TFMLinker: Universal Link Predictor by Graph In-Context Learning with Tabular Foundation Models

Tianyin Liao [1] **Chunyu Hu** [1] **Yicheng Sui** [1] **Xingxuan Zhang** [2] **Peng Cui** [2] **Jianxin Li** [3] **Ziwei Zhang** [3]

Abstract

Link prediction is a fundamental task in graph machine learning with widespread applications. In the foundation model era, how to develop universal link prediction methods across datasets and domains becomes a key problem, with some initial attempts adopting Graph Foundation Models utilizing Graph Neural Networks and Large Language Models. However, the existing methods face notable limitations, including limited pre-training scale or heavy reliance on textual information. Motivated by the success of tabular foundation models in achieving universal prediction across diverse tabular datasets, we explore an alternative approach by TFMs, which are pre-trained on diverse synthetic datasets sampled from structural causal models and support in-context learning independent of textual attributes. Nevertheless, adapting TFMs for link prediction faces severe challenges such as how to obtain the necessary context and capture link-centric topological information. To solve these challenges, we propose **TFMLinker**, aiming to leverage the in-context learning capabilities of TFMs to perform link prediction across diverse graphs without requiring dataset-specific fine-tuning. Specifically, we first develop a prototype-augmented local-global context module to construct context that captures both graph-specific and cross-graph transferable patterns. Next, we design a universal topology-aware link encoder to capture link-centric topological information and generate link representations as inputs for the TFM. Finally, we employ the TFM to predict link existence through in-context learning. Experiments on 6 graph benchmarks across diverse domains show our method outperforms state-of-the-art baselines.

[1]Nankai University, Binhai New Area, Tianjin, China [2]Tsinghua University, Haidian District, Beijing, China [3]Beihang University, Haidian District, Beijing, China. Correspondence to: Ziwei Zhang <zwzhang@buaa.edu.cn>.

Workshop on Graph Foundation Models: A New Era for Graph Machine Learning, Seoul, South Korea. GFM@ICML 2026.

1. Introduction

Link prediction is a fundamental task in graph machine learning with broad applications across social networks, biology, recommendation systems, and beyond (Kovács et al., 2019; Daud et al., 2020). The goal is to infer missing or potential links within a graph from observed ones. Traditional link prediction methods typically rely on hand-crafted heuristics to model pairwise node relationships, such as common neighbors or shortest path distances. While graph neural networks (GNNs) have recently advanced link prediction performance (Chamberlain et al.; Yun et al., 2021; Wang et al.; Shomer et al., 2024), existing methods still face key challenges, including limited supervision due to sparse connectivity, sensitivity to initialization, and poor generalization under distribution shifts (Song et al., 2025; Li et al., 2023), which collectively constrain their applicability across diverse datasets and domains.

In the foundation model era, addressing these challenges to develop universal link prediction methods across domains becomes a key problem. Graph Foundation Models (GFMs), envisioned as models pre-trained on extensive graph data, have been proposed as promising solutions adaptable to diverse link prediction scenarios. Architecturally, GFMs can be classified into three main types (Liu et al., 2025): GNN-based (Song et al., 2025; Dong et al., 2025), LLM-based (Chen et al., 2024), and GNN+LLM-based models (Liu et al.). These architectures leverage different underlying mechanisms to enhance their generalization across diverse link prediction tasks. For example, GNN-based models can utilize graph-structured inductive biases to capture topological patterns, LLM-based models incorporate semantic understanding through natural language processing, and GNN+LLM-based models synergize structural and semantic information for more generalized representations.

However, despite their initial progress, these architectures all face severe limitations. For instance, GNN-based GFMs typically lack pre-training on sufficiently diverse and comprehensive datasets, which limits their ability to generalize across different domains and constrains their performance in tasks requiring broad contextual understanding (Liu et al., 2025). Meanwhile, both LLM-based and GNN+LLM-based GFMs often rely heavily on textual information, leading to

inconsistent data priors when applied to graphs that lack sufficient textual attributes (Eremeev et al., 2025a;b).

To overcome these limitations, and inspired by the success of tabular foundation models (TFMs) in achieving universal prediction across diverse tabular datasets, we propose exploring universal link prediction methods based on TFMs (Hollmann et al.; 2025; Zhang et al., 2025). These methods are pre-trained on diverse synthetic datasets sampled from structural causal models and can make predictions in an in-context learning setting. Recent works, such as TabPFNv2 (Hollmann et al., 2025), LimiX (Zhang et al., 2025), and GraphPFN (Eremeev et al., 2025b), have already demonstrated strong generalization capabilities independent of textual attributes, performing well on both tabular prediction and node classification tasks. Nevertheless, directly applying such TFMs to the link prediction problem faces challenges, primarily arising from the following aspects.

First, how to obtain the context necessary for such TFMs remains unexplored in link prediction tasks. A straightforward approach is to randomly downsample edges from the target graph to gain context. However, relying solely on the target graph only allows the model to learn patterns within a single task but neglects capturing the transferable patterns that exist across diverse tasks and graphs (Wu & Yin).

Second, how to enable TFMs to capture link-centric topological information remains unexplored. TabPFN and LimiX have been proven to be a context-learning model capable of handling data heterogeneity (Ye et al., 2025), showing potential in various domains. However, given their nature as tabular foundation models pre-trained on synthetic tabular data, they cannot inherently capture link-centric topological information or derive link representations from datasets rich in structural diversity.

To address these challenges, we propose **TFMLinker**, aiming to leverage the in-context learning capabilities of TFMs to perform link prediction across diverse graphs on-the-fly, without requiring explicit model fine-tuning tailored to specific downstream graph datasets. Specifically, **TFMLinker** achieves this goal by tailoring three modules for universal link prediction. First, we design a prototype-augmented local-global context module to construct context through random subgraph downsampling and learnable prototypes. This dual mechanism empowers our method to capture two distinct types of pattern: specific patterns within individual graphs using downsampling, and transferable patterns across diverse graphs using prototypes, respectively. Next, we introduce a universal topology-aware link encoder to capture link-centric topological information by multiple link encoders. In this way, we utilize the strengths of both heuristic link-centric methods and the pre-trained GNN to generate link representations, with each embedding vector forming a row of a pseudo table and used as inputs to the TFM, thereby providing it with rich topological information. Finally, we employ the TFM to predict link existence through in-context learning. We conduct extensive experiments on 6 graph benchmarks across various domains for link prediction. The results demonstrate that **TFMLinker** consistently and significantly outperforms various state-of-the-art baselines without requiring fine-tuning on target graphs. In-depth analyses further demonstrate the designs of our proposed method.

Our contributions are summarized as follows:

- We propose **TFMLinker**, a novel link prediction approach across datasets and domains leveraging the in-context learning abilities of TFMs. To the best of our knowledge, this work is the first exploration of adapting TFMs to the link prediction task.
- We develop a prototype-augmented local-global context module to construct context via random subgraph downsampling and learnable prototypes to capture both graph-specific and cross-graph patterns.
- We further design a universal topology-aware link encoder module to capture link-centric topological information by multiple link encoders.
- Extensive empirical results demonstrate **TFMLinker** outperforms state-of-the-art methods across datasets from diverse graph domains without requiring dataset-specific fine-tuning.

2. Preliminary and Problem Formulation

Link prediction. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^o)$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the node set of cardinality n, and $\mathcal{E}^o$ denotes the observed set of links. This set is a subset $\mathcal{E}^o \subseteq \mathcal{E}^*$ of the complete true link set $\mathcal{E}^* \subseteq \mathcal{V} \times \mathcal{V}$, which includes not only the observed links but also all potential links that are currently unobserved or may appear in the future within $\mathcal{G}$. For any two nodes $u, v \in \mathcal{V}$, we denote the shortest-path between them as $\mathrm{SP}(u, v)$. The goal of link prediction tasks is to identify the set of unobserved true links $\mathcal{E}^u \subseteq \mathcal{E}^* \backslash \mathcal{E}^o$ in the given graph $\mathcal{G}$.

Tabular Foundation Model. A tabular dataset is denoted as $\mathcal{T} = (\mathbf{X}, \mathbf{y}')$, where $\mathbf{X} \in \mathbb{R}^{n \times d}$ contains n rows and d columns, and the label vector $\mathbf{y}' \in \mathbb{R}^n$ offers supervision for a downstream task. A TFM is a pre-trained model $f(\cdot)$ designed to learn a universal inference mechanism, thereby enabling generalization to new and unseen tabular datasets without dataset-specific training. State-of-the-art TFMs, such as TabPFN (Hollmann et al.; 2025) and LimiX (Zhang et al., 2025), accomplish this capability by first pre-training on a variety of synthetic datasets sampled from a carefully

designed prior distribution, and then performing in-context Bayesian inference (Liu et al., 2026):

$$p(\mathbf{y}'_q|\mathbf{X}_q, \mathbf{X}_c, \mathbf{y}'_c) \approx f(\mathbf{X}_q, \mathbf{X}_c, \mathbf{y}'_c), \quad (1)$$

where the pairs $(\mathbf{X}_q, \mathbf{y}'_q)$ and $(\mathbf{X}_c, \mathbf{y}'_c)$ refer to the test and in-context samples, respectively.

An intuitive approach to adapt TFMs for link prediction is to treat each link as a table row and encode them into representations $\mathbf{Z}$. Observed positive links and randomly sampled negative links can serve as contexts $(\mathbf{Z}_c, \mathbf{y}_c)$. Given the contexts, TFMs perform in-context learning on the representation of target links $\mathbf{Z}_q$ to predict the existence labels:

$$\hat{\mathbf{y}}_q = f(\mathbf{Z}_q, \mathbf{Z}_c, \mathbf{y}_c). \quad (2)$$

However, this straight-forward approach faces two significant challenges. First, graphs for link prediction typically contain far more links than the context length capacity of TFMs can accommodate (Ye et al., 2025; Feuer et al., 2024). An excessively large context not only fails to improve accuracy but also incurs substantial computational overhead. Randomly downsampling observed links can alleviate computational costs, but it also introduces sampling noise, causing the model to overfit to specific patterns within a single graph and neglect transferable patterns across different graphs in link prediction tasks. Second, it remains unclear how to obtain meaningful link representations $\mathbf{Z}$, since TFMs are pre-trained on tabular data and inherently lack the ability to capture complex graph topological information.

3. Method

In this section, we introduce **TFMLinker**, as illustrated in Figure 1. The prototype-augmented local-global context module builds context via random subgraph sampling and learnable prototypes. The universal topology-aware link encoder then generates representations for context and target links. Finally, the TFM performs in-context learning on the link representations to predict link existence.

3.1. Prototype-Augmented Local-Global Context

The prototype-augmented local-global context module is designed to construct the context necessary for the TFMs' in-context learning setting. As discussed in Section 2, sampling context links solely from a single graph $\mathcal{G}$ faces the challenge of sampling noise, which can lead the model to overfit to task-specific patterns. In particular, the in-context learning performance of the model fluctuates with the sampled context links: performance tends to be better when the sampled context links are similar to the target links, and worse when they are dissimilar.

Therefore, a natural solution is to improve performance in cases where context links are dissimilar to the target links. Given the inductive biases commonly used in various link prediction tasks, such as the principle that node pairs sharing common neighbors are more likely to be linked (Wang et al.), we hypothesize that there exist certain transferable patterns common to all link prediction tasks, even across different domains with distinct topological characteristics. If we can obtain additional context that represents such transferable patterns to complement the sampled context links, it would provide supplementary information, thereby improving the model's in-context learning performance, especially when the sampled context links differ significantly from the target links. Accordingly, we refer to the sampled context links as the local context, and the context representing transferable patterns as the global context.

For the local context, we select k connected node pairs $\mathcal{S}^+ = \{s_1^+, s_2^+, \cdots, s_k^+\} \subseteq \mathcal{E}^o$ as positive samples. Similarly, we collect negative samples $\mathcal{S}^- = \{s_1^-, s_2^-, \cdots, s_k^-\} \subseteq \mathcal{V} \times \mathcal{V} \backslash \mathcal{E}^o$, comprising k node pairs without link. The union $\mathcal{S}^+ \cup \mathcal{S}^-$ approximates the overall properties of $\mathcal{G}$ while remaining computationally efficient. Once the positive and negative samples are obtained, we need to extract their structural representations. Following (Zhang & Chen, 2018; Dong et al., 2025), we start by extracting the ego-subgraph for each sample. For a node pair s, its ego-subgraph $\mathcal{G}^{\text{ego}}_{s,r}$ is defined as the subgraph induced by all r-hop neighboring nodes of the nodes in s on the original graph $\mathcal{G}$. For simplicity, we use 1-hop ego-subgraphs for all positive and negative samples $\mathcal{S}^+ \cup \mathcal{S}^-$ to form the local context:

$$\mathcal{C}_l = \{\mathcal{G}^{\text{ego}}_{s,1} | s \in \mathcal{S}^+ \cup \mathcal{S}^-\}. \quad (3)$$

In addition, to ensure data consistency between the context and target links, we also construct the 1-hop ego-subgraph for the target link $q \in \mathcal{V} \times \mathcal{V}$, denoted as $\mathcal{G}^{\text{ego}}_{q,1}$.

For the global context, inspired by prototype learning (Lin et al., 2022), we maintain a single set of learnable and shared prototypes $\mathbf{Z}_g \in \mathbb{R}^{p \times d}$ along with their associated labels $\mathbf{y}_g$ to capture the global task distribution across all link prediction tasks. Here, p denotes the number of prototypes, d is the dimension of the link representations, and the labels in $\mathbf{y}_g$ are evenly split between 0 and 1. By sharing the same prototypes across diverse tasks, we encourage them to learn transferable patterns. During both training and inference, these prototypes, referred to as the global context, are directly concatenated with the representations of the local context to form the input of the TFM.

The key distinction between our approach and other works that also employ learnable parameters as input to TFMs, often referred to as prompt tuning (Ma et al., 2024a; Feuer et al., 2024), lies in their fundamental objectives. While prompt tuning aims to adapt the TFM to different downstream datasets, our prototypes are designed to capture the

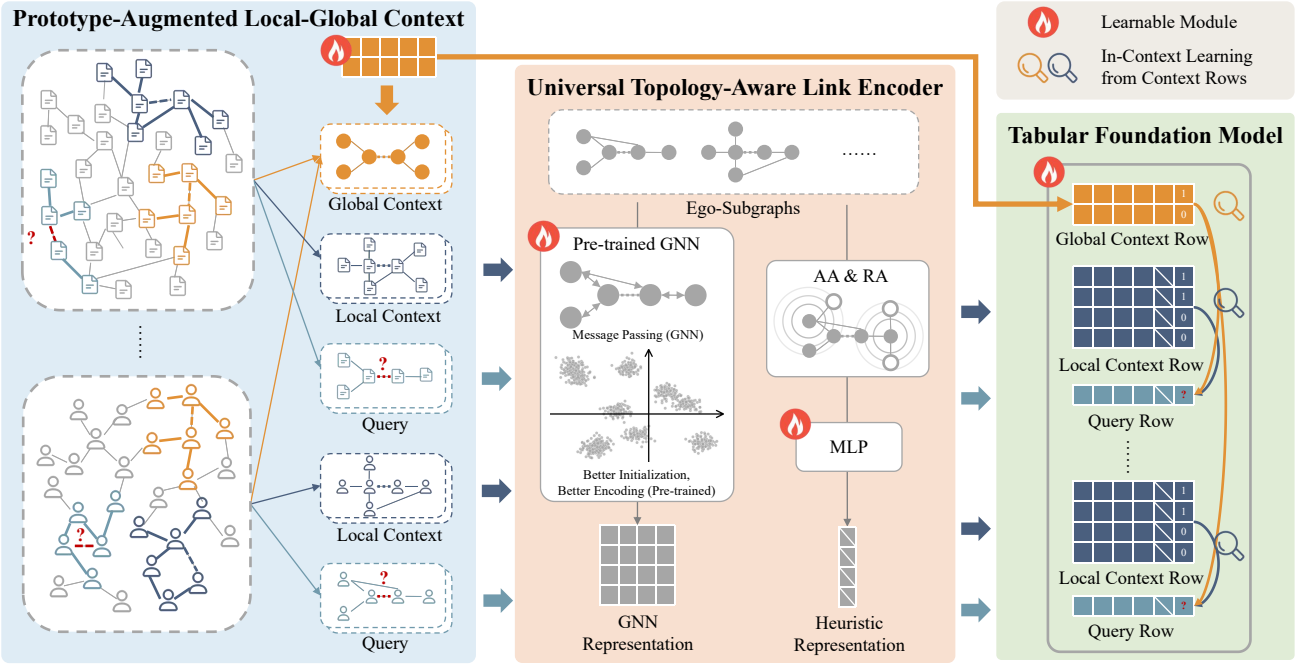

Figure 1. The framework of **TFMLinker**. Our proposed method is composed of three modules: (1) The prototype-augmented local-global context module constructs context through random subgraph downsampling and learnable prototypes, capturing both graph-specific and cross-graph transferable patterns. (2) The universal topology-aware link encoder module employs multiple link encoders to generate representations of context and target links. These representations provide rich and accurate link-centric topological information for the TFM. (3) The TFM performs in-context learning on the representations of context and target links to predict link existence labels.

shared global context across all link prediction tasks, and they are not fine-tuned for each dataset. Consequently, their implementations also differ substantially. Prompt tuning typically requires a large number of learnable parameters (e.g., 1000 rows) to fit the distribution of a given downstream dataset. In contrast, the number of prototypes in our method is kept small (e.g., fewer than 100), as an excessive amount would shift the model's focus overly toward the global context and dilute dataset-specific characteristics.

3.2. Universal Topology-Aware Link Encoder

After obtaining the context using the prototype-augmented local-global context module, we address the fundamental task of transforming the graph-structured data into a two-dimensional, tabular format suitable for the TFM. This step is essential, as TFMs such as LimiX (Zhang et al., 2025) are inherently designed and pre-trained on tabular data, and therefore lack the capacity to capture complex graph topological information. To bridge this gap, we introduce a universal topology-aware link encoder designed to generate representations for both context and target links for the TFM to perform in-context learning.

Specifically, the representation of each context and target link is used to construct a row of a pseudo table, where each column corresponds to one dimension of the representation. The TFM then takes this pseudo table as input and performs in-context learning for the tabular task, predicting the label of the target row (i.e. the label of the target link) based on the labeled context rows and the unlabeled target row. Crucially, the quality of the link representations used to construct the pseudo-table is essential: if the table itself is flawed or information-poor, the model cannot be expected to predict the target correctly. To obtain rich and accurate representations of links, we design our encoder to leverage multiple encoding approaches, including both learning-based methods such as GNNs and heuristic methods such as common neighbors, which have demonstrated strong potential in link prediction tasks (Ma et al., 2024b).

For the learning-based link encoder, we enhance the expressiveness of GNNs (Hamilton et al., 2017) in capturing graph structures by employing the labeling trick technique (Zhang & Chen, 2018; Dong et al., 2025) to obtain link representations. Specifically, we first use DRNL+ (Dong et al., 2025), a variant of Double Radius Node Labeling (Zhang & Chen, 2018) and Distance Encoding (Li et al., 2020), to generate structural features of each node in the ego-subgraph $\mathcal{G}_{e,1}^{\text{ego}}$:

$$\text{DRNL+}(i,e) = \begin{cases} (0, \text{SP}(i,u)), & \text{if SP}(i,v) = \infty, \\ (0, \text{SP}(i,v)), & \text{if SP}(i,u) = \infty, \\ (\text{DRNL}(\text{i},\text{e}), 0), & \text{otherwise}, \end{cases} \tag{4}$$

where $e = (u, v)$ denotes a node pair from the target link or the context links, i.e., $e \in \{q\} \cup \mathcal{S}^+ \cup \mathcal{S}^-$, and $\mathrm{DRNL(i, e)}$ is the integer label to node i by Double Radius Node Labeling in $\mathcal{G}^{\text{ego}}_{e,1}$. After the labeling trick encodes the structural information, we apply a pre-trained GNN $g(\cdot)$ from UniLP (Dong et al., 2025) to update node representations. Following (Dong et al., 2025), the ego-subgraph representation h_e for each link e is obtained by average pooling $\mathrm{Pool}(\cdot)$ over all node representations in $\mathcal{G}^{\text{ego}}_{e,1}$:

$$\mathbf{h}_e = \mathrm{Pool}(g(\mathbf{X}_{\text{ego}}, \mathbf{A}_{\text{ego}})), \tag{5}$$

where $\mathbf{X}_{\text{ego}}$ is the node features formed by $\mathrm{DRNL{+}}(i, e)$ for each node i in $\mathcal{G}^{\text{ego}}_{e,1}$, and $\mathbf{A}_{\text{ego}}$ is the adjacency matrix of $\mathcal{G}^{\text{ego}}_{e,1}$. We employ a pre-trained GNN instead of a randomly initialized one because the latter tends to overfit the specific in-context learning patterns of the TFM during early training stages, which results in performance degradation due to reduced model expressiveness. Using a pre-trained GNN effectively alleviates this issue, allowing the TFM to conduct in-context learning based on richer representations.

For the heuristic methods, we consider Adamic-Ada (AA) (Adamic & Adar, 2003) and Resource Allocation (RA) (Zhou et al., 2009) to provide additional structural information. This is motivated by the fact that the 1-hop ego-subgraph captures only local connectivity, and thus lacks awareness of structural patterns beyond the immediate neighborhood. Specifically, we apply an MLP $w(\cdot)$ to generate the link representation from the heuristic methods for the node pair e:

$$\mathbf{h}'_e = w([\mathrm{AA(e)}, \mathrm{RA(e)}]). \tag{6}$$

After obtaining link representations from both the GNN and heuristic methods, we concatenate these representations to form the final link representation $\mathbf{z}_e = [\mathbf{h}_e, \mathbf{h}'_e] \in \mathbb{R}^d$ for link e, where d denotes the dimensionality. The local context representations $\mathbf{Z}_l$ are then obtained by concatenating the representation $\mathbf{z}_s$ of each local context $s \in \mathcal{S}^+ \cup \mathcal{S}^-$. Similarly, we derive the representation $\mathbf{z}_q$ of the target link q. These representations are used as inputs for the TFM.

3.3. Tabular Foundation Model

In this section, we introduce how TFM performs in-context learning on the representations of context and target links to predict link existence labels. First, we combine the local and global context representations and ground-truth labels to obtain the final context representations and context labels:

$$\mathbf{Z}_c = [\mathbf{Z}_l, \mathbf{Z}_g], \mathbf{y}_c = [\mathbf{y}_l, \mathbf{y}_g], \tag{7}$$

where $\mathbf{y}_l$ and $\mathbf{y}_g$ are the ground-truth labels of the local and global context, respectively. Next, we use the TFM $f(\cdot)$, such as TabPFN (Hollmann et al., 2025) and LimiX (Zhang et al., 2025), to perform tabular-based in-context inference:

$$\hat{y}_q = f(\mathbf{z}_q, \mathbf{Z}_c, \mathbf{y}_c). \tag{8}$$

To adapt **TFMLinker** to diverse link prediction tasks, we perform full fine-tuning of the entire model without employing other complex optimization strategies, as full fine-tuning has been shown to achieve the fastest convergence among fine-tuning methods for such TFMs (Rubachev et al., 2025). Given a set of graphs $\{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_n\}$ from different tasks, where each graph $\mathcal{G}$ belongs to this set, the optimization objective is defined as follows:

$$\min_{\Theta} \mathbb{E}_{\mathcal{G} \in \{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_n\}, q \in \mathcal{V} \times \mathcal{V}} [\ell(\hat{y}_q, y_q)], \tag{9}$$

where Θ are all learnable parameters, including those of the prototypes $\mathbf{Z}_g$, the GNN $g(\cdot)$, the MLP $w(\cdot)$, and the TFM $f(\cdot)$. $\ell(\cdot)$ denotes the loss function on a single instance, and y_q is the ground-truth existence label of link q.

4. Experiments

In this section, we conduct extensive experiments focusing on three questions: **RQ1**: Can **TFMLinker** effectively improve universal link prediction abilities across datasets and domains without fine-tuning? **RQ2**: How effective is **TFMLinker** under different quantities of in-context links? **RQ3**: How do the key components influence the results?

4.1. Experimental Setup

Datasets. The training of **TFMLinker** is based on a collection of graph datasets spanning diverse domains. Following (Mao et al.; Dong et al., 2025), we utilize graph data from fields such as biology (Von Mering et al., 2002; Watts & Strogatz, 1998; Zhang et al., 2018), transportation (Watts & Strogatz, 1998), the web (Ackland, 2005; Adamic & Glance, 2005; Spring et al., 2002), academic collaboration (new, 2006; Shchur et al., 2018), citation networks (Yang et al., 2016), and social networks (Rozemberczki et al., 2021). This diverse selection ensures that our training and evaluation of link prediction models cover a wide spectrum of connectivity patterns. For training, we use datasets such as Ecoli, Yeast, Power, PolBlogs, Router, Physics, Pubmed, Citeseer, Twitch, and Github. For evaluation, the test set comprises Celegans, USAir, PB, NS, CS, Cora, and Facebook. Details of the curated graph datasets are provided in Appendix B.1.

Baselines. We compare **TFMLinker** with state-of-the-art link prediction methods from three groups. The first group consists of heuristic methods, including Adamic-Adar (AA) (Adamic & Adar, 2003), Resource Allocation (RA) (Zhou et al., 2009), and Katz index (Katz) (Katz, 1953). The second group is GNN-based parametric link predictors

Table 1. Performance comparison with competitive link prediction methods on test datasets evaluated by Hits@50. Numbers after ± indicate variances. The best results are in **bold** and the second-best results are underlined.

	Biology C.ele	Transport USAir	Web PB	Collaboration NS	Collaboration CS	Citation Cora	Social Facebook	Average	Ave. Rank
Heuristics									
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	7.71
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	7.29
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	9.00
Pre-train Only GNN									
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	7.71
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	14.86
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	7.71
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	13.14
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	10.57
Pre-train & Fine-tune GNN									
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	4.86
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	15.29
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	6.71
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	5.57
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	8.00
GFM									
OpenGraph	55.94±4.33	72.38±7.33	28.85±2.79	85.57±1.50	37.68±4.83	59.17±2.83	13.52±17.42	50.44	11.00
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	3.86
TFMLinker	**68.62**±5.97	**91.67**±1.37	47.75±2.76	**90.46**±2.43	70.42±1.91	**62.63**±1.77	**69.06**±1.12	**71.52**	**1.86**

that include GAE (Kipf & Welling, 2016), SEAL (Zhang & Chen, 2018), ELPH (Chamberlain et al.), NCNC (Wang et al.), and MPLP (Dong et al., 2024). The third group includes GFMs for cross-domain link prediction, such as OpenGraph (Xia et al., 2024) and UniLP (Dong et al., 2025). Following the setup in (Dong et al., 2025), we provide GAE and NCNC with a 32-dimensional all-one vector as initial node features, as they require attributed input.

Evaluation. Following the evaluation protocol of (Dong et al., 2025), we assess the in-context learning performance of UniLP and **TFMLinker** on unseen datasets. The graph data is partitioned into non-overlapping training and test sets. A single model is trained on the combined training datasets, during which 40 positive and 40 negative links are dynamically sampled from each corresponding training dataset to serve as in-context links for every target link. For evaluation, each test dataset is further divided into training, validation, and test subsets in a 70%:10%:20% ratio. The training subset corresponds to observed links, whereas the validation and test subsets represent unobserved links. During inference, we sample 200 positive and 200 negative links from each test dataset to form the in-context set. Other baselines follow a comparable protocol. For GNNs, we adopt two experimental settings. In the first setting, termed Pre-train Only, models are trained on the combined pretraining datasets and then directly evaluated on each test dataset. In the second setting, referred to as Pre-train & Fine-tune, models are first pretrained and then additionally fine-tuned on each test dataset using 200 sampled positive and 200 sampled negative links for training. For OpenGraph, we use the original model weights and similarly fine-tune it on each test dataset using 400 links.

Implementation Details. We employ a 3-layer GraphSAGE (Hamilton et al., 2017) with 128-dimensional hidden representations as the GNN and employ the LimiX-16M (Zhang et al., 2025) as the default TFM. The heuristic-based link representations have a dimensionality of 16. For each target link, we sample 40 positive and 40 negative links from the corresponding training dataset as in-context links. All experimental results are reported as mean ± standard deviation over 10 independent runs. Additional implementation details are provided in Appendix B.2.

4.2. Performance Comparisons

To answer **RQ1**, we report the results of **TFMLinker** compared to baseline methods on various unseen graph datasets in Table 1. The results show that **TFMLinker** consistently outperforms heuristic methods, GNN-based link prediction models, and GFMs, achieving improvements on 5 out of 7 benchmark datasets. For example, on the C.ele dataset,

TFMLinker achieves a 5.25% improvement in Hits@50 over the second-best method. Moreover, compared to the state-of-the-art in-context learning GFM UniLP, our method advances the overall ranking by 1.85 positions and raises the average metric by 5.18%.

Furthermore, **TFMLinker** demonstrates strong generalization across a wide range of unseen graph datasets, achieving performance comparable or even superior to fine-tuned GNN-based link prediction models, despite not being explicitly trained on the test dataset. This capability stems from its in-context learning mechanism based on TFMs, which enables the model to adapt on-the-fly to specific graph datasets without additional training. By leveraging in-context links during inference, **TFMLinker** dynamically adjusts its understanding of connectivity patterns, resulting in generalizable and flexible performance.

4.3. In-Depth Analyses

4.3.1. Adaptability to Context Length

To answer **RQ2**, we conduct experiments with varying in-context link numbers k, sampled from each test graph in the range of 50 to 400. The results on the USAir, NS, Cora datasets, and the average performance across all seven test datasets are presented in Figure 2. These sampled links serve as contextual input for the model, with UniLP employed as the base model. The findings show that **TFMLinker**'s performance consistently improves as the number of in-context links increases, indicating that our method can more effectively capture the properties of the target graph when provided with richer contextual information. Moreover, regardless of k, **TFMLinker** consistently achieves average performance superior to the best baseline, demonstrating the generalization of its in-context learning ability across different numbers of contextual samples.

4.3.2. Ablation Studies

To answer **RQ3**, we evaluate three key modules of **TFMLinker**: the prototype-augmented local-global context module, the universal topology-aware link encoder, and the TFM. For the first module, we compare against a variant that removes the local context while retaining the global context (denoted as *w/o local context*), and a variant that removes the global context while retaining the local context (denoted as *w/o global context*). For the encoder, we test three configurations: (1) *w/o GNN pre-train*, where we use random initialization instead of pre-trained GNN weights; (2) *w/o GNN module*, where we remove the GNN module; and (3) *w/o heuristics*, where we exclude heuristic methods. For the TFM, we examine the effect of random initialization in place of pre-trained TFM weights (denoted as *w/o TFM pre-train*). The results are shown in Table 2.

Removing either local or global context hurts performance, indicating both are crucial for in-context learning. Removing local context impairs task-specific pattern capture, especially on larger datasets: PB and Facebook. Removing global context reduces the module to random sampling, losing cross-graph transferable patterns. Notably, the variant retaining only the global context still achieves reasonable generalization on C.ele, USAir, NS, and CS. This provides direct evidence that the global context alone is sufficient to capture transferable patterns across graphs.

The ablation experiments on the encoder similarly lead to performance degradation. Replacing pre-trained GNN weights with random initialization results in noticeable performance drop, which underscores the importance of a pre-trained encoder in enabling the TFM to grasp topological graph structures. One plausible explanation is that without pre-training, the GNN tends to overfit to the specific contextual relationships present in a particular graph dataset, thereby losing its ability to capture fine-grained topological information. Moreover, removing either the entire GNN module or heuristic methods consistently degrades the results, confirming that diverse link encoders are crucial for capturing rich structural information in graphs.

Randomly initializing the TFM degrades performance, confirming the value of a pre-trained TFM for cross-benchmark in-context learning. The results show that pre-training on causal synthetic data aids adaptation to downstream tasks.

5. Related Work

Link Prediction. The field of link prediction has long been dominated by two primary paradigms: heuristic methods and GNNs for Link Prediction (GNN4LP). Heuristic approaches rely on manually designed topological measures, such as local similarity indices (Adamic & Adar, 2003; Barabási & Albert, 1999; Liben-Nowell & Kleinberg, 2003; Zhou et al., 2009) or global path-based metrics (Katz, 1953; Page et al., 1999), to estimate the likelihood of missing edges. On the other hand, GNN4LP methods learn structural representations through data-driven architectures. GAE (Kipf & Welling, 2016) pioneered the use of graph autoencoders to learn node embeddings for link scoring. SEAL (Zhang & Chen, 2018) further argued that effective link prediction requires explicit link-level structural encoding and introduced the labeling trick to enable GNNs to capture pairwise structural patterns. Subsequent works such as NCNC (Wang et al.) and MPLP (Dong et al., 2024) have pushed scalability and performance, establishing strong results across standard benchmarks. Nevertheless, both paradigms face limitations: heuristic methods cannot fully capture effective structural features, while GNN4LP methods lack generalization across datasets or domains.

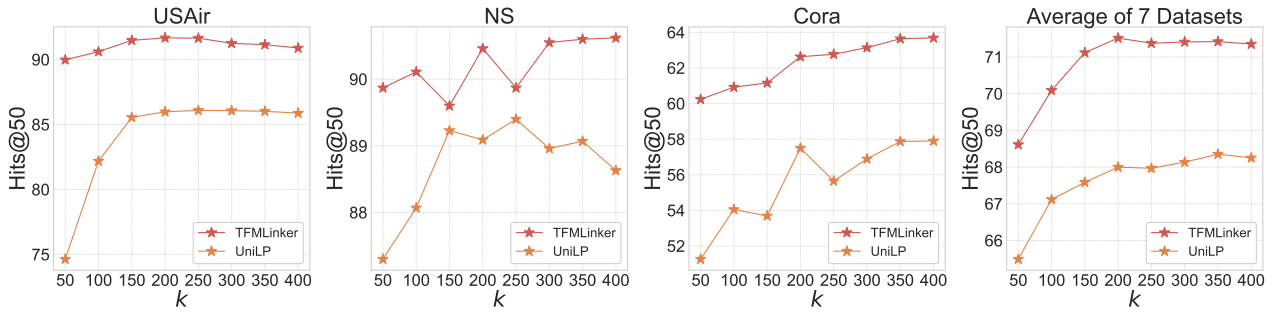

Figure 2. The impact of the number of in-context links k.

Table 2. The results of ablation studies on **TFMLinker**. The best results are in **bold** and the second-best results are underlined.

	Biology C.ele	Transport USAir	Web PB	Collaboration NS	Collaboration CS	Citation Cora	Social Facebook	Average
TFMLinker	68.62±5.97	**91.67**±1.37	47.75±2.76	**90.46**±2.43	**70.42**±1.91	62.63±1.77	69.06±1.12	**71.52**
w/o local context	65.03±6.74	88.80±2.03	1.57±0.37	80.46±2.99	57.03±1.43	40.22±1.75	27.47±0.43	51.51
w/o global context	66.97±5.34	90.59±1.83	45.23±3.00	88.74±1.76	62.10±7.75	61.68±1.58	69.27±1.51	69.23
w/o GNN pre-train	**68.95**±6.66	91.20±1.81	47.56±2.17	81.95±2.55	68.08±2.46	60.93±1.49	**70.49**±1.64	69.88
w/o GNN module	62.17±6.29	88.52±1.33	44.84±2.19	77.79±3.17	68.81±0.96	38.12±3.42	68.89±0.73	64.16
w/o heuristics	67.65±5.76	90.24±1.50	**48.48**±2.83	88.32±1.73	69.27±2.34	**63.18**±1.53	67.52±1.82	70.67
w/o TFM pre-train	66.92±6.64	87.39±2.12	37.84±5.08	85.82±1.59	59.36±1.35	50.67±2.74	62.25±1.27	64.32

Graph Foundation Model. Architecturally, GFMs are organized into three primary categories: GNN-based (Huang et al., 2023; Song et al., 2025; Dong et al., 2025), LLM-based (Chen et al., 2024), and GNN+LLM-based models (Liu et al.; Yang et al., 2025). GNN-based models capture topological patterns through graph-structured inductive biases. For instance, UniLP (Dong et al., 2025) employs in-context learning driven by GNN and attention mechanisms to generalize across unseen graph datasets without requiring task-specific training. LLM-based models integrate semantic understanding via natural language processing. As an example, InstructGLM (Ye et al., 2024) expands the vocabulary of a pre-trained language model by representing node features as distinct tokens, thereby enhancing cross-modal graph comprehension. GNN+LLM-based models combine structural and semantic information to produce more generalized representations. This is exemplified by OFA (Liu et al.), which aligns textual descriptions of nodes and edges into a unified embedding space through language modeling, enabling cross-modal graph generalization. Nevertheless, existing methods still face inherent limitations due to their foundational architectures, such as limited pre-training scale (Liu et al., 2025) or heavy reliance on textual information (Eremeev et al., 2025a;b).

Tabular Foundation Model. Recently, TabPFN (Hollmann et al.; 2025) and LimiX (Zhang et al., 2025) have demonstrated state-of-the-art performance and efficiency on small-to-medium-scale tabular data by fitting diverse generative processes with prior data, contributing to the growing recognition of TFMs as a distinct paradigm. Concurrent works such as TuneTables (Feuer et al., 2024) and BETA (Liu & Ye) further scale these approaches to real-world and larger datasets, narrowing the performance gap between TFMs and classical methods in scenarios with larger sample sizes. Currently, new TFMs are emerging regularly, and their success has extended beyond pure tabular tasks to applications such as time series forecasting (Hoo et al., 2025), node classification (Eremeev et al., 2025a;b), and graph anomaly detection (Liu et al., 2026). In this work, we contribute to this expanding scope by demonstrating, for the first time, that TFMs can also effectively serve as a powerful core component for the challenging task of universal link prediction across diverse graphs, which further broadens the potential application landscape of TFMs.

6. Conclusion

In this work, we propose **TFMLinker**, a novel link prediction approach across graph datasets and domains, leveraging TFMs for flexible adaptation to new graphs without fine-tuning. Our method effectively achieves this goal by integrating three key components, including the prototype-augmented local-global context module, the universal topology-aware link encoder, and the TFM. Extensive experiments on 6 diverse graph benchmark datasets demonstrate the superiority of **TFMLinker** over state-of-the-art methods without requiring dataset-specific finetuning.

Impact Statement

This paper presents work whose goal is to advance the field of graph machine learning, specifically in the area of universal link prediction. Our proposed **TFMLinker** leverages TFMs with in-context learning to perform link prediction across diverse graphs without requiring dataset-specific fine-tuning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. We believe that advancing the generalization capabilities of link prediction methods can benefit a wide range of applications, including recommendation systems, drug discovery, and knowledge graph completion, without introducing disproportionate negative societal impacts.

References

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

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

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

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

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

Chamberlain, B. P., Shirobokov, S., Rossi, E., Frasca, F., Markovich, T., Hammerla, N. Y., Bronstein, M. M., and Hansmire, M. Graph neural networks for link prediction with subgraph sketching. In *The Eleventh International Conference on Learning Representations*.

Chen, Z., Mao, H., Li, H., Jin, W., Wen, H., Wei, X., Wang, S., Yin, D., Fan, W., Liu, H., et al. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2): 42–61, 2024.

Daud, N. N., Ab Hamid, S. H., Saadoon, M., Sahran, F., and Anuar, N. B. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, 166:102716, 2020.

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

Dong, K., Mao, H., Guo, Z., and Chawla, N. V. Universal link predictor by in-context learning on graphs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Eremeev, D., Bazhenov, G., Platonov, O., Babenko, A., and Prokhorenkova, L. Turning tabular foundation models into graph foundation models. In *New Perspectives in Graph Machine Learning*, 2025a.

Eremeev, D., Platonov, O., Bazhenov, G., Babenko, A., and Prokhorenkova, L. Graphpfn: A prior-data fitted graph foundation model. *arXiv preprint arXiv:2509.21489*, 2025b.

Feuer, B., Schirrmeister, R. T., Cherepanova, V., Hegde, C., Hutter, F., Goldblum, M., Cohen, N., and White, C. Tunetables: Context optimization for scalable prior-data fitted networks. *Advances in Neural Information Processing Systems*, 37:83430–83464, 2024.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Hoo, S. B., Müller, S., Salinas, D., and Hutter, F. From tables to time: How tabpfn-v2 outperforms specialized time series forecasting models. *arXiv preprint arXiv:2501.02945*, 2025.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

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

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

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

Kovács, I. A., Luck, K., Spirohn, K., Wang, Y., Pollis, C., Schlabach, S., Bian, W., Kim, D.-K., Kishore, N., Hao, T., et al. Network-based prediction of protein interactions. *Nature communications*, 10(1):1240, 2019.

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

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

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

Lin, S., Liu, C., Zhou, P., Hu, Z.-Y., Wang, S., Zhao, R., Zheng, Y., Lin, L., Xing, E., and Liang, X. Prototypical graph contrastive learning. *IEEE transactions on neural networks and learning systems*, 35(2):2747–2758, 2022.

Liu, H., Feng, J., Kong, L., Liang, N., Tao, D., Chen, Y., and Zhang, M. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations*.

Liu, J., Yang, C., Lu, Z., Chen, J., Li, Y., Zhang, M., Bai, T., Fang, Y., Sun, L., Yu, P. S., et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Liu, S. and Ye, H.-J. Tabpfn unleashed: A scalable and effective solution to tabular classification problems. In *Forty-second International Conference on Machine Learning*.

Liu, Y., He, T., Liu, Y., Yi, C., Jin, H., and Hong, C. Tabular foundation models are strong graph anomaly detectors. *arXiv preprint arXiv:2601.17301*, 2026.

Ma, J., Thomas, V., Yu, G., and Caterini, A. In-context data distillation with tabpfn. *arXiv preprint arXiv:2402.06971*, 2024a.

Ma, L., Han, H., Li, J., Shomer, H., Liu, H., Gao, X., and Tang, J. Mixture of link predictors on graphs. *Advances in Neural Information Processing Systems*, 37:16043–16070, 2024b.

Mao, H., Li, J., Shomer, H., Li, B., Fan, W., Ma, Y., Zhao, T., Shah, N., and Tang, J. Revisiting link prediction: a data perspective. In *The Twelfth International Conference on Learning Representations*.

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.

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

Rubachev, I., Kotelnikov, A., Kartashev, N., and Babenko, A. On finetuning tabular foundation models. *arXiv preprint arXiv:2506.08982*, 2025.

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

Shomer, H., Ma, Y., Mao, H., Li, J., Wu, B., and Tang, J. Lpformer: An adaptive graph transformer for link prediction. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2686–2698, 2024.

Song, Y., Hua, Z., Shomer, H., Xie, Y., Liu, J., Long, B., and Liu, H. A scalable pretraining framework for link prediction with efficient adaptation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2621–2632, 2025.

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

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

Wang, X., Yang, H., and Zhang, M. Neural common neighbor with completion for link prediction. In *The Twelfth International Conference on Learning Representations*.

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

Wu, H. and Yin, J. Moemeta: Mixture-of-experts meta learning for few-shot relational learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xia, L., Kao, B., and Huang, C. Opengraph: Towards open graph foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2365–2379, 2024.

Yang, J., Wang, R., Yang, C., Yan, B., Zhou, Q., Juan, Y., and Shi, C. Harnessing language model for cross-heterogeneity graph knowledge transfer. In *Proceedings*

of the AAAI Conference on Artificial Intelligence, volume 39, pp. 13026–13034, 2025.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Ye, H.-J., Liu, S.-Y., and Chao, W.-L. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Ye, R., Zhang, C., Wang, R., Xu, S., and Zhang, Y. Language is all a graph needs. In *Findings of the association for computational linguistics: EACL 2024*, pp. 1955–1973, 2024.

Yun, S., Kim, S., Lee, J., Kang, J., and Kim, H. J. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. *Advances in Neural Information Processing Systems*, 34:13683–13694, 2021.

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

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

Zhang, X., Ren, G., Yu, H., Yuan, H., Wang, H., Li, J., Wu, J., Mo, L., Mao, L., Hao, M., et al. Limix: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025.

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

Table 3. Statistics of training datasets.

Dataset	Domain	# Nodes	# Edges	Avg. node deg.	Std. node deg.	Max. node deg.	Density
Ecoli[1]	Biology	1,805	29,320	16.24	48.38	1,030	1.8009%
Yeast[1]	Biology	2,375	23,386	9.85	15.5	118	0.8295%
Power[1]	Transport	4,941	13,188	2.67	1.79	19	0.1081%
PolBlogs[2]	Web	1,490	19,025	12.77	20.73	256	1.7150%
Router[1]	Web	5,022	12,516	2.49	5.29	106	0.0993%
Physics[3]	Collaboration	34,493	495,924	14.38	15.57	382	0.0834%
Pubmed[4]	Citation	19,717	88,648	4.5	7.43	171	0.0456%
Citeseer[4]	Citation	3,327	9,104	2.74	3.38	99	0.1645%
Twitch[5]	Social	34,118	429,113	12.58	35.88	1,489	0.0737%
Github[6]	Social	37,700	289,003	7.67	46.59	6,809	0.0407%

Table 4. Statistics of test datasets.

Dataset	Domain	# Nodes	# Edges	Avg. node deg.	Std. node deg.	Max. node deg.	Density
Celegans[1]	Biology	297	4,296	14.46	12.97	134	9.7734%
USAir[1]	Transport	332	4,252	12.81	20.13	139	7.7385%
PB[1]	Web	1,222	33,428	27.36	38.42	351	4.4808%
NS[1]	Collaboration	1,589	5,484	3.45	3.47	34	0.4347%
CS[3]	Collaboration	18,333	163,788	8.93	9.11	136	0.0975%
Cora[4]	Citation	2,708	10,556	3.9	5.23	168	0.2880%
Facebook[7]	Social	22,470	171,002	7.61	15.26	472	0.0677%

A. Time Complexity Analyses

The time complexity of our **TFMLinker** is $O(k\overline{m}_e d_1 + k\overline{m}_v d_1^2 + (2k+p)dd_2(2k+p+d+d_2))$, where k denotes the number of positive or negative context samples, p is the number of learnable prototypes, $\overline{m}_e$ is the average number of links in an ego-subgraph, $\overline{m}_v$ is the average number of nodes in an ego-subgraph, d is the dimensionality of the link representation produced by the GNN and heuristic methods, d_1 is the feature dimension of the GNN, and d_2 is the encoding dimension of the TFM. Specifically, extracting the $2k+1$ 1-hop ego-subgraphs (including context and target links) requires $O(k\overline{m}_e)$ operations, while the GNN encoding of these subgraphs costs $O(k\overline{m}_e d_1 + k\overline{m}_v d_1^2)$. Heuristic feature computation, which can be pre-computed in batch for the entire graph, incurs negligible overhead. Within the TFM, the feature-level attention mechanism contributes $O((2k+p)d^2 d_2)$, the sample-level attention mechanism contributes $O((2k+p)^2 dd_2)$, and the feed-forward network adds $O((2k+p)dd_2^2)$. Summing these components yields the final complexity expression. Notably, the computational cost outside the TFM remains consistent with typical GNN-based ego-subgraph approaches such as SEAL (Zhang & Chen, 2018). Although incorporating the TFM introduces additional overhead for in-context learning, it effectively eliminates the need for computationally expensive fine-tuning on downstream tasks.

B. Experimental Details

B.1. Datasets

The complete details of both the training and test graph datasets are summarized in Table 3 and 4. Following (Dong et al., 2025), these datasets are selected from diverse domains and exhibit varied graph statistics. They are intentionally chosen to ensure that **TFMLinker** encounters a broad spectrum of link prediction connectivity patterns. This diversity in training data is essential for enabling **TFMLinker** to adapt effectively to previously unseen graphs. Each test dataset is randomly divided into training, validation, and test subsets in a 70%:10%:20% ratio, and the metrics are reported on the test subsets.

B.2. Implementation Details

We employ a 3-layer GraphSAGE (Hamilton et al., 2017) with 128-dimensional hidden representations as the GNN and employ the LimiX-16M version as the default TFM. The heuristic-based link representations have a dimensionality of 16.

Table 5. The results of **TFMLinker** with different backbones. The best results are in **bold** and the runner-ups are underlined.

	Biology C.ele	Transport USAir	Web PB	Collaboration NS	Collaboration CS	Citation Cora	Social Facebook	Average
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
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
TFMLinker + TabPFNv2	67.86±6.59	**92.09**±1.18	47.91±2.23	86.19±1.43	**70.77**±1.56	61.39±1.85	**69.92**±1.57	70.88
TFMLinker + Limix-2M	65.66±5.31	91.15±1.46	47.03±3.03	87.77±1.82	70.13±1.87	**63.43**±1.60	69.80±1.17	70.71
TFMLinker + Limix-16M	**68.62**±5.97	91.67±1.37	47.75±2.76	**90.46**±2.43	70.42±1.91	62.63±1.77	69.06±1.12	**71.52**

The activation functions used are ReLU and GELU. The number of learnable prototypes is set to 80, and no dropout is applied. The model is trained for 5 epochs with a batch size of 16 and a learning rate of 3e-5. For each target link, we sample 40 positive and 40 negative links from the corresponding training dataset as in-context links. This design introduces variability: different target links from the same dataset, or the same target link across training batches, may be associated with different sets of in-context links. All experimental results are reported as mean ± standard deviation over 10 independent runs. The training phase of GFMs is guided by performance on a merged validation set comprising 200 links from the validation subset of each test dataset, while fine-tuning of the GNN relies solely on in-domain validation performance.

C. Additional Experiments

C.1. Impact of TFM Backbone

We further evaluate **TFMLinker** with various TFMs, including TabPFNv2, Limix-2M and Limix-16M, as presented in Table 5. For comparison, we utilize SEAL (under the Pre-train & Fine-tune setting) and UniLP as the base models, given their status as two of the best-performing baselines. The results demonstrate that **TFMLinker** consistently surpasses these strong baselines on 6 out of 7 benchmark datasets, showing superior overall performance across different TFMs. This further confirms the effectiveness of **TFMLinker**, irrespective of the specific TFM employed. Furthermore, as the TFM size increases from Limix-2M to TabPFNv2 and then to Limix-16M, the average performance metrics of **TFMLinker** show consistent improvement. This demonstrates that **TFMLinker** can continuously benefit from the enhanced capabilities of larger TFMs.

C.2. Hyper-parameter Sensitivity

We further evaluate the sensitivity of the number of learnable prototypes p, with UniLP employed as the base model. The results on USAir, Cora, Facebook, and average performance are shown in Figure 3. This parameter plays a crucial role in balancing the modeling scope: if set too high, the model may overemphasize common and transferable patterns across diverse tasks and graphs, whereas if set too low, it risks learning only patterns specific to a single task. While an appropriate choice of p can enhance performance, our method is not highly sensitive to this hyperparameter and consistently outperforms the best baselines across a wide range of values.

C.3. Running Time Analysis

To evaluate the computational overhead introduced by **TFMLinker**, we conduct a comparative analysis of average inference times between **TFMLinker** and the in-context learning method UniLP on the Celegans dataset. All key factors influencing runtime—including batch size and hardware configuration (H200 GPU on a Linux server with 192-core Intel Xeon Platinum 8558 processors)—are kept constant during testing. UniLP requires 10.94 seconds, while **TFMLinker** requires 12.02

[1] https://github.com/Barcavin/context_LP/tree/master/data
[2] https://netset.telecom-paris.fr/datasets/polblogs.tar.gz
[3] https://github.com/shchur/gnn-benchmark/raw/master/data/npz
[4] https://github.com/kimiyoung/planetoid/raw/master/data
[5] https://snap.stanford.edu/data/twitch.zip
[6] https://snap.stanford.edu/data/git_web_ml.zip
[7] https://snap.stanford.edu/data/facebook_large.zip

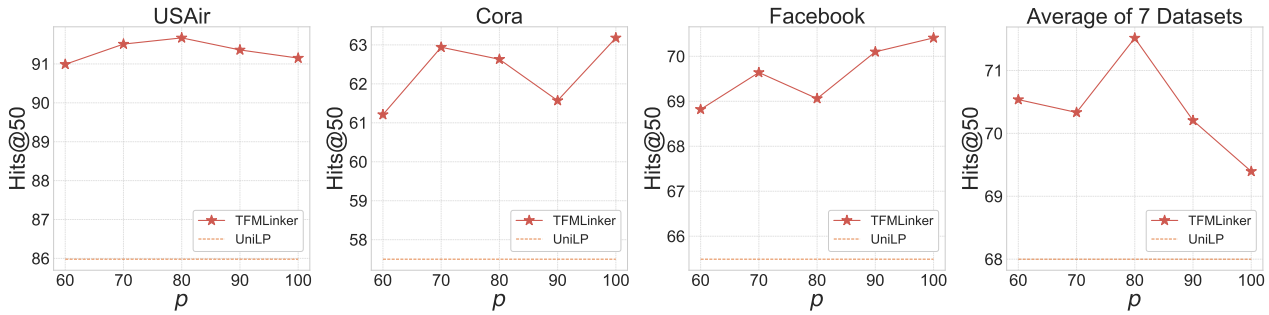

Figure 3. The impact of the number of learnable prototypes p.

Table 6. Performance comparison with competitive link prediction methods on Photo and OGBL-Citation2, evaluated by Hits@50 and MMR, respectively. The best results are in **bold** and the second-best results are underlined.

	Photo	OGBL-Citation2
SEAL	37.74	81.35
GAE	6.54	9.06
ELPH	35.15	20.36
NCNC	37.25	24.79
MPLP	40.19	47.13
UniLP	38.16	83.29
TFMLinker	**43.63**	**84.69**

seconds, demonstrating that **TFMLinker** maintains computational efficiency comparable to the existing in-context learning approach. The slight additional overhead of **TFMLinker** is attributed to the TFM, which has been demonstrated to deliver significant performance improvements. Compared to other baselines that require fine-tuning to adapt to downstream tasks, **TFMLinker** incurs significantly lower computational costs as it eliminates the need for additional training on specific datasets.

C.4. More Datasets

We further extended the evaluation to a product network (Photo (Shchur et al., 2018)) and the large-scale OGBL-Citation2 (Hu et al., 2020) graph, as summarized in Table 6. Except for UniLP, all baselines follow the Pre-train & Fine-tune setting. **TFMLinker** outperforms the baselines on both datasets, even though it has never been trained on the product domain or large-scale graphs. This further validates the model's generalization capability for link prediction.