ATLASULP: Domain-aware Universal Link Prediction via Relation Atlas

Yujing Liu[1], **Yixin Liu**[1], **Yu Zheng**[1], **Lianhua Chi**[2]
Alan Wee-Chung Liew[1], **Heng Tao Shen**[3], **Shirui Pan**[1*]

[1]School of Information and Communication Technology, Griffith University, Australia
[2]Department of Computer Science and Information Technology, La Trobe University, Australia
[3]School of Computer Science and Technology, Tongji University, China

yujing.liu@griffithuni.edu.au s.pan@griffith.edu.au

Abstract

Link prediction (LP) is a widely applied task in graph learning. Conventional LP approaches typically follow a dataset-specific paradigm, requiring independent training for each graph and incurring high computational and maintenance costs in large-scale applications. Motivated by these limitations, recent work explores Universal Link Prediction (ULP), aiming to enable training-free inference on arbitrary unseen graphs. However, existing methods primarily rely on subgraph sampling strategies to construct universal representations, which are unable to capture higher-order information due to the exponential growth of higher-order neighborhoods. As a result, they often under-perform on graphs where long-range signals are essential, such as biological networks. Moreover, their prediction models are typically domain-agnostic and lack the ability to adapt to different graph domains, leading to limited generalization. To address these challenges, we propose an Atlas-based Universal Link Prediction framework (ATLASULP), which introduces the relation atlas to capture structural associations between node pairs from both global and local perspectives, enabling a comprehensive characterization of connectivity signals across diverse graph domains. Building upon this representation, we develop a domain-guided prompting ULP model, which generates domain-aware prompt tokens from contextual structures and performs prompt-guided in-context prediction for adaptive link inference. Extensive experiments demonstrate that ATLASULP consistently outperforms state-of-the-art methods across diverse graph domains.

1 Introduction

Link prediction (LP) aims to infer missing or potential interactions between entities in graph data, and is one of the most fundamental tasks in graph learning. With the widespread adoption of graph-structured data in real-world scenarios such as social networks [4, 18], biological networks [16, 36], and information networks [38, 42], LP techniques have been extensively applied to address practical problems such as friend recommendation [51, 52], protein–protein interaction prediction [40, 45], and knowledge graph completion [12, 48]. Despite the remarkable success of Graph Neural Networks (GNNs) [24, 28, 58] and various heuristic algorithms [11, 14] in LP, most existing approaches follow a dataset-specific paradigm, requiring independent model training or heuristic selection for new graphs. This results in heavy computational and operational costs, hindering their large-scale deployment in industrial settings and their applicability in real-world applications with limited training data.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To overcome the above limitations, a promising direction is to develop universal models [30, 25, 31] that enable training-free LP on unseen graphs without dataset-specific adaptation. Distinct from other graph learning tasks such as node classification [29] and graph classification [19], LP is inherently more dependent on topological structures than node attributes, as it aims to infer missing links based on structural connectivity patterns. Moreover, in several real-world scenarios (e.g., protein-protein interaction networks), graph data are often unattributed, where only topological information is available. As a result, graph foundation models designed for other graph learning tasks often exhibit limited performance or even become inapplicable in LP due to their reliance on node attributes. In this context, Dong et al. conducted a pioneering exploration into universal link prediction (ULP), aiming to train models on source graphs from multiple domains and enable effective link prediction on unseen target graphs without fine-tuning. To this end, they proposed UniLP [10], a framework that extends Double Radius Node Labeling (DRNL) [54] to construct universal structural representations and incorporates an attention-based in-context learning model to predict links on unseen graphs.

Despite its pioneering efforts, UniLP still exhibits limited generalizable prediction capability across different domains, as it struggles to tackle two fundamental challenges in ULP. The first would be: ***C1 - How to construct a universal representation that can capture connectivity patterns across different graph domains?*** Due to domain heterogeneity, graphs from different domains exhibit distinct structural patterns and hence rely on different structural cues for link prediction [8, 26, 33]. For example, in social networks, links exhibit strong local clustering, well predicted by local patterns such as common neighbors [26]. In contrast, in citation networks, local patterns are often ineffective due to the scarcity of local topology. In these cases, global approaches such as the Katz index can better characterize higher-order path dependencies [8, 33]. Such significant diversity underscores the need for comprehensive representations that can capture multi-scale structural information for ULP. Nevertheless, UniLP only leverages DRNL on local ego-subgraphs as the universal representation, which emphasizes the importance of 1–2-hop neighborhoods but overlooks higher-order structural signals, leading to suboptimal performance on graphs where long-range structural dependencies are critical, such as citation (Cora) and biological (Celegans) networks (Fig. 1).

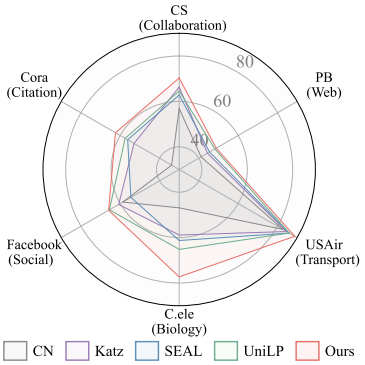


Figure 1: Generalization across diverse graph domains.

While the representations serve as a unified input that provides rich structural information, the universal model still plays the critical role of learning both shared knowledge and domain-specific variations across different graph domains. This further motivates the second challenge: ***C2 - How to design universal models that can adaptively identify domain-specific connectivity signals?*** Since different graph domains exhibit distinct connectivity patterns, a comprehensive universal representation inevitably contains both useful and redundant structural signals depending on the target domain. Therefore, an ideal ULP model should go beyond a static prediction paradigm, possess the ability to perceive the connectivity patterns of the target domain. Ideally, it should further leverage structural priors learned from pre-training to capture discriminative structural signals from universal representations, thereby enabling cross-domain transfer. However, UniLP only partially fulfills this ideal, as its adaptive mechanism is confined to leveraging few-shot context samples for prediction, while failing to model domain-specific structural patterns with dynamic adaptation.

In this work, we tackle the above two challenges by proposing ATLASULP, a novel ULP framework. Our theme is to learn a comprehensive representation that captures multi-scale structural information, and couple it with a domain-adaptive inference model for ULP. More specifically, to address ***C1***, we introduce Relation Atlas (RA), a new pair-level universal representation tailored for ULP, encoding multi-scale structural information from both global and local perspectives. At the global level, we leverage multi-order Chebyshev recursion to capture multi-order path responses, thereby modeling long-range topological dependencies. At the local level, we characterize fine-grained structural relationships using statistical measures that capture local connectivity patterns and structural role similarity, complementing global representations. Building upon RA, to handle ***C2***, we propose a domain-guided prompting ULP model, consisting of two modules: domain-aware prompt learning and prompt-guided in-context prediction. Conditioned on the RA representations of the target dataset,

the former constructs domain-specific structural prompts from a set of learnable structural probes, enabling the model to explicitly model domain-specific information. Powered by a Transformer backbone, the latter performs domain-adaptive link prediction by leveraging domain prompts and contextual information, thereby facilitating knowledge transfer across diverse graph domains. To sum up, the main contributions of this paper are three-fold:

- **Unified Representation.** We rethink the construction of universal representations in existing LP methods and identify their limitations, motivating a relation atlas (RA) that captures both global and local structural associations between node pairs.
- **Universal Model.** We propose ATLASULP, an Atlas-grounded universal LP framework that builds upon RA to enable effective cross-domain generalization via domain-aware prompt learning and prompt-guided in-context link prediction.
- **Experiments.** Extensive experimental results on graph datasets from six real-world domains validate the effectiveness, transferability, and scalability of ATLASULP.

2 Related Work

2.1 Link Prediction.

With the increasing prevalence of graph-structured data, graph learning techniques [56, 37, 32] have been widely applied to various real-world problems. Among them, link prediction (LP) is a fundamental problem in graph mining. Early approaches mainly rely on heuristic measures that quantify node similarity based on graph topology. Representative methods include local heuristics such as Common Neighbors (CN) [26], Adamic-Adar (AA) [2], and Resource Allocation (RA) [57], as well as global measures such as the Katz index [20], which captures higher-order path dependencies. While these methods are computationally efficient and interpretable, they rely on predefined rules and lack adaptability.

More recently, learning-based approaches have achieved significant success by automatically learning structural representations. Early embedding-based methods, such as DeepWalk [39] and node2vec [13], learn node representations via random walks, while matrix factorization methods [34] model latent interactions. Graph neural networks (GNNs) further advance LP by learning expressive node representations through message passing [21]. Variants such as SEAL [54] enhance performance by leveraging enclosing subgraphs and structural labeling (e.g., DRNL) to encode local topology, while recent methods (e.g., ELPH [7], NCNC [49], MPLP [9]) improve scalability and structural modeling. Despite their strong performance, most learning-based methods follow a dataset-specific training paradigm, requiring separate training for each graph, which limits scalability and generalization to unseen domains.

2.2 Universal Link Prediction.

To overcome the limitations of the dataset-specific paradigm, recent efforts have begun to explore foundation models for graph learning, aiming to generalize across diverse graph domains. Graph Foundation Models (GFMs), such as OFA [27] and PRODIGY [17], have shown promising results in zero-shot node and graph classification by leveraging node attributes and prompt-based task adaptation. However, their effectiveness in link prediction remains limited, as LP primarily depends on structural connectivity rather than semantic attributes, making attribute-centric approaches less suitable for unattributed graphs.

Consequently, Universal Link Prediction (ULP) has emerged to explicitly learn transferable topological patterns. UniLP [10] represents a pioneering effort in this direction. It extends Double Radius Node Labeling to construct universal structural representations and employs an attention-based in-context learning mechanism with target graph exemplars to enable training-free inference. Despite its promising results, UniLP still faces notable limitations. First, its reliance on local ego-subgraphs restricts the modeling of higher-order structural dependencies (akin to those captured by Katz or NBFNet). Second, its adaptation mechanism utilizes a domain-agnostic matching function. Since different domains exhibit highly heterogeneous connectivity priors (e.g., homophily in social networks versus structural equivalence in biological networks), such a design may limit its ability to adaptively capture domain-specific patterns.

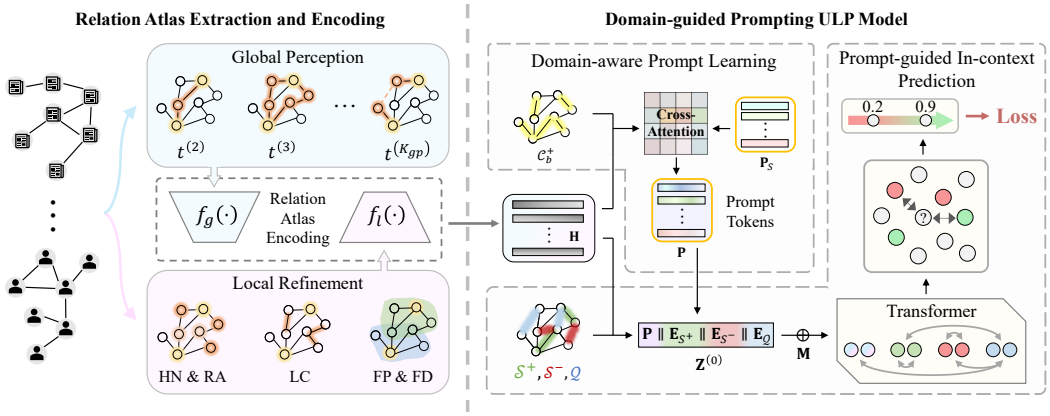


Figure 2: The framework of the proposed **ATLASULP**.

3 Preliminary

Notations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_1, \cdots, v_n\}$ is the set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of observed edges. The graph topology is encoded by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Given a edge $e = (v_i, v_j)$, the link prediction (LP) task aims to estimate the likelihood of an edge existing between them. We denote $y_e \in \{0, 1\}$ as the link label, where $y_e = a_{ij}$ indicates whether the edge exists.

Universal Link Prediction Problem. The goal of universal link prediction (ULP) is to learn a universal model $f_\theta(v_i, v_j)$ that can generalize across diverse graph domains. Formally, let $\mathcal{T}_s = \{\mathcal{G}_s^{(1)}, \cdots, \mathcal{G}_s^{(N)}\}$ denote a collection of source-domain graphs used for training, and $\mathcal{T}_t = \{\mathcal{G}_t^{(1)}, \cdots, \mathcal{G}_t^{(N')}\}$ denote unseen target-domain graphs. The objective is to learn a ULP parameterized by θ from source-domain graphs, which can be directly applied to unseen target graphs without further fine-tuning. During inference on target graphs, observed edges naturally serve as contextual references, providing in-context information without additional cost. The model leverages such structural context to guide link inference. The final objective is to correctly rank missing edges higher than non-edge node pairs in the target graph.

4 Method

In this section, we present ATLASULP, an relation **Atlas**-based **U**niversal **L**ink **P**rediction framework (Fig. 2). ATLASULP consists of two components: Relation Atlas extraction and encoding, which captures and encodes multi-scale topological associations between node pairs into a universal LP representation; and a domain-guided prompting ULP model, which performs domain-adaptive link inference via domain-aware prompt learning and prompt-guided in-context prediction.

4.1 Relation Atlas Extraction and Encoding

To develop a generalizable ULP framework, a key prerequisite is to map graph data from different domains into a unified and comparable representation space. As predicting links across graph domains may rely on different structural cues, a universal representation should integrate multi-scale topology to cover comprehensive connectivity patterns. To this end, we propose Relation Atlas (RA), a node-pair-level unified representation that explicitly models these structural priors through both *global perception* and *local refinement*.

4.1.1 Relation Atlas Extraction

Global Perception. As the main component of RA, the *global perception* characterizes global structural relationships by capturing multi-order path response intensities between node pairs. Such higher-order signals are essential in sparse graphs such as citation networks, where many linked paper pairs share no common neighbors due to sparse citation structures and are only connected

via multi-hop citation chains. While several traditional heuristics (e.g., Katz and PageRank) are designed for capturing high-order structural information, they aggregate multi-order topology via predefined schemes, which may blur contributions across scales and limit their ability to distinguish heterogeneous higher-order dependencies across graph domains. To address this, we propose multi-order path response modeling, which leverages Chebyshev Polynomial Recursion to explicitly encode order-wise response strengths. Given the normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, we compute the a-th order polynomial matrix $\mathbf{T}_a$ recursively:

$$\mathbf{T}_a = 2\widetilde{\mathbf{A}}\mathbf{T}_{a-1} - \mathbf{T}_{a-2}, \tag{1}$$

where $\mathbf{T}_0 = \mathbf{I}$ and $\mathbf{T}_1 = \widetilde{\mathbf{A}}$. The response strength for a node pair (v_i, v_j) at order a can be extracted from the corresponding matrix element $t_{i,j}^{(a)} = [\mathbf{T}_a]_{i,j}$. Intuitively, $t_{i,j}^{(a)}$ reflects the degree of path-based association between v_i and v_j at a hops, capturing scale-specific dependencies in the global topology. This strategy effectively suppresses backtrack signals through differential cancellation between recursive terms, mitigating the homogenization of high-order signals common in $\tilde{\mathbf{A}}^k$-based methods [23]. We then concatenate the responses across K_{gp} orders to obtain the global perception representation of node pair (v_i, v_j):

$$\mathbf{h}_{global}^{(i,j)} = \left[t_{i,j}^{(1)} \parallel t_{i,j}^{(2)} \parallel \cdots \parallel t_{i,j}^{(K_{gp})} \right], \tag{2}$$

where K_{gp} denotes the maximum path order considered in global perception. This yields a comprehensive multi-scale characterization of global structural relationships, providing clear and informative topological signals for downstream general models.

Local Refinement. While global perception captures long-range structural signals, it is less effective in modeling fine-grained local interactions, which is also crucial in many graphs. To complement this, we introduce *local refinement*, another key component of RA that captures K_{lr}-hop local structural patterns using lightweight statistical indicators, avoiding costly subgraph extraction, where K_{lr} denotes the maximum hop considered in local refinement. Specifically, we characterize local structural patterns from two complementary perspectives: *local connectivity patterns* and *structural role similarity*. Detailed mathematical formulations are deferred to Appendix A.1.

Local connectivity patterns serve as strong link indicators in many graphs [26, 33], where node pairs with more shared neighbors (e.g., mutual friends in social networks) are more likely to form links. In this paper, we characterize local connectivity patterns using three measures: Hierarchical Neighborhood Overlap ($\mathbf{s}_{\text{HN}}$), Resource-Allocated Overlap ($\mathbf{s}_{\text{RA}}$), and Local Connectivity Density ($\mathbf{s}_{\text{LC}}$). $\mathbf{s}_{\text{HN}}$ captures the quantity of neighbor overlap across multi-hop structural scales. $\mathbf{s}_{\text{RA}}$ introduces degree information to differentiate the importance of common neighbors, acknowledging that low-degree shared neighbors often indicate tighter connections than high-degree hubs. $\mathbf{s}_{\text{LC}}$ further quantifies the edge connection strength between cross-level neighborhoods to capture topological compactness that cannot be expressed by overlap alone. Through these three metrics, we obtain a detailed characterization of local subgraph structure. Notably, as all metrics are computed via sparse matrix operations without subgraph extraction, the design is computationally efficient.

Structural role similarity characterizes the node pairs by reflecting each node's position in local topology (e.g., hubs or peripheral nodes) and serves as an important signal for predicting connectivity behavior in real-world graphs [41, 15]. For example, prominent hubs (e.g., celebrities in social networks) inherently possess a higher propensity to interconnect, whereas peripheral nodes are usually confined to isolated cliques. We characterize structural role similarity using Frontier Size Product ($\mathbf{s}_{\text{FP}}$) and Frontier Size Difference ($\mathbf{s}_{\text{FD}}$). $\mathbf{s}_{\text{FP}}$ measures the product of frontier set sizes across different orders, while $\mathbf{s}_{\text{FD}}$ computes their absolute discrepancy. Jointly modeling these metrics provides a comprehensive view of structural roles. For instance, a small $\mathbf{s}_{\text{FD}}$ coupled with a high $\mathbf{s}_{\text{FP}}$ suggests symmetric hub-like nodes, which typically exhibit higher link formation likelihood than symmetric peripheral ones.

By combining all five structural metrics, we obtain a unified representation of local refinement representation, which is defined as follows:

$$\mathbf{h}_{local}^{(i,j)} = \left[\mathbf{s}_{\text{HN}}^{(i,j)} \parallel \mathbf{s}_{\text{RA}}^{(i,j)} \parallel \mathbf{s}_{\text{LC}}^{(i,j)} \parallel \mathbf{s}_{\text{FP}}^{(i,j)} \parallel \mathbf{s}_{\text{FD}}^{(i,j)} \right]. \tag{3}$$

Finally, by combining the global perception and local refinement representations, we obtain a comprehensive modeling of multi-scale topological signals, forming an effective universal LP representation.

4.1.2 Relation Atlas Encoding

After constructing the global perception representation $\mathbf{h}_{global}^{(i,j)}$ and the local refinement representation $\mathbf{h}_{local}^{(i,j)}$, we project them into a continuous embedding space. Since global and local signals capture different topological scales and exhibit distinct feature distributions, directly fusing them may cause one signal to dominate the other or dilute its contribution. Therefore, we employ a dual-stream encoder that enables separate non-linear transformations for each component. Specifically, the global perception and local refinement representations are separately encoded and then concatenated to obtain the final embedding for any given edge $e = (v_i, v_j)$:

$$\mathbf{h}_e = \left[f_g(\mathbf{h}_{global}^{(i,j)}) \parallel f_l(\mathbf{h}_{local}^{(i,j)}) \right], \tag{4}$$

where $f_g(\cdot)$ and $f_l(\cdot)$ denote the global perception encoder and local refinement encoder, respectively. This comprehensive representation $\mathbf{h}_e$ captures multi-scale topological signals and serves as the foundation for the subsequent training-free inference.

4.2 Domain-guided Prompting ULP Model

While the aforementioned universal representation comprehensively captures pairwise structural correlations, real-world graph domains often exhibit heterogeneous connectivity patterns. Consequently, structural signals that are highly predictive in one domain may act as redundant noise in another. To perform effective training-free inference, an ideal universal model must adaptively distill valid link-predictive signals from the universal representation while suppressing domain-irrelevant interference. To this end, we design a domain-guided prompting ULP model, consisting of Domain-aware Prompt Learning and prompt-guided in-context prediction.

4.2.1 Domain-aware Prompt Learning

To explicitly model the connection patterns of target domain, we propose a perceiver-based domain prompt generator that extracts K domain-aware prompt tokens from context samples. These tokens, carrying domain-level information, can provide domain-related guidance for inference.

The domain-aware prompt learning module takes edge representations as input to construct prompt tokens. However, given the diverse scale of potential target graphs, using all edges to construct prompt tokens may incur prohibitive computational and memory costs for some large-scale data. In the other hand, random sampling may fail to capture rare structural patterns (e.g., sparse connections involving low-degree nodes). To address this, we adopt a lightweight context selection strategy to select representative samples. Specifically, we rank candidate edges by the sum of endpoint degrees and apply uniform interval sampling to select M background reference edges, i.e., $\mathcal{C}_b^+ = e_1, \cdots, e_M$. Then, their representations $\mathbf{H}_b = [\mathbf{h}_{e_1}, \cdots, \mathbf{h}_{e_M}]^\top \in \mathbb{R}^{M \times d}$ can serve as the input of the prompt learning module. This strategy ensures balanced coverage from sparse regions to hub-dominated structures with low overhead, preserving the diversity of structural patterns in the context.

Based on the selected background reference edges, we design a perceiver-based domain-aware prompt generator that uses K learnable structural probes $\mathbf{P} \in \mathbb{R}^{K \times d}$ to distill the topological patterns of target dataset via a cross-attention block where $\mathbf{H}_b$ serves as keys and values:

$$\mathbf{P}^{'} = \text{Softmax}\left(\frac{\mathbf{P}\mathbf{H}_b^\top}{\sqrt{d}}\right)\mathbf{H}_b. \tag{5}$$

Subsequently, we feed the intermediate embedding $\mathbf{P}^{'}$ into a Feed-Forward Network (FFN) for non-linear mapping and refinement to obtain K domain-aware prompt tokens:

$$\mathbf{P}_{domain} \in \mathbb{R}^{K \times d} = \text{FFN}(\text{LayerNorm}(\mathbf{P} + \mathbf{P}^{'})) + \mathbf{P}^{'}. \tag{6}$$

Through this process, we compress domain-related background information into a fixed number of domain-level prompt tokens, mapping instance-level structural observations to domain-level

summaries. These prompts are independent of specific edges, providing compact domain-aware signals for subsequent inference.

4.2.2 Prompt-guided In-context Prediction

After obtaining the domain-aware prompt tokens $\mathbf{P}_{domain}$, we perform prompt-guided in-context prediction to adaptively predict query edges under prompt conditioning. We use a Transformer as the in-context encoder, as its expressive power and scalability make it well suited for serving as a foundation model. We also construct a refined support set from the filtered background context to serve as a concrete reference of connected/disconnected examples. To ensure low inference overhead, we design a deterministic node pair sampling strategy to construct the context sample set. Given background reference edges $\mathcal{C}_b^+$ and negative pool $\mathcal{C}^-$ (globally sampled negatives), we assign each candidate edge $e = (v_i, v_j)$ a structural typicality score. The score is computed by weighting the 1-hop components of hierarchical neighborhood overlap and local connectivity density:

$$o(e) = \lambda_1 \cdot (\mathbf{f}_{v_i}^{(1)})^\top \mathbf{f}_{v_j}^{(1)} + \lambda_2 \cdot (\mathbf{f}_{v_i}^{(1)})^\top \mathbf{A} \mathbf{f}_{v_j}^{(1)} + \epsilon \cdot \text{idx}(e), \tag{7}$$

where $\mathbf{f}_{v_i}^{(a)} \in \{0,1\}^n$ is the binary indicator vector of the a-hop frontier set of node v_i, λ_1 and λ_2 are weighting coefficients, and ϵ is a small tie-breaking regularizer based on the edge index. Subsequently, we sort $\mathcal{C}_b^+$ and $\mathcal{C}^-$ by $o(e)$ and apply uniform interval sampling to obtain support sets $\mathcal{S}^+$ and $\mathcal{S}^-$. This ensures that $\mathcal{S}^-$ covers hard negatives with high structural overlap but no observed links, improving boundary separation in in-context inference.

Next, we feed domain prompts, context references, and query edges into the Transformer for in-context representation learning. Given the refined support set $\mathcal{S} = \mathcal{S}^+, \mathcal{S}^-$ and query set $\mathcal{Q}$, we first obtain their edge embeddings via the dual-stream encoder (Eq. (4)). We then construct the input sequence of Transformer consisting of domain prompts, context references, and query samples, distinguished by role embeddings:

$$\mathbf{Z}^{(0)} = [\mathbf{P}_{domain} \parallel \mathbf{E}_{\mathcal{S}^+} \parallel \mathbf{E}_{\mathcal{S}^-} \parallel \mathbf{E}_{\mathcal{Q}}], \tag{8}$$

where learnable role embeddings are incorporated into each component, e.g., $\hat{\mathbf{P}}_{domain} = \mathbf{P}_{domain} + \mathbf{e}_{role}^{(p)}$, and $\mathbf{E}_{\mathcal{S}^+}$, $\mathbf{E}_{\mathcal{S}^-}$, $\mathbf{E}_{\mathcal{Q}}$ are obtained by adding corresponding role embeddings $\mathbf{e}_{role}^{(s+)}, \mathbf{e}_{role}^{(s-)}, \mathbf{e}_{role}^{(q)}$ to the dual-stream edge embeddings $\mathbf{H}$. To enforce the inductive setting in in-context inference, we design an attention mask $\mathbf{M} \in \mathbb{R}^{L_{seq} \times L_{seq}}$ to prevent information leakage among query samples, and feed the input sequence $\mathbf{Z}^{(0)}$ together with $\mathbf{M}$ into an L-layer Transformer encoder (see Appendix A.2 for details), yielding the final representations $\mathbf{Z}^{(L)} = \left[\tilde{\mathbf{P}}_{domain} \parallel \tilde{\mathbf{E}}_{\mathcal{S}^+} \parallel \tilde{\mathbf{E}}_{\mathcal{S}^-} \parallel \tilde{\mathbf{E}}_{\mathcal{Q}}\right]$. The domain prompt tokens are then discarded, as they serve as contextual priors, and extract updated representations for positive supports $\tilde{\mathbf{E}}_{\mathcal{S}^+}$, negative supports $\tilde{\mathbf{E}}_{\mathcal{S}^-}$, and queries $\tilde{\mathbf{E}}_{\mathcal{Q}}$ from the final representations for link inference.

Subsequently, for each query edge $e_i \in \mathcal{Q}$, we evaluate its topological affinity by calculating its average similarity with the positive and negative support sets:

$$s^+(e_i) = \frac{1}{|\mathcal{S}^+|} \sum_{e^+ \in \mathcal{S}^+} \text{sim}(\mathbf{z}_{e_i}, \mathbf{z}_{e^+}) \quad \text{and} \quad s^-(e_i) = \frac{1}{|\mathcal{S}^-|} \sum_{e^- \in \mathcal{S}^-} \text{sim}(\mathbf{z}_{e_i}, \mathbf{z}_{e^-}), \tag{9}$$

where $\mathbf{z}_{e_i}$, $\mathbf{z}_{e^+}$, and $\mathbf{z}_{e^-}$ are feature vectors extracted from the updated representations, and $\text{sim}(\cdot,\cdot)$ denotes cosine similarity in a linearly projected metric space. Based on the relative similarity, the final link probability of query edge q is computed as:

$$\tilde{y}_{e_i} = \sigma\big(\tau \cdot (s^+(e_i) - s^-(e_i)) + b\big), \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function, τ is a learnable temperature scaling the logits, and b is a bias term. In summary, our model enables adaptation to unseen graph domains without fine-tuning through three mechanisms. First, domain-aware prompt tokens encode domain-level structural patterns, guiding the model to focus on link-indicative signals within RA. Second, positive and negative support sets

provide concrete references for calibrating query edges against observed structures. Finally, the similarity-based scoring mechanism performs relative comparison instead of absolute classification, improving robustness to distribution shifts. Together, these components enable efficient and robust link inference in new graph domains.

4.2.3 Model Training

On the training datasets, we optimize the model by minimizing the binary cross-entropy (BCE) loss between the predicted link probabilities $\tilde{y}_q$ and the ground-truth labels $y_q \in \{0, 1\}$ of query edges:

$$\mathcal{L} = -\frac{1}{|\mathcal{T}|} \sum_{e_i \in \mathcal{T}} \Big[y_{e_i} \log \tilde{y}_{e_i} + (1 - y_{e_i}) \log(1 - \tilde{y}_{e_i}) \Big], \tag{11}$$

where $\mathcal{T}$ denotes the set. The framework is trained in an end-to-end episodic manner, enabling the model to capture transferable topological connection patterns that generalize robustly to unseen target domains for training-free inference. Detailed algorithmic descriptions are provided in Appendix B.

4.3 Complexity Analysis

We analyze the computational complexity of ATLASULP, which consists of Relation Atlas (RA) construction and Prompt-guided In-context Prediction.

In RA construction, global perception is computed via K_{gp}-order sparse matrix propagation, while local refinement is obtained through K_{lr}-hop structural statistics. Both rely on sparse operations over the edge set, yielding a unified complexity of $\mathcal{O}(K_{\max}|\mathcal{E}|)$, where $K_{\max} = \max(K_{gp}, K_{lr})$ and $|\mathcal{E}|$ denotes the number of edges. In the Prediction stage, a Transformer operates over K domain prompts, $|\mathcal{S}|$ support samples, and $|\mathcal{Q}|$ query edges with hidden dimension d. Each query attends only to a fixed-size context composed of prompts and support samples, resulting in complexity $\mathcal{O}(|\mathcal{Q}|(K + |\mathcal{S}|)d + (K + |\mathcal{S}|)^2 d)$. Since K and $|\mathcal{S}|$ are small constants, the inference cost reduces to $\mathcal{O}(|\mathcal{Q}|d)$. Overall, the total complexity of ATLASULP is: $\mathcal{O}(K_{\max}|\mathcal{E}| + |\mathcal{Q}|d)$.

5 Experiments

5.1 Experimental Setup

Datasets. Following prior work on universal link prediction, we collect graph datasets from diverse domains, including biology (Ecoli, Yeast, Celegans) [47, 50, 55], transport (Power, USAir) [6, 50], web (PolBlogs, Router, PB) [1, 3, 46], collaboration (Physics, CS, NS) [35, 44], citation (Pubmed, Citeseer, Cora) [53], and social networks (Twitch, Github, Facebook) [43]. The datasets are partitioned into two disjoint groups: a pre-training (source) set and a testing (target) set. For each target graph, edges are split into 70%, 10%, and 20% for context (observed edges), validation, and testing (unobserved edges), respectively. Detailed statistics are provided in Appendix C.1.

Comparison Methods. We compare ATLASULP with a comprehensive set of representative baselines, including: (1) classical heuristic methods, such as Common Neighbors (CN) [26], Adamic-Adar (AA) [2], Resource Allocation (RA) [57], Preferential Attachment (PA) [5], Shortest Path (SP), and Katz index [20].; (2) supervised and pretraining-based link prediction models, including SEAL [54], GAE [22], ELPH [7], NCNC [49], and MPLP [9]; and (3) the state-of-the-art universal link prediction model UniLP [10]. Detailed descriptions of these methods are provided in Appendix C.2.

Evaluation and Implementation. Following prior work, we adopt Hits@50 as the evaluation metric for link prediction, reporting the average performance over multiple runs with different random seeds to ensure reliable comparisons. To systematically assess cross-domain transferability, parametric GNN baselines are evaluated under two settings: (1) *Pretrain Only*, where pretrained models are directly applied to target graphs without any parameter updates; and (2) *Pretrain & Finetune*, where pretrained models are further adapted to target graphs using 200 sampled positive and negative edges from the context split. In contrast, ATLASULP and UniLP perform inference on unseen target graphs in a strictly training-free manner, without requiring any additional parameter updates or fine-tuning. Further implementation details and experimental configurations are provided in Appendix C.3.

Table 1: Link prediction results on test datasets evaluated by Hits@50. The format is average score ± standard deviation. The top three models are colored by First, Second, Third.

Method	C.ele	USAir	PB	NS	CS	Cora	Facebook	Ave.
Heuristics								
CN	46.88±12.28	82.75±1.54	41.15±3.77	74.03±1.59	56.84±15.56	33.85±0.93	58.70±0.35	56.31
AA	61.07±5.16	86.96±2.24	44.12±3.36	74.03±1.59	68.22±1.08	33.85±0.93	**67.80±2.12**	62.29
RA	62.80±4.84	**87.27±1.89**	43.72±2.86	74.03±1.59	68.21±1.08	33.85±0.93	**68.84±2.03**	62.67
PA	43.85±4.12	77.69±2.29	28.93±1.91	35.35±3.01	6.49±0.61	22.09±1.52	12.95±0.63	32.48
SP	0.00±0.00	0.00±0.00	0.00±0.00	80.00±1.11	41.34±35.58	52.97±1.53	0.00±0.00	24.90
Katz	58.86±6.48	84.64±1.86	44.36±3.65	78.96±1.35	66.32±5.59	52.97±1.53	60.79±0.60	63.84
Pretrain Only								
SEAL	61.28±3.76	86.00±1.56	45.44±2.68	84.07±1.96	62.82±1.62	56.21±2.24	54.57±1.48	64.34
GAE	44.71±3.39	76.12±2.27	27.56±2.34	15.20±2.14	5.08±0.48	24.22±1.53	6.65±0.57	28.51
ELPH	59.23±4.50	84.42±2.22	43.69±2.90	84.27±1.43	**70.69±3.63**	56.91±1.43	61.80±2.46	65.86
NCNC	48.07±4.79	75.44±4.22	25.66±1.42	80.07±1.43	34.27±2.28	52.51±2.54	19.28±1.58	47.90
MPLP	56.74±5.31	82.94±2.30	47.78±2.55	80.33±1.54	24.26±1.28	46.71±2.25	48.06±2.09	55.26
Pretrain & Finetune								
SEAL	**64.45±4.14**	**88.49±2.16**	47.78±3.32	84.84±2.32	61.54±3.09	**62.19±3.27**	58.70±2.78	**66.86**
GAE	44.71±4.07	74.47±2.96	25.92±2.64	18.34±2.27	4.95±0.44	25.31±1.48	6.11±0.39	28.54
ELPH	60.51±5.72	84.52±2.08	43.58±3.48	**86.08±0.69**	**71.10±3.48**	57.18±1.89	63.31±3.63	66.61
NCNC	**64.45±5.10**	85.85±2.46	47.75±6.90	**88.25±2.25**	58.75±5.74	**60.00±2.50**	59.32±6.93	66.34
MPLP	62.56±4.79	85.08±1.54	**48.01±2.94**	80.16±1.18	50.35±1.01	56.02±2.09	56.72±1.20	62.70
Universal LP model								
UniLP	**65.20±4.40**	85.98±2.00	**48.14±2.99**	**89.09±2.05**	64.59±2.65	57.50±2.40	65.49±2.05	**68.00**
Ours	**77.30±1.20**	**88.80±0.77**	**48.84±0.33**	**88.25±1.23**	**70.32±0.51**	**62.41±0.64**	**65.91±1.51**	**71.69**

5.2 Experimental Results

Performance Comparison. We evaluate AtlasULP on 7 unseen target graphs, results are reported in Table 1. ❶ AtlasULP achieves SOTA performance, attaining the best average Hits@50 (71.69%) and consistently outperforming all baselines, benefiting from the integration of Relation Atlas and domain-aware in-context inference for effective training-free transfer. ❷ It surpasses both heuristics and GNNs, with gains of 7.85% over Katz and 4.83% over SEAL, This benefits from domain-aware prompt learning and in-context prediction, which enable adaptive modeling of target-specific connectivity patterns. ❸ It also outperforms UniLP by 3.69% on average, as Relation Atlas captures richer multi-scale structural signals (global and local), yielding stronger representations for link inference.

Ablation Study of Relation Atlas. To assess the contribution of each component in the proposed Relation Atlas (RA), we conduct experiments on four datasets (Celegans, USAir, NS, and Facebook), considering four variants: removing local connectivity patterns ($-L_1$), removing structural role similarity ($-L_2$), removing the entire Local Refinement ($-L_{1,2}$), and removing Global Perception ($-G$). All variants consistently degrade performance, indicating that each component contributes to representation quality (Fig. 3). In particular, removing Local Refinement ($-L_{1,2}$) leads to an average 10.5% drop, highlighting the importance of local structural signals. These results confirm that RA effectively integrates complementary global and local information for structural modeling.

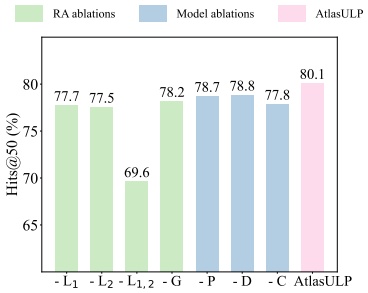


Figure 3: Ablation Studies

Ablation Study of ULP Model. To evaluate the contribution of each component in the proposed Domain-guided Prompting ULP model, we consider three variants: removing domain-aware prompt learning ($-$P), removing deterministic structural sampling ($-$D), and removing in-context scoring ($-$C). Removing any component consistently degrades performance, indicating that each module plays a critical role (Fig. 3). Moreover, these components work in a complementary manner to enable effective cross-domain generalization.

Efficiency Analysis To evaluate computational efficiency, we compare the running time of our method with UniLP on test graphs. As shown in Fig. 4, Our method achieves significant speedups across graph types. For example, on Facebook, UniLP requires 1862.31s while ours takes 29.47s (63× speedup); on PB, the speedup reaches 268× (345.2s vs. 1.29s). These gains stem from our efficient unified universal representation, which eliminates the need for expensive and repeated ego-subgraph extraction required by UniLP.

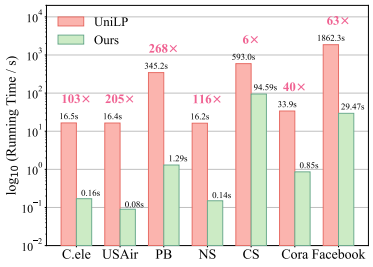


Figure 4: Time comparison

Parameter Sensitivity Analysis We investigate the sensitivity of ATLASULP to three key hyper-parameters: global perception hops K_{gp}, local refinement hops K_{lr}, and the support set size $|\mathcal{S}|$. The results are illustrated in Fig. 5. First, the performance of our method gradually improves as K_{gp} increases, as a larger K_{gp} provides richer and more comprehensive global information. Second, our method achieves optimal performance when K_{lr} is set to 3. This is because an excessively small K_{lr} is insufficient to fully characterize local neighborhoods, whereas an overly large K_{lr} introduces noisy signals due to the exponential growth of neighborhoods. Finally, the performance of our method stabilizes once $|\mathcal{S}|$ exceeds 20, demonstrating that our proposed deterministic sampling effectively captures representative contextual references.

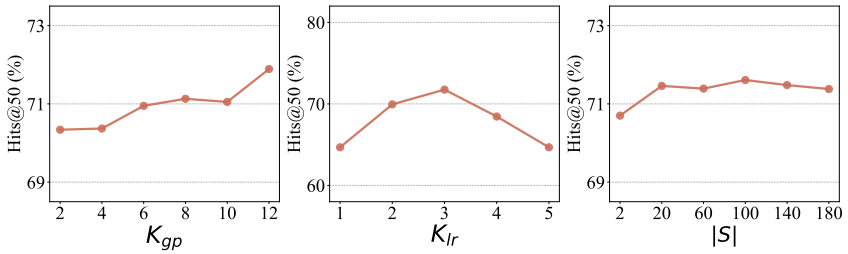


Figure 5: Parameter Sensitivity

Model Generalization Ability Analysis To assess the scalability of ATLASULP as a universal model, we study the relationship between its predictive performance (Hits@50) and two key factors: training data volume and model size (Fig. 6). Performance improves monotonically with both. At larger model scales (∼30k parameters), more data (∼100k samples) is required to fully utilize the model capacity. The best results occur in the large-data, large-model regime (top red region), where Hits@50 exceeds 70%. This trend aligns with standard scaling laws, demonstrating the effectiveness of ATLASULP in large-scale, cross-domain settings.

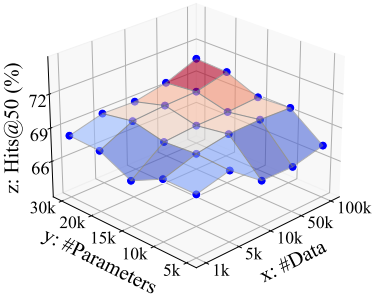


Figure 6: Scaling Law

6 Conclusion

In this paper, we propose a novel universal link prediction method, ATLASULP, aiming to construct a universal LP representation that comprehensively captures link-indicative information across diverse graph domains. We introduce the Relation Atlas (RA), which characterizes structural correlations between node pairs from both local and global perspectives. Built upon this, we further develop an Atlas-grounded ULP model that identifies domain-specific connectivity patterns via domain-aware prompting and enables effective adaptive transfer through prompt-based in-context prediction. Extensive experiments on graph datasets from six real-world domains demonstrate that ATLASULP achieves strong predictive performance, positive transferability, and scalable capability. The main **limitations** of ATLASULP lies in the need for manual design in constructing the Relation Atlas. Therefore, a promising **future directions** is to explore automated relational atlas learning.

Acknowledgments and Disclosures

Funding. The work of S. Pan was partially supported by the Australian Research Council (ARC) under Grant Nos. DP240101547 and FT210100097. The work of Y. Liu was partially supported by the ARC under Grant No. DE260101172.

Competing Interests. The authors declare no competing interests.

References

[1] R. Ackland. Mapping the us political blogosphere: Are conservative bloggers more prominent? BlogTalk Downunder 2005 Conference, Sydney, 2005.

[2] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[3] L. A. Adamic and N. Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

[4] G. A. Amran, X. Li, and A. A. Al-Bakhrani. Link prediction in social networks and e-commerce: A comprehensive review and bibliometric analysis. *Expert Systems with Applications*, 299: 129914, 2026.

[5] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439): 509–512, 1999.

[6] V. Batagelj and A. Mrvar. Pajek. *Encyclopedia of social network analysis and mining*, 1: 1245–1256, 2014.

[7] B. P. Chamberlain, S. Shirobokov, E. Rossi, F. Frasca, T. Markovich, N. Hammerla, M. M. Bronstein, and M. Hansmire. Graph neural networks for link prediction with subgraph sketching. *arXiv preprint arXiv:2209.15486*, 2022.

[8] A. Clauset and C. Moore. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[9] K. Dong, Z. Guo, and N. V. Chawla. Pure message passing can estimate common neighbor for link prediction. *Advances in Neural Information Processing Systems*, 37:73000–73035, 2024.

[10] K. Dong, H. Mao, Z. Guo, and N. V. Chawla. Universal link predictor by in-context learning on graphs. *arXiv preprint arXiv:2402.07738*, 2024.

[11] S. Fang, L. Li, S. Bai, Z. Ma, and X. Chen. Link prediction based on fundamental heuristic elements. *International Journal of Modern Physics C*, 35(12):2450161, 2024.

[12] M. Galkin, E. Denis, J. Wu, and W. L. Hamilton. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. *arXiv preprint arXiv:2106.12144*, 2021.

[13] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[14] C. Gui. Link prediction based on spectral analysis. *PloS one*, 19(1):e0287385, 2024.

[15] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1231–1239, 2012.

[16] I. Horecka and H. Röst. Risk: a next-generation tool for biological network annotation and visualization. *Bioinformatics*, 42(1):btaf669, 2026.

[17] Q. Huang, H. Ren, P. Chen, G. Kržmanc, D. Zeng, P. S. Liang, and J. Leskovec. Prodigy: Enabling in-context learning over graphs. *Advances in Neural Information Processing Systems*, 36:16302–16317, 2023.

[18] J. Jaskari, C. Roy, F. Ogushi, M. Saukkoriipi, J. Sahlsten, and K. Kaski. Temporal social network modeling of mobile connectivity data with graph neural networks. *PLoS One*, 20(12): e0335267, 2025.

[19] W. Ju, Z. Mao, S. Yi, Y. Qin, Y. Gu, Z. Xiao, J. Shen, Z. Qiao, and M. Zhang. Cluster-guided contrastive class-imbalanced graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11924–11932, 2025.

[20] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[21] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[22] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[23] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

[24] J. Li, H. Shomer, H. Mao, S. Zeng, Y. Ma, N. Shah, J. Tang, and D. Yin. Evaluating graph neural networks for link prediction: Current pitfalls and new benchmarking. *Advances in Neural Information Processing Systems*, 36:3853–3866, 2023.

[25] S. Li, Y. Zhao, Y. Tan, Q. Chen, Y. Liu, and S. Pan. Towards anomaly detection on relational data. *arXiv preprint arXiv:2606.18621*, 2026.

[26] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, 2003.

[27] H. Liu, J. Feng, L. Kong, N. Liang, D. Tao, Y. Chen, and M. Zhang. One for all: Towards training one graph model for all classification tasks. In *12th International Conference on Learning Representations, ICLR 2024*, 2024.

[28] W. Liu, K. Yue, Z. Zhang, and L. Duan. Hierarchical structure enhanced graph representation for link prediction. In *2025 IEEE International Conference on Data Mining (ICDM)*, pages 507–516. IEEE, 2025.

[29] Y. Liu, Z. Wu, Z. Lu, C. Nie, G. Wen, Y. Zhu, and X. Zhu. Noisy node classification by bi-level optimization based multi-teacher distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19033–19040, 2025.

[30] Y. Liu, Y. Liu, Y. Tan, X. Cao, A. W.-C. Liew, H. T. Shen, and S. Pan. Is real-world training data necessary for generalist graph anomaly detection? *arXiv preprint arXiv:2610.12167*, 2026.

[31] Y. Liu, Y. Liu, Y. Zheng, A. W.-C. Liew, X. Cao, and S. Pan. Rethinking feature alignment in generalist graph anomaly detection: A relational fingerprint-based approach. In *International Conference on Machine Learning*, 2026.

[32] Y. Liu, Y. Liu, Y. Zheng, Y. Tan, A. W.-C. Liew, and S. Pan. Pasta: Noisy node classification with partial label learning. *arXiv preprint arXiv:2608.25365*, 2026.

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

[34] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2011.

[35] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 74(3):036104, 2006.

[36] R. Nicolle, B. Caron, V. Malan, and A. Rausell. Pagan predicts digenic interactions by generalizing single-gene representations in biological networks. *medRxiv*, pages 2026–01, 2026.

[37] J. Pan, Y. Liu, Y. Zheng, L. Chi, A. W.-C. Liew, and S. Pan. Camera: Adapting to semantic camouflage in unsupervised text-attributed graph fraud detection. In *International Joint Conference on Artificial Intelligence*, 2026.

[38] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36 (7):3580–3599, 2024.

[39] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[40] M. Réau, N. Renaud, L. C. Xue, and A. M. Bonvin. Deeprank-gnn: a graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics*, 39(1):btac759, 2023.

[41] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 385–394, 2017.

[42] J. Robinson, R. Ranjan, W. Hu, K. Huang, J. Han, A. Dobles, M. Fey, J. E. Lenssen, Y. Yuan, Z. Zhang, et al. Relational deep learning: Graph representation learning on relational databases. In *NeurIPS 2024 third table representation learning workshop*, 2024.

[43] B. Rozemberczki, C. Allen, and R. Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[44] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[45] S. Sledzieski, R. Singh, L. Cowen, and B. Berger. D-script translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982, 2021.

[46] N. Spring, R. Mahajan, and D. Wetherall. Measuring isp topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32(4):133–145, 2002.

[47] C. Von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, 417(6887):399–403, 2002.

[48] L. Wang, W. Zhao, Z. Wei, and J. Liu. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 4281–4294, 2022.

[49] X. Wang, H. Yang, and M. Zhang. Neural common neighbor with completion for link prediction. *arXiv preprint arXiv:2302.00890*, 2023.

[50] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

[51] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

[52] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4753–4766, 2020.

[53] M. Zhang and Y. Chen. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 575–583, 2017.

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

[55] M. Zhang, Z. Cui, S. Jiang, and Y. Chen. Beyond link prediction: Predicting hyperlinks in adjacency space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[56] Y. Zhao, Y. Liu, Q. Chen, S. Li, Y. Tan, and S. Pan. Fedcigar: A personalized reconstruction approach for federated graph-level anomaly detection. In *International Joint Conference on Artificial Intelligence*, 2026.

[57] T. Zhou, L. Lü, and Y.-C. Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.

[58] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in neural information processing systems*, 34:29476–29490, 2021.

A Detail of ATLASULP

A.1 Local Refinement Construction

Although global perception captures global signals, it struggles with local structural interactions. Thus We introduce Local Refinement, which replaces subgraph sampling with lightweight statistical indicators to model local structure efficiently. Specifically, we model local connectivity patterns and structural role similarity using five metrics, starting with three measures of local connectivity: Hierarchical Neighborhood Overlap, Resource-Allocated Overlap, and Local Connectivity Density:

❶ **Hierarchical Neighborhood Overlap (HN)** characterizes neighbor overlap across multi-hop neighborhoods. We construct the multi-hop frontier sets for nodes v_i and v_j and calculate the intersection size between different hop levels:

$$\mathbf{s}_{\text{HN}}^{(i,j)} = \text{Concat}\left(\left\{|\mathcal{F}_i^{(a)} \cap \mathcal{F}_j^{(b)}| \;\middle|\; a, b \in \{1, \ldots, K_{lr}\}\right\}\right), \tag{12}$$

where $\mathcal{F}_i^{(a)}$ denotes the a-hop frontier set of node v_i, directly obtained from the non-zero entries of $\mathbf{T}_a$ in Eq. 1. K_{lr} denotes the maximum path order considered in local refinement. This metric captures neighborhood overlap across structural scales.

❷ **Resource-Allocated Overlap(RA)** aims to differentiate the contributions of various common neighbors. Low-degree neighbors often indicate a tighter connection than high-degree hubs. Therefore, we introduce degree information to weight the overlapping region:

$$\mathbf{s}_{\text{RA}}^{(i,j)} = \text{Concat}\left(\left\{\sum_{v_k \in \mathcal{F}_i^{(a)} \cap \mathcal{F}_j^{(b)}} \frac{1}{d(v_k)} \;\middle|\; a, b \in \{1, \ldots, K_{lr}\}\right\}\right), \tag{13}$$

where $d(v_k)$ represents the degree of node v_k. While HN models the quantity of common neighbors, RA distinguishes their importance. Combining both provides a comprehensive view of neighborhood overlap.

❸ **Local Connectivity Density (LC)** further quantifies the connection density between cross-level neighborhoods, capturing topological compactness that cannot be expressed by overlap alone. We calculate the edge connection strength between different frontiers:

$$\mathbf{s}_{\text{LC}}^{(i,j)} = \text{Concat}\left(\left\{(\mathbf{f}_i^{(a)})^\top \mathbf{A} \mathbf{f}_j^{(b)} \;\middle|\; a, b \in \{1, \ldots, K_{lr}\}\right\}\right), \tag{14}$$

where $\mathbf{f}_i^{(a)} \in \{0, 1\}^n$ is the binary indicator vector of the frontier set $\mathcal{F}_i^{(a)}$. Through these three metrics, we obtain a detailed characterization of local subgraph structure. As all metrics are computed via sparse matrix operations without subgraph extraction, the design is computationally efficient.

Next, we model structural role similarity between node pairs. Structural role reflects a node's position in local topology (e.g., hubs or peripheral nodes). Nodes with similar roles tend to exhibit stronger connectivity, making this signal useful for link prediction. We characterize structural role similarity using two metrics: Frontier Size Product and Frontier Size Difference:

❹ **Frontier Size Product (FP)** measures the product of frontier set sizes across different orders of node pairs:

$$\mathbf{s}_{\text{FP}}^{(i,j)} = \text{Concat}\left(\left\{|\mathcal{F}_i^{(a)}| \cdot |\mathcal{F}_j^{(a)}| \;\middle|\; a \in \{1, \ldots, K_{lr}\}\right\}\right). \tag{15}$$

❺ **Frontier Size Difference (FD)** measures the absolute discrepancy in neighborhood volumes of the node pair:

$$\mathbf{s}_{\text{FD}}^{(i,j)} = \text{Concat}\left(\left\{\left||\mathcal{F}_i^{(a)}| - |\mathcal{F}_j^{(a)}|\right| \;\middle|\; a \in \{1, \ldots, K_{lr}\}\right\}\right). \tag{16}$$

By jointly modeling these two metrics, we obtain a comprehensive characterization of structural role similarity. For example, a small FD indicates similar roles; when coupled with a high FP, it suggests

symmetric hub-like nodes, which tend to exhibit higher link formation likelihood than symmetric peripheral ones. By combining all five structural metrics, we obtain a unified representation of local connectivity patterns and structural role similarity for node pairs. The local refinement representation is defined as follows:

$$\mathbf{h}_{local}^{(i,j)} = \left[\mathbf{s}_{\text{HN}}^{(i,j)} \parallel \mathbf{s}_{\text{RA}}^{(i,j)} \parallel \mathbf{s}_{\text{LC}}^{(i,j)} \parallel \mathbf{s}_{\text{FP}}^{(i,j)} \parallel \mathbf{s}_{\text{FD}}^{(i,j)}\right]. \tag{17}$$

Finally, by combining the global perception and local refinement representations, we obtain a comprehensive modeling of multi-scale topological signals, forming an effective universal LP representation.

A.2 Attention Mask and Transformer

To enforce the inductive setting in in-context inference, we design an attention mask $\mathbf{M} \in \mathbb{R}^{L_{seq} \times L_{seq}}$ to prevent information leakage among query samples. Specifically, domain prompts and support samples are fully visible, while each query can only attend to prompts, support, and itself. The mask is defined as:

$$m_{i,j} = \begin{cases} 0, & \text{if } j \leq K + |\mathcal{S}^+| + |\mathcal{S}^-| \text{ or } i = j, \\ -\infty, & \text{otherwise.} \end{cases} \tag{18}$$

We then feed the input sequence $\mathbf{Z}^{(0)}$ along with the mask $\mathbf{M}$ into an L-layer Transformer encoder to obtain the deep contextualized representation $\mathbf{Z}^{(L)}$, where each layer can be formulated as:

$$\mathbf{Z}^{(l+1)} = \text{FFN}\left(\text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d'}} + \mathbf{M}\right)\mathbf{V}\right), \tag{19}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the query, key, and value matrices, computed by $\mathbf{Q} = \mathbf{Z}^{(l)}\mathbf{W}_Q^{(l)}$, $\mathbf{K} = \mathbf{Z}^{(l)}\mathbf{W}_K^{(l)}$, and $\mathbf{V} = \mathbf{Z}^{(l)}\mathbf{W}_V^{(l)}$, with each $\mathbf{W}$ as learnable parameters, and d' represents the dimension of the attention heads.

B Algorithm of ATLASULP

The algorithm pseudo-code of ATLASULP is shown in Algo. 1.

C Experimental Details

C.1 Datasets

We collect graph datasets from six domains, including biology, transport, web, collaboration, citation, and social networks, to evaluate the generalization of universal link prediction (ULP) models. Such diversity introduces heterogeneous structural patterns across graphs, posing significant challenges for transferring link prediction knowledge. Following prior work on universal link prediction, we collect graph datasets from diverse domains, including biology (Ecoli, Yeast, Celegans) [47, 50, 55], transport (Power, USAir) [6, 50], web (PolBlogs, Router, PB) [1, 3, 46], collaboration (Physics, CS, NS) [35, 44], citation (Pubmed, Citeseer, Cora) [53], and social networks (Twitch, Github, Facebook) [43]. The datasets are partitioned into two disjoint groups: a pre-training (source) set and a testing (target) set. Table 2 summarizes the key statistics. Below we briefly describe each dataset.

Source (Pre-training) Datasets.

- **Ecoli** is a biological network where nodes represent operons and edges denote interactions.
- **Yeast** is a protein–protein interaction (PPI) network, where nodes represent proteins and edges denote physical interactions.
- **Power** is a power grid network, where nodes represent power plants or transformers and edges denote transmission lines.

Algorithm 1 Pseudo-Code of the Proposed ATLASULP

Require: Source-domain graphs $\{\mathcal{G}_s^{(1)}, \cdots, \mathcal{G}_s^{(N)}\}$;
Target-domain graph $\mathcal{G}_t$;
Model parameters Θ.
Ensure: Trained universal LP model with parameters Θ^*.
Step 1: Relation Atlas Extraction and Encoding
for each graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ **do**
  Compute global perception representation $\mathbf{H}_{global}$ according to Eqs. (1) and (2);
  Compute local refinement representation $\mathbf{H}_{local}$ according to Eqs. (12)– (17);
  Encoding final embedding of edges $\mathbf{h}_e$ according to Eq. (4);
end for
Step 2: Train Domain-guided Prompting ULP Model
for each training episode **do**
  Select background reference edges $\mathcal{C}_b^+ = \{e_1, \cdots, e_M\}$ and support sets $\mathcal{S}^+$ and $\mathcal{S}^-$;
  Domain-aware Prompt Learning:
  Encode domain-aware prompt token according to Eqs. (5) and (6);
  Prompt-guided In-context Prediction:
  Construct attention mask and the input sequence of Transformer according to Eqs. (18) and (8);
  Encode node representations via the Transformer according to Eqs. (19);
  Link Scoring:
  Compute link probabilities based on support-query similarity according to Eqs. (9) and (10);
  Update model parameters Θ by minimizing the BCE loss according to Eq. (11);
end for
Return trained model parameters Θ^*;
Training-Free Prediction on Target Domain
Extract relation atlas of the target graph $\mathcal{G}_t$;
Encode edge representations using the trained model with parameters Θ^*;
Compute link probabilities for target edge.

- **PolBlogs** is a web network, where nodes represent weblogs and edges denote hyperlink references.
- **Router** is a web network, where nodes represent routers and edges denote physical connections.
- **Physics** is a collaboration network, where nodes represent researchers and edges denote co-authorship relations.
- **Pubmed** is a citation network, where nodes represent scientific articles and edges denote citation relationships.
- **Citeseer** is a citation network, where nodes represent publications and edges denote citation links.
- **Twitch** is a social network, where nodes represent users and edges denote interactions.
- **Github** is a social network, where nodes represent users and edges denote follower relationships.

Target (Testing) Datasets.

- **Celegans** is a biological network, where nodes represent neurons and edges denote neural connections.
- **USAir** is a transportation network, where nodes represent airports and edges denote flight routes.
- **PB** is a web network, where nodes represent blogs and edges denote hyperlink references.
- **CS** is a collaboration network, where nodes represent authors and edges denote co-authorship relations.

Table 2: Statistics of the graph datasets from diverse domains.

Dataset	Pretrain	Test	# Nodes	# Edges	Avg. node deg.	Std. node deg.	Max. node deg.	Density
Biology								
Ecoli	✓	-	1805	29320	16.24	48.38	1030	1.8009%
Yeast	✓	-	2375	23386	9.85	15.5	118	0.8295%
Celegans	-	✓	297	4296	14.46	12.97	134	9.7734%
Transport								
Power	✓	-	4941	13188	2.67	1.79	19	0.1081%
USAir	-	✓	332	4252	12.81	20.13	139	7.7385%
Web								
PolBlogs	✓	-	1490	19025	12.77	20.73	256	1.7150%
Router	✓	-	5022	12516	2.49	5.29	106	0.0993%
PB	-	✓	1222	33428	27.36	38.42	351	4.4808%
Collaboration								
Physics	✓	-	34493	495924	14.38	15.57	382	0.0834%
CS	-	✓	18333	163788	8.93	9.11	136	0.0975%
NS	-	✓	1589	5484	3.45	3.47	34	0.4347%
Citation								
Pubmed	✓	-	19717	88648	4.5	7.43	171	0.0456%
Citeseer	✓	-	3327	9104	2.74	3.38	99	0.1645%
Cora	-	✓	2708	10556	3.9	5.23	168	0.2880%
Social								
Twitch	✓	-	34118	429113	12.58	35.88	1489	0.0737%
Github	✓	-	37700	289003	7.67	46.59	6809	0.0407%
Facebook	-	✓	22470	171002	7.61	15.26	472	0.0677%

- **NS** is a collaboration network, where nodes represent researchers and edges denote co-authorship relations.
- **Cora** is a citation network, where nodes represent publications and edges denote citation links.
- **Facebook** is a social network, where nodes represent users and edges denote interactions.

C.2 Comparison Methods

Classical Heuristic Methods

- **CN** [26] (Common Neighbors) measures link likelihood based on the number of shared neighbors between two nodes.
- **AA** [2] (Adamic-Adar) refines CN by assigning higher weights to low-degree common neighbors.
- **RA** [57] (Resource Allocation) is similar to AA but uses a different normalization scheme based on node degrees.
- **PA** [5] (Preferential Attachment) estimates link likelihood based on the product of node degrees.
- **SP** (Shortest Path) measures node proximity using the shortest path distance.
- **Katz** [20] sums over all paths between two nodes with exponential decay on longer paths.

Supervised and Pretraining-based Models

- **GAE** [22] is a graph auto-encoder that learns node embeddings and reconstructs graph structure for link prediction.
- **SEAL** [54] is a subgraph-based method that extracts enclosing subgraphs and applies GNNs for link prediction.

- **ELPH** [7] captures structural features via efficient message passing for scalable link prediction.
- **NCNC** [49] models common neighbors using neural architectures to enhance link prediction.
- **MPLP** [9] leverages message passing to estimate structural patterns for link prediction.

Universal Link Prediction Methods

- **UniLP** [10] is a universal link prediction framework that constructs structural representations via DRNL and performs in-context inference for cross-domain prediction.

C.3 Evaluation and Implementation

Evaluation Protocol

To evaluate the cross-domain effectiveness of ATLASULP under the universal link prediction (ULP) setting, we split all graph datasets into two disjoint groups, i.e., a pre-training (source) set and a testing (target) set (see Table 2). A single model is trained on the combined pre-training datasets and evaluated on all target datasets. To mitigate the imbalance caused by varying dataset sizes, we adopt a balanced episodic sampling strategy during pre-training. Specifically, we uniformly sample an equal number of edges from each source dataset per epoch, ensuring balanced contributions across different graph domains.

During pre-training, for each query link, we first sample 1000 candidate links from the corresponding source dataset as background reference examples $\mathcal{C}_b$. From $\mathcal{C}_b$, we further select 64 links as in-context samples. For each target dataset, we follow a standard LP split strategy, where edges are divided into 70%/10%/20% for training (observed edges), validation, and testing (unobserved edges), respectively. During inference, 1000 background reference links are sampled from each target dataset, from which 32 in-context samples are selected for prediction. We report Hits@50 as the evaluation metric. Hits@50 measures the proportion of true positive links ranked within the top 50 predictions among all candidate links, reflecting the ranking quality of the model. Our experiments are conducted on a single NVIDIA RTX 4000 Ada GPU using PyTorch 2.7.1 with Python 3.10.

Evaluation of Baselines

We evaluate all baseline methods under two transfer settings to ensure a fair comparison:

Pretrain Only: models are trained on the combined pre-training datasets and directly evaluated on each target dataset without adaptation.

Pretrain & Finetune: after pretraining, models are further fine-tuned on each target dataset using 200 sampled positive and negative links.

For GAE and NCNC, which require initial node features, we use a 32-dimensional all-one vector as input features. All other methods can directly handle non-attributed graphs without additional node features. All methods follow the same data splits and evaluation protocol to ensure a consistent and fair cross-domain comparison.

D Broader Impact

As a foundational framework for Universal Link Prediction, ATLASULP has potential positive societal impacts, such as accelerating discoveries in biological networks (e.g., protein-protein interactions) and improving the efficiency of large-scale recommendation systems.

We note that, in some applications, link prediction models may also raise potential privacy considerations when applied to sensitive social networks, where inferred connections could reveal latent relationships. Appropriate safeguards should be considered in such scenarios.

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: We clearly state our contributions in the abstract and introduction, and successfully validate them through extensive experiments in the subsequent sections.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 explicitly discuss the limitations of our work, specifically the reliance on manual design for constructing the Relation Atlas, in the Conclusion section.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The primary contribution of this paper is an empirical methodology rather than formal mathematical theorems, though a rigorous computational complexity analysis is provided in Appendix 4.3.

Guidelines:

- The answer [N/A] 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: All experimental details, including dataset splits, training protocols, and hyperparameters, are described in Appendix C.

Guidelines:

- The answer [N/A] 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 provide code and data access instructions in the supplementary material to ensure reproducibility.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed experimental settings, including data splits, sampling strategies, and hyper-parameters in Appendix C.

   Guidelines:

   - The answer [N/A] 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: We report the mean and standard deviation over 5 independent runs with different random seeds, where the standard deviation reflects the variability across runs.

   Guidelines:

   - The answer [N/A] 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: We describe the computational setup, including GPU usage and training configurations, in Appendix C.3.

   Guidelines:

   - The answer [N/A] 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: This research conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] 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: We discuss the potential positive applications and negative societal impacts (e.g., privacy concerns in social networks) of universal link prediction in the Appendix D.

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The proposed method does not involve high-risk data or model release.

    Guidelines:

    - The answer [N/A] 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: All datasets and resources used are properly cited and follow their respective usage licenses.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper evaluates the proposed model on existing public benchmark graph datasets and does not introduce new data assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This research does not involve crowdsourcing, data labeling by human workers, or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This research does not involve human subjects, hence IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the proposed method.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.