# GNNs Meet Sequence Models Along the Shortest-Path: an Expressive Method for Link Prediction

**Francesco Ferrini**[*]
University of Trento
Trento, Italy
`francesco.ferrini@unitn.it`

**Veronica Lachi**[*]
Fondazione Bruno Kessler
Trento, Italy
`vlachi@fbk.eu`

**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`

## Abstract

Graph Neural Networks (GNNs) often struggle to capture the link-specific structural patterns crucial for accurate link prediction, as their node-centric message-passing schemes overlook the subgraph structures connecting a pair of nodes. Existing methods to inject such structural context either incur high computational cost or rely on simplistic heuristics (e.g., common neighbor counts) that fail to model multi-hop dependencies. We introduce SP4LP (Shortest Path for Link Prediction), a novel framework that combines GNN-based node encodings with sequence modeling over shortest paths. Specifically, SP4LP first applies a GNN to compute representations for all nodes, then extracts the shortest path between each candidate node pair and processes the resulting sequence of node embeddings using a sequence model. This design enables SP4LP to capture expressive multi-hop relational patterns with computational efficiency. Empirically, SP4LP achieves state-of-the-art performance across link prediction benchmarks. Theoretically, we prove that SP4LP is strictly more expressive than standard message-passing GNNs and several state-of-the-art structural features methods, establishing it as a general and principled approach for link prediction in graphs.

## 1 Introduction

Graph Neural Networks (GNNs) are widely adopted for link-level tasks such as link prediction [33, 18, 35], link classification [21, 28, 3] and link regression [16, 6] with applications spanning recommender systems [31], knowledge graph completion [20], and biological interaction prediction [12].

Despite their popularity, standard GNNs struggle to accurately represent links, as they typically construct link embeddings by aggregating the representations of the two endpoint nodes. This node-centric strategy leads to a key limitation: structurally distinct links may be mapped to the same representation when their endpoints are automorphic [22, 2, 34]. For example, in the graph of Figure 1, links $(v, u)$ and $(v, u')$ yield identical representations under any standard GNN, even if one pair shares a common neighbor and the other does not. This issue, known as the automorphic node problem [2], highlights an expressivity bottleneck in message-passing schemes for link representation.

---

[*] Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Graph Machine Learning.

To address this, several methods enhance GNNs with structural features (SFs), which can be broadly classified into three paradigms [27]: SF-then-GNN, which injects structural context into the graph before message passing (e.g., SEAL [34], NBFNet [36]); SF-and-GNN, which computes SFs and node embeddings in parallel (e.g., Neo-GNN [32], BUDDY [2]); and GNN-then-SF, which applies message passing once to compute node representations and then combines them using task-specific structural context (e.g., NCN and NCNC [27]).

While SF-then-GNN methods are expressive, they are computationally inefficient, often requiring subgraph extraction or retraining per link. SF-and-GNN models are efficient but rely on predefined heuristics, limiting their ability to capture rich relational patterns. GNN-then-SF approaches offer a compelling trade-off between expressivity and scalability, but current methods in this class, i.e., NCN and NCNC, depend on the presence of common neighbors between link endpoints. When such neighbors are absent, they revert to standard GNN behavior and lose expressive power.

In this paper we propose SP4LP, a novel method in the GNN-then-SF paradigm that combines high expressiveness with computational efficiency. SP4LP constructs a path-aware representation by incorporating the embeddings of all nodes along the shortest path connecting the two endpoints. These node embeddings, obtained via a base GNN, are then processed as a sequence using a dedicated sequence model, such as a Transformer [23], LSTM [10]. Unlike structural features such as common neighbors, which may not exist for many node pairs and can lead to degenerate cases in sparse graphs, the shortest path is always defined between any two nodes if the graph is connected. Shortest paths are more broadly defined, as common neighbors imply a path, but not vice versa. We formally prove that SP4LP is strictly more expressive than existing SF-and-GNN and GNN-then-SF approaches.

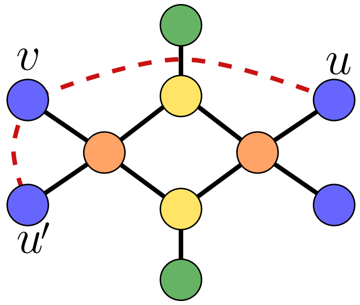

Figure 1: Links $(v, u)$ and $(v, u')$ have different structural roles within the graph, yet a GNN assigns them identical representations.

SP4LP achieves **state-of-the-art performance** across several benchmark datasets. Under the challenging HeaRT evaluation protocol [15], it consistently outperforms existing link prediction methods.

## 2 Preliminaries

**Definition 2.1** (*graph*)**.** A **graph** is a tuple $G = (V, E, \mathbf{X}^0)$ where $V = \{1, \ldots, n\}$ is a set of nodes, $E \subseteq V \times V$ is a set of edges and $\mathbf{X}^0 \in \mathbb{R}^{n \times f}$ is the node features matrix. To each graph is associated an adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ with $\mathbf{A}_{i,j} = 1$ if and only if $(i, j) \in E$. In this work, we consider simple, finite and undirected graphs.

**Definition 2.2** (*message passing*)**.** Let $G = (V, E, \mathbf{X}^0)$ be a graph. In **message passing** scheme, representation of nodes $v \in V$ is iteratively updated as follows:

$$\mathbf{x}_v^0 = \mathbf{X}^0_{[v,:]} \tag{1}$$

$$\mathbf{x}_v^l = \text{UPDATE}\left(\mathbf{x}_v^{l-1}, \text{AGGREGATE}\left(\{\mathbf{x}_u^{l-1} \mid u \in N(v)\}\right)\right) \tag{2}$$

where $N(v)$ is the first-order neighborhood of node $v$.

GNNs are a class of neural architectures that operate on graphs by iteratively updating node representations through the message passing scheme. It has been proven that GNNs are at most as effective as the Weisfeiler–Lehman (WL) test in distinguishing between graphs [19, 29].

**Definition 2.3** (GNN link representation model)**.** A **GNN link representation model** $M$ is a class of functions $F : ((u, v), G) \mapsto \mathbf{x}_{(u,v)} \in \mathbb{R}^d$ which maps node pairs in $(u, v) \in V \times V$ to vector representations using the message passing scheme defined in Definition 2.2.

Note that the pair $(u, v)$ belongs to $V \times V$, meaning the model learns representations for all possible pairs (links), not just those connected by an edge. A widely adopted approach for learning such representations is what we refer to as a *pure GNN*, defined as follows:

**Definition 2.4** (*pure GNN*). A **pure GNN** model calculates representation $\mathbf{x}_{(u,v)} \in \mathbb{R}^d$ for each pair of nodes $(u, v)$ with $u, v \in V$ as follows:

$$\mathbf{x}_{(u,v)} = g(\mathbf{x}_u^L, \mathbf{x}_v^L) \tag{3}$$

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 $L$ layers of message passing as defined in Definition 2.2.

Pure GNNs are inherently limited in terms of expressiveness. In particular even when the base GNN is the most powerful, they may assign the same representation to structurally different links. We provide a formal definition of what it means for two links to be different.

**Definition 2.5** (*node permutation*). A **node permutation** $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$ 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 $\Pi_n$. Given a subset of nodes $S \subseteq V$, we define the permutation $\pi$ on $S$ as $\pi(S) := \{\pi(i) | i \in S\}$. Additionally, we define $\pi(\mathbf{A})$ as the matrix $\mathbf{A}$ with rows and columns permutated based on $\pi$, i.e., $\pi(\mathbf{A})_{\pi(i),\pi(j)} = \mathbf{A}_{i,j}$.

**Definition 2.6** (*automorphism*). An **automorphism** on the graph $G = (V, E, \mathbf{X}^0)$ is a permutation $\sigma \in \Pi_n$ such that $\sigma(\mathbf{A}) = \mathbf{A}$. All the possible automorphisms on a graph constitute the automorphism group $\Sigma_n^G$.

**Definition 2.7** (*automorphic nodes*). Let $G = (V, E, \mathbf{X}^0)$ be a graph and $\Sigma_n^G$ its automorphism group. Two nodes $u, v \in V$ are said to be **automorphic nodes** ( $u \simeq v$ ) if:

$$\exists \sigma \in \Sigma_n^G \quad \text{s.t.} \quad \sigma(\{u\}) = \{v\}. \tag{4}$$

**Definition 2.8** (*automorphic links*). Let $G = (V, E, \mathbf{X}^0)$ be a graph and $\Sigma_n^G$ its automorphism group. Two pairs of nodes $(u, v), (u', v') \in V \times V$ are said to be **automorphic links** ( $(u, v) \simeq (u', v')$ ) if:

$$\exists \sigma \in \Sigma_n^G \quad \text{s.t.} \quad \sigma(\{u, v\}) = \{u', v'\}. \tag{5}$$

**Proposition 2.9.** *Pure GNN methods suffer from the automorphic node problem, i.e., for any graph $G = (V, E, \mathbf{X}^0)$, for pairs of links $(u, v), (u', v') \in V \times V$ such that there exist $\sigma_1 \in \Sigma_n^G$ and $\sigma_2 \in \Sigma_n^G$ with $\sigma_1(u) = u'$ and $\sigma_2(v) = v'$, $\mathbf{x}_{(u,v)} = \mathbf{x}_{(u',v')}$, independently whether $(u, v)$ and $(u', v')$ are isomorphic, i.e, whether exist $\sigma \in \Sigma_n^G$ with $\sigma(\{u, v\}) = \{u', v'\}$.*

This limitation does not arise from the expressiveness bounds of GNNs, which are constrained by the WL test. Even considering higher-order GNNs, i.e., $k$-GNN [19], automorphic nodes will be assigned to the same representation as the $k$-WL algorithm preserves graph automorphisms for every $k$ [17, 5, 1]. To tackle this, several models have been proposed that enhance message passing by incorporating structural features [27, 36, 2, 26, 34], thereby increasing the expressive power. We provide a formal definition of what it means for one link representation model to be more expressive and strictly more expressive than another.

**Definition 2.10** (*more expressive*). Let $M_1$ and $M_2$ be two link representation models (Def. 2.3). $M_2$ is **more expressive** than $M_1$ ($M_1 \preceq M_2$) if, for any graph $G = (V, E, \mathbf{X}^0)$ and any pair $(u, v), (u', v') \in V \times V$ with $(u, v) \not\simeq (u', v')$:

$$\exists F_1 \in M_1 : F_1((u, v), G) \neq F_1((u', v'), G) \Rightarrow \exists F_2 \in M_2 : F_2((u, v), G) \neq F_2((u', v'), G). \tag{6}$$

**Definition 2.11** (*strictly more expressive*). Let $M_1$ and $M_2$ be two link representation models (Def. 2.3). We say that $M_2$ is **strictly more expressive** than $M_1$ ($M_1 \prec M_2$) if:

- $M_2$ is more expressive than $M_1$ (Def. 2.10), and
- there exists a graph $G = (V, E, \mathbf{X}^0)$ and a pair of links $(u, v), (u', v') \in V \times V$ with $(u, v) \not\simeq (u', v')$ such that:

$$\begin{aligned} &\forall F_1 \in M_1 : \quad F_1((u, v), G) = F_1((u', v'), G) \\ \text{and} \quad &\exists F_2 \in M_2 : \quad F_2((u, v), G) \neq F_2((u', v'), G). \end{aligned}$$

In the following section, we introduce our model SP4LP and demonstrate its improved expressive power in distinguishing structurally different links.

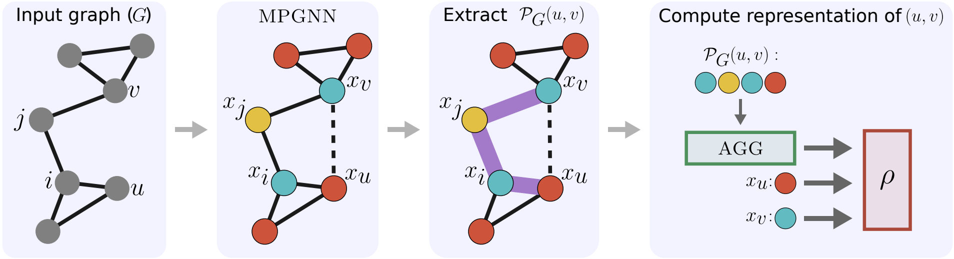

Figure 2: Overview of the SP4LP framework. First, a GNN is used to compute contextualized embeddings for all nodes in the graph. Then, for each target link, the shortest path connecting the two endpoints is extracted. The embeddings of the nodes along this path are passed to a sequence model (e.g., Transformer or LSTM) to compute a path-aware link representation.

## 3 SP4LP: An Expressive SF-then-GNN Model for link Representation

We propose a model, SP4LP, which adheres to the GNN-then-SF framework, but incorporates additional pairwise information by encoding the sequence of node embeddings along the shortest path connecting the endpoints. This design choice provides a crucial advantage in terms of expressiveness: unlike common neighbors, the shortest path is always defined for nodes in a connected graph and captures richer structural patterns, even in sparse or incomplete graphs. In the following, we introduce the necessary definitions, formally describe the model, and present theoretical results characterizing its expressive power.

**Definition 3.1** (*path*)**.** Let $G = (V, E, \mathbf{X}^0)$ be a graph and $u, v \in V$ to nodes. A **path** in $G$ from $u$ to $v$ is a sequence of nodes $P = (u_0, u_1, \ldots, u_k)$ with (i) $u_i \in V$ for all $i = 0, \ldots, k-1$, (ii) $u_0 = u$ and $u_k = v$, (iii) $(u_i, u_{i+1}) \in E$ for all $i = 0, \ldots, k-1$ , and (iv) all nodes in the sequence are distinct (i.e., $u_i \neq u_j$ for all $i \neq j$). The length of a path $P$, $\text{len}(P)$ is the number of edges it contains.

**Definition 3.2** (*shortest path length*)**.** Let $\mathcal{P}_G(u, v)$ denote the set of all paths from $u$ to $v$ in $G$. The **shortest path length** $d_G(u, v)$ is the minimum length among all paths i.e., $d_G(u, v) = \min_{P \in \mathcal{P}(u,v)} \text{len}(P)$.

**Definition 3.3** (*shortest path*)**.** A **shortest path** between $u$ to $v$ in $G$ is any path $P^* \in \mathcal{P}_G(u, v)$ such that $\text{len}(P^*) = d_G(u, v)$. The set of all the shortest path from $u$ to $v$ in $G$ is denoted as $\mathcal{P}^*_G(u, v)$.

Let $G = (V, E, \mathbf{X}^0)$ be a graph, $u, v \in V$. SP4LP is a GNN link representation model (see Definition 2.3) that computes link representation as follows:

$$\text{SP4LP}((u,v), G) = \rho\left(\text{GNN}(u, G), \text{GNN}(v, G), \text{AGG}\left(\left\{\phi\left(\text{GNN}(u_i, G)\right)_{i=1}^{k} \mid (u_i)_{i=1}^{k} \in \mathcal{P}^*_G\{u, v\}\right\}\right)\right) \tag{7}$$

where $k = d_G(u, v)$, $\text{GNN}(u, G) \in \mathbb{R}^d$ is the representation of node $u \in V$ obtained ad the final layer of message passing as in Definition 2.2, $\phi : \mathbb{R}^{k \times d} \to \mathbb{R}^d$ is a sequence model on the GNN representations of nodes in the shortest path from $u$ to $v$, AGG is an aggregation function over multiset of shortest paths representations and $\rho : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ combine the endpoint nodes representations with the shortest paths representation to get a final link representation. Since we consider undirected graphs, we consider $\mathcal{P}^*_G\{u, v\} := \mathcal{P}^*_G(u, v) \cup \mathcal{P}^*_G(v, u)$.

For a graph with $n$ nodes and $m$ edges, the shortest paths from a single source node can be computed via a breadth-first search (BFS) [4] in $O(m)$ time. Consequently, computing shortest paths between all pairs of nodes requires $O(nm)$ time overall. In sparse graphs, where $m = O(n)$, this yields a quadratic cost $O(n^2)$, which remains tractable in practice. Notably, this computation is performed only once as a **preprocessing step** and can be amortized across multiple downstream predictions.

SP4LP is a general framework: the GNN component can be instantiated with architectures such as GCN [13], GAT [24] or GraphSAGE [9], while the sequence model can range from simple aggregation functions like injective summation as the one proposed in Xu et al. [29], to more complex architectures such as LSTMs [10] or Transformers [23]. An overview of SP4LP is illustrated in Figure 2.

**Proposition 3.4.** *SP4LP does not suffer from the automorphic node problem.*

Table 1: MRR and Hits@K (%) results across all datasets, following the HeaRT evaluation setting [15]. The top three results for each metric are highlighted using **first**, **second**, and **third**. *OOM* indicates that the model ran out of memory, while *>24h* denotes that the method did not complete within 24 hours. Standard deviations over 5 runs are reported in the appendix B.

| | **Models** | **Cora** | | **Citeseer** | | **Pubmed** | | **Ogbl-ddi** | | **Ogbl-collab** | | **Ogbl-ppa** | | **Ogbl-Citation2** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MRR | Hits@10 | MRR | Hits@10 | MRR | Hits@10 | MRR | Hits@20 | MRR | Hits@20 | MRR | Hits@20 | MRR | Hits@20 |
| GNN | GCN | 16.61 | 36.26 | 21.09 | 47.23 | 7.13 | 15.22 | **13.46** | 64.76 | 6.09 | **22.48** | 26.94 | 68.38 | 19.98 | 51.72 |
| | GAT | 13.84 | 32.89 | 19.58 | 45.30 | 4.95 | 9.99 | 12.92 | **66.83** | 4.18 | 18.30 | *OOM* | *OOM* | *OOM* | *OOM* |
| | SAGE | 14.74 | 34.65 | 21.09 | 48.75 | **9.40** | **20.54** | 12.60 | **67.19** | 5.53 | 21.26 | 27.27 | 69.49 | **22.05** | **53.13** |
| | GAE | **18.32** | **37.95** | 25.25 | 49.65 | 5.27 | 10.50 | 3.49 | 17.81 | *OOM* | *OOM* | *OOM* | *OOM* | *OOM* | *OOM* |
| SF GNN | SEAL | 10.67 | 24.27 | 13.16 | 27.37 | 5.88 | 12.47 | 9.99 | 49.74 | **6.43** | **21.57** | 29.71 | 76.77 | 20.60 | 48.62 |
| | BUDDY | 13.71 | 30.40 | 22.84 | 48.35 | 7.56 | 16.78 | 12.43 | 58.71 | 5.67 | **23.35** | 27.70 | 71.50 | 19.17 | 47.81 |
| | Neo-GNN | 13.95 | 31.27 | 17.34 | 41.74 | 7.74 | 17.88 | 10.86 | 51.94 | 5.23 | 21.03 | 21.68 | 64.81 | 16.12 | 43.17 |
| | NCN | 14.66 | 35.14 | **28.65** | **53.41** | 5.84 | 13.22 | 12.86 | **65.82** | 5.09 | 20.84 | **35.06** | **81.89** | **23.35** | **53.76** |
| | NCNC | 14.98 | **36.70** | 24.10 | **53.72** | 8.58 | 18.81 | *>24h* | *>24h* | 4.73 | 20.49 | **33.52** | **82.24** | 19.61 | 51.69 |
| | NBFNet | 13.56 | 31.12 | 14.29 | 31.39 | *>24h* | *>24h* | *>24h* | *>24h* | *OOM* | *OOM* | *OOM* | *OOM* | *OOM* | *OOM* |
| | PEG | 15.73 | 36.03 | 21.01 | 45.56 | 4.40 | 8.70 | 12.05 | 50.12 | 4.83 | 18.29 | *OOM* | *OOM* | *OOM* | *OOM* |
| | SP4LP (our) | **17.27** | **38.52** | **41.08** | **66.28** | **10.87** | **23.01** | **15.00** | 47.96 | **9.46** | 20.00 | **36.45** | **76.90** | **24.91** | **55.45** |

The proof can be found in Appendix A. As an example of non-automorphic links composed of automorphic nodes that SP4LP can successfully distinguish, consider the links $(v, u)$ and $(v, u')$ shown in Figure 1. While $u' \simeq u'$ via the identity, and $v \simeq u$ via an automorphism induced by a vertical axis of symmetry (i.e., a mirror reflection), the links $(v, u)$ and $(v, u')$ are not automorphic. This asymmetry is captured by the distinct shortest paths between the endpoints: the shortest path from $v$ to $u'$ consists of $v$, and orange node, and $u'$, whereas the shortest path from $v$ to $u$ includes $v$, and orange node, a yellow node, another orange node, and finally $u$. In addition to overcoming this limitation, SP4LP is strictly more expressive than several state-of-the-art message passing methods for link representation learning.

**Theorem 3.5.** *SP4LP is strictly more expressive than Pure GNNs, NCN, BUDDY, NBFnet and Neo-GNN.*

The proof can be found in Appendix A. In the following sections, we complement the theoretical analysis with an extensive experimental evaluation, showing that SP4LP also achieves state-of-the-art performance on standard link prediction benchmarks.

## 4 Experiments

We evaluate the performance of SP4LP on real-world link prediction benchmarks against several baselines. In particular, we use Cora, Citeseer, and Pubmed [30] as well as datasets from Open Graph Benchmark [11]. For Cora, Citeseer, and Pubmed, we use a single fixed data split in all experiments. Table 6 in appendix C provides a summary of dataset statistics.

As baseline methods we consider two class of models: 1) **Pure GNN methods**: Graph Convolutional Network (GCN) [13], Graph Attention Network (GAT) [25], GraphSAGE [8], and Graph Autoencoder (GAE) [14]; 2) **Structural Features GNN methods**: SEAL [33], BUDDY [2], Neo-GNN [32], NBFNet [36], NCN [27], NCNC [27] and PEG [26].

Importantly, as described in Section 3, SP4LP is a general framework that allows for different choices of both the underlying GNN architecture and the sequence model ($\phi$ Equation 7). In our experimental setting, we treat the choice of GNN and the choice of $\phi$ as hyperparameters, and perform hyperparameter tuning based on validation set performance. Specifically, we explore GCN, GAT, and GraphSAGE as GNN backbones, and LSTM, Transformer, and an injective sum Xu et al. [29] aggregator as sequence models. Moreover, we choose an MLP for $\rho$ and as AGG we choose to select the first shortest path retrieved by the BFS procedure for computational efficiency. Appendix D provides implementation details. Code to reproduce all experiments is available at[2].

**Evaluation Setting** We evaluate model performance under the more challenging and realistic HeaRT evaluation setting [15]. We adopt two standard ranking metrics: Hits@K and Mean Reciprocal Rank (MRR). Following the HeaRT protocol, we report Hits@10 and MRR for Cora, Citeseer, and Pubmed, and Hits@20 along with MRR for ogbl-collab and ogbl-ddi.

---

[2] `https://anonymous.4open.science/r/sp4lp-3875/README.md`

**Results on Real-World Benchmarks** As shown in Table 1, SP4LP ranks first in terms of MRR on four out of five datasets and second on the remaining one. The improvements in MRR are often substantial: on Citeseer, for instance, SP4LP achieves a 43% gain over the second-best method, NCN. SP4LP also achieves the best Hits@K score on three out of five datasets. On the Ogbl-Collab dataset, SP4LP is comparable on Hits@20 to the third-best model (SEAL), when accounting for standard deviations (Appendix B). On Ogbl-ddi, where SP4LP performs worse, the lower score can be explained by the lack of node features. Our model benefits from the availability of node features, as it leverages nodes representations obtained via message passing. In settings where such features are absent, like in Ogbl-ddi, the discriminative power of the learned representations is reduced. In addition to achieving the best performance in several datasets, SP4LP is also the most consistent model across all benchmarks.

**Ablation Study** We perform an ablation study to evaluate the contribution of the main components of our model. In particular, we investigate two simplified variants: **(1) Sequence Model Only**: in this variant, the sequential model operates directly on the raw input features of the nodes along the shortest path, without incorporating node representations learned by the GNN. **(2) GNN + Shortest Path Length:** in this variant, the sequential model is completely removed and link prediction is performed using only the learned node representations from the GNN, combined with the length of the shortest path between the target nodes.

Table 2: Ablation study results (%). Standard deviations over 5 runs are reported in Appendix B

| **Models** | **Cora** | | **Citeseer** | | **Pubmed** | |
|---|---|---|---|---|---|---|
| | MRR | Hits@10 | MRR | Hits@10 | MRR | Hits@10 |
| *GNN + SP len.* | 14.21 | 33.43 | 20.90 | 47.82 | 7.12 | 5.63 |
| *Sequence Model* | 16.86 | 36.03 | 27.45 | 54.20 | 8.58 | 12.87 |
| SP4LP | **17.27** | **38.52** | **41.08** | **66.28** | **10.87** | **23.01** |

Table 2 reports the results, conducted on Cora, Citeseer, and Pubmed. Both variants show a performance drop compared to the full model, demonstrating the importance of jointly leveraging node representations and sequential modeling of the shortest paths.

**Scalability Analysis** We assess the scalability of SP4LP by examining how its GPU memory consumption and inference time evolve as the batch size increases, in comparison to several baseline methods. The results, presented in Figure 3, highlight the superior resource efficiency of SP4LP across a wide range of batch sizes. In terms of GPU memory usage, SP4LP exhibits remarkable efficiency: memory consumption remains nearly constant across small to medium batch sizes, and increases moderately only for the largest batches. PEG also maintains low memory usage; however, this advantage is undermined by its impractically slow inference. SEAL, while competitive with SP4LP in terms of inference speed, suffers from excessive memory consumption.

Considering inference time, SP4LP matches the efficiency of SEAL. Although NCN and Buddy consistently achieve low inference times, this comes at the cost of substantially lower predictive performance, as shown in Table 1.

In summary, SP4LP achieves an excellent balance between low memory consumption, fast inference, high predictive accuracy, and strong model expressiveness. It effectively scales to large batch sizes where alternative approaches either become prohibitively slow, fail due to memory constraints, or cannot deliver competitive results.

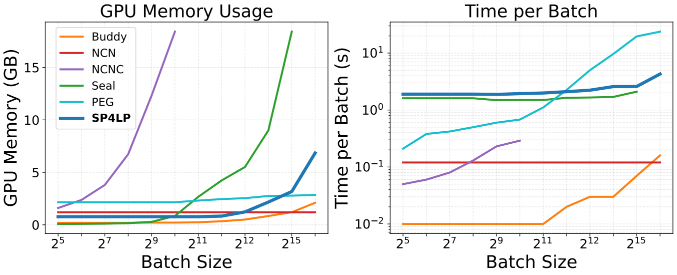

Figure 3: Inference time and GPU memory usage on ogbl-collab, measured during the prediction of a single batch of test links.

## 5 Conclusion

We introduced SP4LP, a novel message-passing based framework for link representation that enhances the expressiveness of standard GNNs by incorporating sequential modeling over the shortest path between target nodes. We formally proved that SP4LP is strictly more expressive than several state-of-the-art link representation models. Extensive experiments under the HeaRT evaluation protocol confirm that SP4LP achieves state-of-the-art performance across diverse datasets.

## 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] 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.

[2] 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`.

[3] 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.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 3rd edition, 2009.

[5] Anuj Dawar and Danny Vagnozzi. Generalizations of k-dimensional weisfeiler–leman stabilization. *Moscow Journal of Combinatorics and Number Theory*, 9(3):229–252, 2020.

[6] 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.

[7] Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

[8] 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/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf`.

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

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[12] Kanchan Jha, Sriparna Saha, and Hiteshi Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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

[15] 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=YdjWXrdOTh`.

[16] 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.

[17] 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.

[18] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[19] 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.

[20] 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.

[21] 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.

[22] Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. *arXiv preprint arXiv:1910.00452*, 2019.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[24] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[25] 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`.

[26] Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=e95i1IHcWj`.

[27] 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`.

[28] 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.

[29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.

[30] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[31] 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.

[32] 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.

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

[34] 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.

[35] Tao Zhou. Progresses and challenges in link prediction. *Iscience*, 24(11), 2021.

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

## 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 5.

   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: Implementation details are included in Appendix D. 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 the full source code along with usage instructions to ensure the reproducibility of our experiments. All datasets used in this study are publicly available and properly cited in the manuscript. `https://anonymous.4open.science/r/sp4lp-3875/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 D.

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 (Appendix 5).

   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 D 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 expressive models in link prediction (Section 4). 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 model. We release well-documented code and data under a non-restrictive license `https://anonymous.4open.science/r/sp4lp-3875/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

**Proposition 3.4** SP4LP does not suffer from the automorphic node problem.

*Proof.* According to Proposition 2.9, a model $M$ suffers from the *automorphic node problem* if, for any graph $G = (V, E, \mathbf{X}^0)$, for any pairs of links $(u, v), (u', v') \in V \times V$, for any $F \in M$ it holds that:

$$(u, v) \not\simeq (u', v'), \quad u \simeq u', \quad v \simeq v', \quad \text{and} \quad F((u, v), G) \neq F((u', v'), G).$$

In order to prove that SP4LP does not suffer from the automorphic node problem, it suffices to provide an example of a graph $G = (V, E, X^0)$ and node pairs $(u, v), (u', v') \in V \times V$ such that:

$$(u, v) \not\simeq (u', v'), \quad u \simeq u', \quad v \simeq v', \quad \text{and} \quad \text{SP4LP}((u, v), G) \neq \text{SP4LP}((u', v'), G).$$

Such an example is provided in Figure 1: the shortest path from $v$ to $u'$ consists of $v$, an orange node, and $u'$, whereas the shortest path from $v$ to $u$ includes $v$, an orange node, a yellow node, another orange node, and finally $u$. Thus, simply using a summation as function $\phi$, leads to distinct representations for links $(v, u)$ and $(v, u')$.

□

**Theorem 3.5** SP4LP is strictly more expressive than Pure GNN, NCN, BUDDY, NBFnet and Neo-GNN.

*Proof.* We proceed by prove each comparison separately.

- SP4LP is strictly more expressive than Pure GNN.

  We prove this by noting that SP4LP architecture (Equation 7) generalizes that of pure GNNs, meaning that SP4LP can simulate any pure GNN by simply ignoring the shortest path information. Thus, for any pair of non-automorphic links that a specific pure GNN can distinguish, there exists a configuration of SP4LP that distinguishes them as well. We can conclude that SP4LP is strictly more expressive than pure GNNs considering as example of pair of links indistinguishable by GNNs but distinguishable by SP4LP the one provided is provided in the proof of Proposition 3.4.

- SP4LP is strictly more expressive than NCN.

  We prove this in two steps: (1) When two links share no common neighbors, NCN on them reduces to a pure GNN. As we have already proved that SP4LP is strictly more expressive than pure GNNs, it follows that SP4LP is also strictly more expressive than NCN in this case. (2) When the links have common neighbors, setting AGG as summation, $\rho$ as the Hadamard product between the endpoint representations and the concatenation with the result of the aggregation, and $\phi$ as the identity function, SP4LP reduces exactly to NCN. Therefore, if NCN can distinguish two links under some configuration, SP4LP can as well. By definition 2.10), this implies that SP4LP is more expressive than NCN. We can conclude that SP4LP is strictly more expressive than NCN considering the example in Figure 4. The graph is regular and thus all nodes receive the same embedding from a GNN. Consider nodes $u$ and $v$: $N(u) \cap N(v) = N(u) \cap N(v') = \emptyset$. In this case, NCN reduces to a pure GNN and is thus unable to distinguish the links $(u, v)$ and $(u', v')$. Now, let $\mathbf{x} \in \mathbb{R}^d$ be the representation assigned to every node by a GNN. Then, the representation of the shortest path $\mathcal{P}^*_G(u, v)$ is simply $(\mathbf{x}, \mathbf{x}, \mathbf{x})$, while $\mathcal{P}^*_G(u', v') = (\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$. Even using a simple sum as aggregation function, SP4LP successfully distinguishes between the two links.

- SP4LP is strictly more expressive than Neo-GNN and BUDDY.

  We have already shown that SP4LP is strictly more expressive than NCN. In Theorem 2 of the NCN paper [27], it has been proved that NCN is more expressive than both Neo-GNN and BUDDY. It follows that SP4LP is strictly more expressive than Neo-GNN and BUDDY as well.

- SP4LP is strictly more expressive than NBFNet.

  We prove that NBFNet is as expressive as a pure GNN. Therefore, since we have already proven that SP4LP is strictly more expressive than pure GNNs, it immediately follows that SP4LP is also strictly more expressive than NBFNet.

  To complete the argument, we prove that NBFNet is as expressive as a pure GNN. First of all we report the formulation of NBFNet for simple undirected graph. Given a graph $G = (V, E, \mathbf{X}^0)$, NBFNet assigns a representation $\mathbf{x}(u, v)$ to each edge $(u, v) \in E$. The iterative update rule follows a message-passing scheme:

  $$\begin{aligned} \mathbf{x}^{(0)}_{(u,v)} &= \text{INDICATOR}(\mathbf{x}^0_u, \mathbf{x}^0_v), \\ \mathbf{x}^{(l)}_{(u,v)} &= \text{AGGREGATE}\left(\left\{\text{MESSAGE}(h^{(l-1)}_{(i,j)}) \mid (i,j) \in N(u,v)\right\} \cup \{h^{(0)}_{(u,v)}\}\right) \end{aligned} \tag{8}$$

  where INDICATOR assigns an initial representation based on the nodes $u, v \in V$ and $N(u, v)$ is the set of edges incident to $(u, v)$.

  We prove that, at any layer $l$, the representations of the two links produced by a pure GNN are equal if and only if also the ones produced by NBFNet are equal, i.e.,

  $$\mathbf{x}^{\text{GNN}^l}_{(u,v)} = \mathbf{x}^{\text{GNN}^l}_{(i,j)} \Leftrightarrow \mathbf{x}^{\text{NBF}^l}_{(u,v)} = \mathbf{x}^{\text{NBF}^l}_{(i,j)} \quad \forall l \tag{9}$$

  where $\mathbf{x}^{\text{NBF}^l}_{(u,v)}$ and $\mathbf{x}^{\text{NBF}^l}_{(i,j)}$ are calculated via Equation 8, while $\mathbf{x}^{\text{GNN}^l}_{(u,v)}$ and $\mathbf{x}^{\text{GNN}^l}_{(i,j)}$ are calculated following the standard message passing scheme reported below:

  $$\begin{aligned} \mathbf{x}^{\text{GNN}^0}_v &= \mathbf{x}^0_v, \\ \mathbf{x}^{\text{GNN}^l}_v &= \text{COMB}\left(\mathbf{x}^{\text{GNN}^{l-1}}_v, \text{AGG}\left(\{\mathbf{x}^{\text{GNN}^{l-1}}_u \mid u \in N(v)\}\right)\right) \\ \mathbf{x}^{\text{GNN}^l}_{(u,v)} &= g(\mathbf{x}^{\text{GNN}^l}_u, \mathbf{x}^{\text{GNN}^l}_v) \end{aligned} \tag{10}$$

  Let the functions INDICATOR, AGGREGATE and MESSAGE of Equation 8, as well as the functions COMB, AGG and $g$ be injective. We prove Equation 9 by induction on the number of layer $l$.

  **Base Case:** $l = 0$

  $$\mathbf{x}^{\text{GNN}^0}_{(u,v)} = \mathbf{x}^{\text{GNN}^0}_{(i,j)} \overset{(10)}{\Longleftrightarrow} g(\mathbf{x}^0_u, \mathbf{x}^0_v) = g(\mathbf{x}^0_i, \mathbf{x}^0_j) \overset{\text{inj}}{\Longleftrightarrow} (\mathbf{x}^0_u, \mathbf{x}^0_v) = (\mathbf{x}^0_i, \mathbf{x}^0_j) \overset{\text{inj}}{\Longleftrightarrow}$$

  $$\overset{\text{inj}}{\Longleftrightarrow} \text{INDICATOR}(\mathbf{x}^0_u, \mathbf{x}^0_v) = \text{INDICATOR}(\mathbf{x}^0_i, \mathbf{x}^0_j) \overset{(8)}{\Longleftrightarrow} \mathbf{x}^{\text{NBF}^0}_{(u,v)} = \mathbf{x}^{\text{NBF}^0}_{(i,j)}$$

  **Inductive Step** We assume Equation 9 holds for $l - 1$ and prove it holds for $l$.

  In particular, we want to prove

  $$\mathbf{x}^{\text{GNN}^l}_{(u,v)} = \mathbf{x}^{\text{GNN}^l}_{(i,j)} \Longleftrightarrow \mathbf{x}^{\text{NBF}^l}_{(u,v)} = \mathbf{x}^{\text{NBF}^l}_{(i,j)} \tag{11}$$

  using the inductive hypothesis

  $$\mathbf{x}^{\text{GNN}^{l-1}}_{(u,v)} = \mathbf{x}^{\text{GNN}^{l-1}}_{(i,j)} \Longleftrightarrow \mathbf{x}^{\text{NBF}^{l-1}}_{(u,v)} = \mathbf{x}^{\text{NBF}^{l-1}}_{(i,j)} \tag{12}$$

  Applying Equation 10 to the left-hand side of Equation 11 we get

  $$\mathbf{x}^{\text{GNN}^l}_{(u,v)} = \mathbf{x}^{\text{GNN}^l}_{(i,j)}$$

  $$\overset{(10)}{\Longleftrightarrow}$$

  $$g(\text{COMB}(\mathbf{x}^{\text{GNN}^{l-1}}_u, \text{AGG}(\{\mathbf{x}^{\text{GNN}^{l-1}}_x \mid x \in N(u)\})), \text{COMB}(\mathbf{x}^{\text{GNN}^{l-1}}_v, \text{AGG}(\{\mathbf{x}^{\text{GNN}^{l-1}}_y \mid y \in N(u)\})))$$

  $$=$$

  $$g(\text{COMB}(\mathbf{x}^{\text{GNN}^{l-1}}_i, \text{AGG}(\{\mathbf{x}^{\text{GNN}^{l-1}}_m \mid m \in N(i)\})), \text{COMB}(\mathbf{x}^{\text{GNN}^{l-1}}_j, \text{AGG}(\{\mathbf{x}^{\text{GNN}^{l-1}}_n \mid n \in N(j)\}))).$$

Given the injectivity of $g$, COMB and AGG, this is equivalent to

$$
\begin{gathered}
\mathbf{x}_u^{\text{GNN}^{l-1}} = \mathbf{x}_i^{\text{GNN}^{l-1}} \wedge \{\mathbf{x}_x^{\text{GNN}^{l-1}} \mid x \in N(u)\} = \{\mathbf{x}_m^{\text{GNN}^{l-1}} \mid m \in N(i)\} \wedge \\
\wedge\, \mathbf{x}_v^{\text{GNN}^{l-1}} = \mathbf{x}_j^{\text{GNN}^{l-1}} \wedge \{\mathbf{x}_y^{\text{GNN}^{l-1}} \mid y \in N(v)\} = \{\mathbf{x}_n^{\text{GNN}^{l-1}} \mid n \in N(j)\} \\
\Longleftrightarrow \\
\{(\mathbf{x}_u^{\text{GNN}^{l-1}}, \mathbf{x}_x^{\text{GNN}^{l-1}}) \mid x \in N(u)\} = \{(\mathbf{x}_i^{\text{GNN}^{l-1}}, \mathbf{x}_m^{\text{GNN}^{l-1}}) \mid m \in N(i)\} \\
\wedge \\
\{(\mathbf{x}_v^{\text{GNN}^{l-1}}, \mathbf{x}_y^{\text{GNN}^{l-1}}) \mid y \in N(v)\} = \{(\mathbf{x}_j^{\text{GNN}^{l-1}}, \mathbf{x}_n^{\text{GNN}^{l-1}}) \mid n \in N(j)\} \\
\Longleftrightarrow \\
\{(\mathbf{x}_u^{\text{GNN}^{l-1}}, \mathbf{x}_x^{\text{GNN}^{l-1}}) \mid x \in N(u)\} \cup \{(\mathbf{x}_v^{\text{GNN}^{l-1}}, \mathbf{x}_y^{\text{GNN}^{l-1}}) \mid y \in N(v)\} \\
= \\
\{(\mathbf{x}_i^{\text{GNN}^{l-1}}, \mathbf{x}_m^{\text{GNN}^{l-1}}) \mid m \in N(i)\} \cup \{(\mathbf{x}_j^{\text{GNN}^{l-1}}, \mathbf{x}_n^{\text{GNN}^{l-1}}) \mid n \in N(j)\}.
\end{gathered}
$$

By Definition of $N(u, v)$ (Equation 8), this is equivalent to

$$
\begin{gathered}
\{(\mathbf{x}_w^{\text{GNN}^{l-1}}, \mathbf{x}_t^{\text{GNN}^{l-1}}) \mid (w,t) \in N(u,v)\} = \{(\mathbf{x}_a^{\text{GNN}^{l-1}}, \mathbf{x}_b^{\text{GNN}^{l-1}}) \mid (a,b) \in N(i,j)\} \\
\overset{\text{inj}}{\Longleftrightarrow} \\
\{g(\mathbf{x}_w^{\text{GNN}^{l-1}}, \mathbf{x}_t^{\text{GNN}^{l-1}}) \mid (w,t) \in N(u,v)\} = \{g(\mathbf{x}_a^{\text{GNN}^{l-1}}, \mathbf{x}_b^{\text{GNN}^{l-1}}) \mid (a,b) \in N(i,j)\} \\
\overset{(10)}{\Longleftrightarrow} \\
\{\mathbf{x}_{(w,t)}^{\text{GNN}^{l-1}} \mid (w,t) \in N(u,v)\} = \{\mathbf{x}_{(a,b)}^{\text{GNN}^{l-1}} \mid (a,b) \in N(i,j)\} \\
\overset{\text{IND. HYP.}(12)}{\Longleftrightarrow} \\
\{\mathbf{x}_{(w,t)}^{\text{NBF}^{l-1}} \mid (w,t) \in N(u,v)\} = \{\mathbf{x}_{(a,b)}^{\text{NBF}^{l-1}} \mid (a,b) \in N(i,j)\}.
\end{gathered}
$$

Using the hypotheses of injective AGG and MESSAGE, this is equivalent to:

$$
\begin{gathered}
\text{AGG}(\{\text{MESSAGE}(\mathbf{x}_{(w,t)}^{\text{NBF}^{l-1}}) \mid (w,t) \in N(u,v)\}) = \text{AGG}(\{\text{MESSAGE}(\mathbf{x}_{(a,b)}^{\text{NBF}^{l-1}}) \mid (a,b) \in N(i,j)\}) \\
\overset{(8)}{\Longleftrightarrow} \\
\mathbf{x}_{(u,v)}^{\text{NBF}^{l}} = \mathbf{x}_{(i,j)}^{\text{NBF}^{l}} \qquad (13)
\end{gathered}
$$

which complete the proof.

□

# B Additional results

We complement the main results of Section 4 with additional tables reporting the standard deviation computed over five different random seeds, to better assess the stability of each method.

**Real-world Datasets: Results with Standard Deviations** Table 3 and Table 4 expand the main results in Table 1 by including both the mean and standard deviation of MRR and Hits@$K$ across runs.

**Ablation Study: Results with Standard Deviations** Similarly, Table 5 complements the ablation results in Table 2 by reporting mean and standard deviation for Cora, Citeseer, and Pubmed.

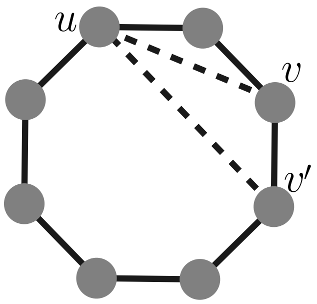

Figure 4: Links $(u, v), (u, v')$ are not distinguished by NCN while are distinguished by SP4LP.

| Models | Cora<br>MRR | Citeseer<br>MRR | Pubmed<br>MRR | Ogbl-ddi<br>MRR | Ogbl-collab<br>MRR |
|---|---|---|---|---|---|
| CN | 9.78 | 8.42 | 2.28 | 7.11 | 4.20 |
| AA | 11.91 | 10.82 | 2.63 | 7.37 | 5.07 |
| RA | 11.81 | 10.84 | 2.47 | 9.10 | **6.29** |
| Shortest Path | 5.04 | 5.83 | 0.86 | 0 | 3.06 |
| Katz | 11.41 | 11.19 | 3.01 | 7.11 | 6.31 |
| Node2Vec | 14.47 (± 0.60) | 21.17 (± 1.01) | 3.94 (± 0.24) | 11.14 (± 0.95) | 4.68 (± 0.08) |
| MF | 6.20 (± 1.42) | 7.80 (± 0.79) | 4.46 (± 0.32) | **13.99 (± 0.47)** | 4.89 (± 0.25) |
| MLP | 13.52 (± 0.65) | 22.62 (± 0.55) | 6.41 (± 0.25) | N/A | 5.37 (± 0.14) |
| GCN | **16.61 (± 0.30)** | 21.09 (± 0.88) | 7.13 (± 0.27) | **13.46 (± 0.34)** | 6.09 (± 0.38) |
| GAT | 13.84 (± 0.68) | 19.58 (± 0.84) | 4.95 (± 0.14) | 12.92 (± 0.39) | 4.18 (± 0.33) |
| SAGE | 14.74 (± 0.69) | 21.09 (± 1.15) | **9.40 (± 0.70)** | 12.60 (± 0.72) | 5.53 (± 0.50) |
| GAE | **18.32 (± 0.41)** | **25.25 (± 0.82)** | 5.27 (± 0.25) | 3.49 (± 1.73) | OOM |
| SEAL | 10.67 (± 3.46) | 13.16 (± 1.66) | 5.88 (± 0.53) | 9.99 (± 0.90) | **6.43 (± 0.32)** |
| BUDDY | 13.71 (± 0.59) | 22.84 (± 0.36) | 7.56 (± 0.18) | 12.43 (± 0.50) | 5.67 (± 0.36) |
| Neo-GNN | 13.95 (± 0.39) | 17.34 (± 0.84) | 7.74 (± 0.30) | 10.86 (± 2.16) | 5.23 (± 0.90) |
| NCN | 14.66 (± 0.95) | **28.65 (± 1.21)** | 5.84 (± 0.22) | 12.86 (± 0.78) | 5.09 (± 0.38) |
| NCNC | 14.98 (± 1.00) | 24.10 (± 0.65) | **8.58 (± 0.59)** | >24h | 4.73 (± 0.86) |
| NBFNet | 13.56 (± 0.58) | 14.29 (± 0.80) | >24h | >24h | OOM |
| PEG | 15.73 (± 0.39) | 21.01 (± 0.77) | 4.40 (± 0.41) | 12.05 (± 1.14) | 4.83 (± 0.21) |
| SP4LP | **17.27 (± 0.57)** | **41.08 (± 1.84)** | **10.87 (± 0.31)** | **15.00 (± 0.57)** | **9.46 (± 0.55)** |

Table 3: MRR results across all datasets, following the HeaRT evaluation setting [15]. The top three results for each metric are highlighted using **first**, **second**, and **third**. *OOM* indicates that the model ran out of memory, while *>24h* denotes that the method did not complete within 24 hours.

## C Datasets statistics

Table 6 summarizes the main datasets used in our link prediction experiments. Cora, Citeseer, and Pubmed are well-known citation networks frequently used as benchmarks for graph-based learning methods. These datasets are relatively small, both in the number of nodes and edges. In contrast, the datasets from the Open Graph Benchmark (OGB), namely ogbl-collab and ogb-ddi, are substantially larger and more complex, offering challenging scenarios for evaluating model scalability and performance on large-scale 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 data splits provided by the OGB benchmark.

| Models | Cora<br>Hits@10 | Citeseer<br>Hits@10 | Pubmed<br>Hits@10 | Ogbl-ddi<br>Hits@20 | Ogbl-collab<br>Hits@20 |
|---|---|---|---|---|---|
| CN | 20.11 | 18.68 | 4.78 | 39.09 | 16.46 |
| AA | 24.10 | 22.20 | 5.51 | 40.15 | 19.59 |
| RA | 24.48 | 22.86 | 4.90 | 44.01 | **24.29** |
| Shortest Path | 15.37 | 16.26 | 0.38 | 0 | 16.38 |
| Katz | 22.77 | 24.84 | 5.98 | 39.09 | 24.34 |
| Node2Vec | 32.77 (± 1.29) | 45.82 (± 2.01) | 8.51 (± 0.77) | 63.63 (± 2.05) | 16.84 (± 0.17) |
| MF | 15.26 (± 3.39) | 16.72 (± 1.99) | 9.42 (± 0.80) | 59.50 (± 1.68) | 18.86 (± 0.40) |
| MLP | 31.01 (± 1.71) | 48.02 (± 1.79) | 15.04 (± 0.67) | N/A | 16.15 (± 0.27) |
| GCN | 36.26 (± 1.14) | 47.23 (± 1.88) | 15.22 (± 0.57) | 64.76 (± 1.45) | **22.48 (± 0.81)** |
| GAT | 32.89 (± 1.27) | 45.30 (± 1.30) | 9.99 (± 0.64) | **66.83 (± 2.23)** | 18.30 (± 1.42) |
| SAGE | 34.65 (± 1.47) | 48.75 (± 1.85) | **20.54 (± 1.40)** | **67.19 (± 1.18)** | 21.26 (± 1.32) |
| GAE | **37.95 (± 1.24)** | 49.65 (± 1.48) | 10.50 (± 0.46) | 17.81 (± 9.80) | OOM |
| SEAL | 24.27 (± 6.74) | 27.37 (± 3.20) | 12.47 (± 1.23) | 49.74 (± 2.39) | 21.57 (± 0.38) |
| BUDDY | 30.40 (± 1.18) | 48.35 (± 1.18) | 16.78 (± 0.53) | 58.71 (± 1.63) | **23.35 (± 0.73)** |
| Neo-GNN | 31.27 (± 0.72) | 41.74 (± 1.18) | 17.88 (± 0.71) | 51.94 (± 10.33) | 21.03 (± 3.39) |
| NCN | 35.14 (± 1.04) | **53.41 (± 1.46)** | 13.22 (± 0.56) | **65.82 (± 2.66)** | 20.84 (± 1.31) |
| NCNC | **36.70 (± 1.57)** | **53.72 (± 0.97)** | **18.81 (± 1.16)** | >24h | 20.49 (± 3.97) |
| NBFNet | 31.12 (± 0.75) | 31.39 (± 1.34) | >24h | >24h | OOM |
| PEG | 36.03 (± 0.75) | 45.56 (± 1.38) | 8.70 (± 1.26) | 50.12 (± 6.55) | 18.29 (± 1.06) |
| SP4LP | **38.52 (± 1.19)** | **66.28 (± 0.63)** | **23.01 (± 0.39)** | 47.96 (± 3.82) | 20.00 (± 1.20) |

Table 4: Hits@K (%) results across all datasets, following the HeaRT evaluation setting [15]. The top three results for each metric are highlighted using **first**, **second**, and **third**. *OOM* indicates that the model ran out of memory, while *>24h* denotes that the method did not complete within 24 hours.

| **Models** | **Cora** | | **Citeseer** | | **Pubmed** | |
|---|---|---|---|---|---|---|
| | MRR | Hits@10 | MRR | Hits@10 | MRR | Hits@10 |
| *GNN + SP len.* | 14.21 (± 1.44) | 33.43 (± 2.69) | 20.90 (± 0.79) | 47.82 (± 1.11) | 7.12 (± 0.41) | 5.63 (± 0.52) |
| *Sequence Model* | 16.86 (± 1.26) | 36.03 (± 1.75) | 27.45 (± 1.55) | 54.20 (± 2.35) | 8.58 (± 0.75) | 12.87 (± 0.85) |
| SP4LP | **17.27 (± 0.57)** | **38.52 (± 1.19)** | **41.08 (± 1.84)** | **66.28 (± 0.63)** | **10.87 (± 0.31)** | **23.01 (± 0.39)** |

Table 5: Ablation study results (%). MRR and Hits@K with mean and std. deviations over 5 runs with different seeds.

## D Experimental Settings

This section outlines the experimental setup used to evaluate all models. We describe the computational resources and the hyperparameter search space. Moreover for SP4LP we include details regarding how the calculation of the shortest path is performed. Details are reported below.

**Computational Resources** 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).

**Hyperparameters** All models are tuned using a grid search over learning rate $\in [1\times10^{-2}, 1\times10^{-3}]$, dropout $\in [0, 0.7]$, weight decay $\in [0, 10^{-4}, 10^{-7}]$, number of GNN layers $\in \{1, 2, 3\}$, hidden dimensions $\in \{32, 64, 128, 256\}$ and prediction layers $\in \{1, 2, 3\}$. For large-scale datasets, we follow the reduced search space adopted in Li et al. [15] to avoid excessive compute. For SP4LP, we additionally explore the choice of GNN component $\in \{\text{GCN}, \text{GraphSAGE}, \text{GAT}\}$ and sequence model $\in \{\text{LSTM}, \text{Transformer}\}$, the best models are shown in Table 7. The best hyperparameters are selected based on validation performance. All reported metrics are averaged over 5 different seeds.

| **Dataset** | Cora | Citeseer | Pubmed | ogbl-collab | ogbl-ddi |
|---|---|---|---|---|---|
| #Nodes | 2,708 | 3,327 | 18,717 | 235,868 | 4,267 |
| #Edges | 5,278 | 4,676 | 44,327 | 1,285,465 | 1,334,889 |
| Mean Degree | 3.90 | 2.81 | 4.74 | 10.90 | 625.68 |
| Split Ratio | 85/5/10 | 85/5/10 | 85/5/10 | 92/4/4 | 80/10/10 |

Table 6: Dataset statistics. The split ratio indicates the percentages used for train/validation/test.

| **Dataset** | GNN model | Sequence model |
|---|---|---|
| Cora | GCN | Transformer |
| Citeseer | GCN | LSTM |
| Pubmed | SAGE | Transformer |
| ogbl-collab | SAGE | Transformer |
| ogbl-ddi | GCN | Transformer |

Table 7: Best GNN and sequence models selected via hyperparameter tuning.

**Shortest path calculation for SP4LP** For the computation of shortest paths between node pairs $u$ and $v$, we used the `shortest_path` function from the `networkx` library [7]. This function returns a single shortest path between two nodes to ensure computational efficiency, even when multiple shortest paths exist. If no path was found between $u$ and $v$ (i.e., they belonged to different connected components), we assigned a synthetic path of length one directly connecting $u$ and $v$.