (± 0.59)
SEAL	10.67 (± 3.46)	13.16 (± 1.66)	5.88 (± 0.53)

Table 14: MRR with standard deviations on real-world link prediction datasets.

Models	OGBL-CITATION2	OGBL-DDI	OGBL-PPA	OGBL-COLLAB
$\hat{r}_G$	0.01	0.10	0.12	0.13
GCN	19.98 (± 0.35)	13.46 (± 0.34)	26.94 (± 0.48)	6.09 (± 0.38)
GAT	22.05 (± 0.12)	12.92 (± 0.39)	27.27 (± 0.30)	4.18 (± 0.33)
SAGE	OOM	12.60 (± 0.72)	OOM	5.53 (± 0.50)
GAE	OOM	3.49 (± 1.73)	OOM	OOM
BUDDY	19.17 (± 0.20)	12.43 (± 0.50)	27.00 (± 0.33)	5.67 (± 0.36)
Neo-GNN	16.12 (± 0.25)	10.86 (± 2.16)	21.68 (± 1.14)	5.23 (± 0.90)
NCN	23.35 (± 0.28)	12.86 (± 0.78)	35.06 (± 0.26)	5.09 (± 0.38)
NCNC	19.61 (± 0.54)	>24h	33.52 (± 0.26)	4.73 (± 0.86)
SEAL	20.60 (± 1.28)	9.99 (± 0.90)	29.71 (± 0.71)	6.43 (± 0.32)

Table 15: Hits@10 with standard deviations on real-world link prediction datasets.

Models $\hat{r}_G$	NSC 0.16	YST 0.24	GRQ 0.30	AIFB 0.41	EDIT-TSW 0.67
<i>GCN</i>	43.08 (± 1.36)	0.75 (± 0.00)	13.47 (± 2.97)	17.56 (± 0.92)	5.91 (± 0.97)
<i>GAT</i>	2.36 (± 4.09)	0.00 (± 0.00)	0.44 (± 0.76)	1.51 (± 1.73)	0.00 (± 0.00)
<i>SAGE</i>	24.06 (± 3.68)	3.56 (± 3.96)	2.44 (± 4.17)	0.41 (± 0.38)	10.67 (± 10.07)
<i>GAE</i>	30.38 (± 9.90)	3.38 (± 0.00)	3.38 (± 0.00)	16.92 (± 0.09)	3.80 (± 0.00)
<i>BUDDY</i>	28.93 (± 2.68)	22.89 (± 0.69)	67.26 (± 0.25)	24.28 (± 0.57)	36.00 (± 2.27)
<i>ELPH</i>	38.37 (± 0.72)	22.34 (± 1.15)	59.44 (± 0.99)	26.19 (± 1.83)	21.45 (± 15.19)
<i>Neo-GNN</i>	37.11 (± 0.72)	10.29 (± 0.17)	44.50 (± 0.24)	24.29 (± 1.74)	12.36 (± 1.09)
<i>NCN</i>	47.17 (± 0.94)	20.03 (± 0.00)	68.67 (± 0.14)	9.42 (± 0.76)	16.48 (± 0.56)
<i>NCNC</i>	46.86 (± 0.27)	22.44 (± 1.04)	68.28 (± 0.23)	OOM	15.03 (± 1.87)
<i>SEAL</i>	51.26 (± 0.98)	28.36 (± 1.46)	71.64 (± 1.04)	31.84 (± 0.70)	42.55 (± 2.39)

Table 16: Hits@10 with standard deviations on real-world link prediction datasets.

Models $\hat{r}_G$	CORA 0.02	CITeseer 0.02	PUBMED 0.08
<i>GCN</i>	36.26 (± 1.14)	47.23 (± 1.88)	15.22 (± 0.57)
<i>GAT</i>	32.89 (± 1.27)	45.30 (± 1.30)	9.99 (± 0.64)
<i>SAGE</i>	34.65 (± 1.47)	48.75 (± 1.85)	20.54 (± 1.40)
<i>GAE</i>	37.95 (± 1.24)	49.65 (± 1.48)	10.50 (± 0.46)
<i>BUDDY</i>	30.40 (± 1.18)	48.35 (± 1.18)	16.78 (± 0.53)
<i>Neo-GNN</i>	31.27 (± 0.72)	41.74 (± 1.18)	17.88 (± 0.71)
<i>NCN</i>	35.14 (± 1.04)	53.41 (± 1.46)	13.22 (± 0.56)
<i>NCNC</i>	36.70 (± 1.57)	53.72 (± 0.97)	18.81 (± 1.16)
<i>SEAL</i>	24.27 (± 6.74)	27.37 (± 3.20)	12.47 (± 1.23)

Table 17: Hits@20 with standard deviations on real-world link prediction datasets.

Models $\hat{r}_G$	OGBL-CITATION2 0.01	OGBL-DDI 0.10	OGBL-PPA 0.12	OGBL-COLLAB 0.13
<i>GCN</i>	51.72 (± 0.46)	64.76 (± 1.45)	68.38 (± 0.73)	22.48 (± 0.81)
<i>GAT</i>	53.13 (± 0.15)	66.83 (± 2.23)	69.49 (± 0.43)	18.30 (± 1.42)
<i>SAGE</i>	OOM	67.19 (± 1.18)	OOM	21.26 (± 1.32)
<i>GAE</i>	OOM	17.81 (± 9.80)	OOM	OOM
<i>BUDDY</i>	47.81 (± 0.37)	58.71 (± 1.63)	71.50 (± 0.68)	23.35 (± 0.73)
<i>Neo-GNN</i>	43.17 (± 0.53)	51.94 (± 10.33)	64.81 (± 2.26)	21.03 (± 3.39)
<i>NCN</i>	53.76 (± 0.20)	65.82 (± 2.66)	81.89 (± 0.31)	20.84 (± 1.31)
<i>NCNC</i>	51.69 (± 1.48)	>24h	82.24 (± 0.40)	20.49 (± 3.97)
<i>SEAL</i>	48.62 (± 1.93)	49.74 (± 2.39)	76.77 (± 0.94)	21.57 (± 0.38)