Link Prediction via Adversarial Knowledge Distillation and Feature Aggregation

Wen Li
Jiangnan University
Xiaoning Song
x.song@jiangnan.edu.cn

Jiangnan University
Wenjie Zhang
Jiangnan University
Yang Hua
Jiangnan University
Xiaojun Wu
Jiangnan University

Research Article

Keywords: Link Prediction, Adversarial Knowledge Distillation, Knowledge Graph, Feature Aggregation

Posted Date: December 12th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4894235/v1

Additional Declarations: No competing interests reported.

Link Prediction via Adversarial Knowledge Distillation and Feature Aggregation

Wen Li[1], Xiaoning Song[1,2], Wenjie Zhang[1], Yang Hua[1], Xiaojun Wu[1]

[1*]School of Artificial Intelligence and Computer Science, jiangnan University, Wuxi, 214122, China.
[2]DiTu (Suzhou) Biotechnology Co., Ltd, 215000, Suzhou, China.

Contributing authors: 6223111046@stu.jiangnan.edu.cn; x.song@jiangnan.edu.cn; wenjie.zhang@stu.jiangnan.edu.cn; 7211905018@stu.jiangnan.edu.cn; wu_xiaojun@jiangnan.edu.cn;

Abstract

Graph neural networks (GNN) have shown strong performance in link prediction tasks. However, it is susceptible to higher latency due to the trivial correlation of data in its neighborhood, which poses a challenge for its practical application. In contrast, although Multi-layer Perceptron (MLP) performs poorly, it has a shorter inference time and is more flexible in practical applications. We utilize a distillation model to combine the powerful inference capabilities of GNN with the inference efficiency of MLP. Distillation models usually use a predefined distance function to quantify the differences between teacher-student networks, but this cannot be well applied to various complex scenarios. In addition, the limited node information severely affects the learning ability of MLP. Therefore, to cope with these problems. Firstly, we propose an Adversarial Generative Discriminator (AGD), which trains the discriminators and generators against each other to adaptively detect and reduce the differences. Secondly, we also propose the Feature Aggregation Module (FAM) to help the MLP obtain sufficient feature information before distillation starts. In the experiments, it is shown that our approach can achieve good results in link prediction tasks, outperforming the baseline model Linkless Prediction (LLP) and maintaining a good inference speed on eight datasets in two different settings.*

Keywords: Link Prediction, Adversarial Knowledge Distillation, Knowledge Graph,Feature Aggregation

*The code on https://github.com/lwuen/LPVAKD.git

1 Introduction

Nowadays, Graph Neural Network (GNN) has become a standard toolkit in the field of graphics research, including recommendation systems [1–3], social networks [4–6], and biomedical [7–9]. Besides, GNN exhibits strong expressive power on large-scale graphs that are often hyperparameterized. The advent of large-scale graphs benchmarking, including Microsoft Academic Graph (MAG) [10] and Open Graph Benchmark (OGB) [11], has led to the development of complex and hyper-stacked GNN models to achieve the most advanced performance. Previous studies [12–14] have demonstrated that deep and complex GNN models significantly outperform shallow models, indicating that hyperparameter GNN models possess strong expressive and generalization abilities. However, large-scale models may encounter parametric and time-efficiency issues when deployed to devices with limited computational and processing power (mobile phones, etc.). To address this challenge, we explore Knowledge Distillation (KD) techniques that have recently received close attention in the field of graph research.

In a variety of Knowledge Graph [15–17] and Natural Language Processing tasks [18–20], knowledge distillation is a powerful technique for compressing large models into small ones. In particular, distillation models, as described by Hinton [21], are highly effective in compressing complex neural networks. The fundamental concept is that student networks emulate the behavior of teacher networks to achieve competitive and even outstanding performance. Therefore we have conceptualized, discussed, and explored a distillation method for us to distil the knowledge information from the GNN into the MLP model.

The traditional distillation method [22–26] relies on other predefined distance functions to impart teacher behaviors to the student model through predefined rules, which largely limits the student model's ability to learn. As if the teacher is just repeating the textbook to the students according to the textbook, and the students can only learn a single piece of information from the textbook to learn a single piece of information, and cannot summarise it well. Therefore, this traditional distillation method is prone to two inherent limitations:

Firstly, they compel student networks to emulate teacher networks using handcrafted distance functions, where the optimal formula [27] is difficult to determine. Furthermore, [28, 29] have highlighted that students trained in this manner always perform suboptimally due to the difficulty in learning the exact distribution from the teacher.

Secondly, the limited information about the node features of the student model MLP makes it challenging to learn sufficient information about the node features and structure through the defined function distance.

1. To address the first problem, we draw on the ideas of He [30] and propose an adversarial knowledge refinement framework to address some of the aforementioned issues. In particular, we design an Adversarial Generative Discriminator (AGD) to help student models adversarially learn information from teacher models by the perspective of entity pairs. The information distribution generated by the generator (also known as the student network) after training is similar to that of the teacher

entity, therefore the AGD is unable to distinguish between them, by alternately updating the AGD and MLP generators, our model can transform the correlation of entity pairs from complex teacher GNN to compact student MLP. Furthermore, the topology-aware AGD proposed in this study has the potential to mitigate the risk of overfitting.

2. To address the second issue, we developed an MLP node feature aggregation module. This module enables each node in the MLP to obtain sufficient feature information. Each node can capture the feature information of neighboring nodes through the aggregator and subsequently assign weights according to their importance. We believe this is an effective solution to the problem of insufficient learning of feature and structural information by the student model, which occurs during the distillation process. This is also demonstrated in the subsequent experimental results.
3. In order to enhance the capture of crucial information during distillation, we integrate three distillation strategies that collectively offer a comprehensive approach. The initial strategy is distillation utilizing real label-assisted models, which effectively convey the authentic information in the links to the student model. However, since the real label-assisted distillation does not accurately reflect the relative importance of each link, we consider the distillation strategy of contribution distribution. The information distilled about the links around a node is used for the distillation decision regarding the node's importance. Finally, the adversarial generation strategy, which is the discriminative method proposed in the first point, employs the method of denying the student MLP and affirming the GNN, thereby enabling the MLP to learn as much knowledge information from the GNN as possible. The experimental results demonstrate that the combination of these three strategies can assist the model in achieving satisfactory refinement results.

In order to verify the effectiveness of our proposed method, we used a total of eight link prediction standard datasets and arranged two experimental settings: a common standard Transductive setting and a production setting (which simulates the situation of real-life scenarios). From the final experiments, it can also be concluded that our two proposed innovative modules perform very well in the link prediction task. Our model outperforms the standalone MLP by an average of 21.58% percentage points over all datasets and by an average of 16.14% percentage points over the standalone MLP over the productive dataset. In addition, our model outperforms baseline on all datasets. Finally, our model is substantially faster than GNN in terms of inference speed and as fast as MLP and baseline.

2 Related work

2.1 Definition

Following, we introduce the notation related to knowledge graphs, we define $G = (V, E)$ to denote an undirected graph, where the undirected graph contains the node and edge information in the dataset, where V denotes the desired node in the graph, and E denotes the desired edge, where $E \in (V, V)$ denotes that E is a link between two nodes. Define $a_{ij} \in (0, 1)$ to denote the existence of an edge between two nodes, where 1 means that there is a link between the two nodes and 0 means that there is no link, and i and j denote the ordinal numbers of the nodes. The overall information of a node is denoted by $X \in \mathbb{R}^{N \times F}$, where N represents the number of nodes and F represents the feature dimension of a node. In addition, we define $H \in \mathbb{R}^{N \times D}$ to denote the node information in the model training, where D represents the dimension of the hidden layer. Finally, the prediction results of GNN and MLP are denoted by Y_{ij} and y_{ij}, respectively. Translated with DeepL.com (free version).

2.2 Link Prediction with GNN and MLP

GNN typically utilizes both A and X as inputs. However, MLP typically employs only X as input, where A is typically understood to represent the link information of a node and X is typically understood to represent the feature information. In our work, we have borrowed some ideas for linking predictive task models [31–36], where teacher GNN is used for pre-training to learn node feature representations and student MLP is used to predict the probability of link presence through distillation learning. The majority of GNN employs a message-passing method for processing node information. This framework enables the model to obtain the neighborhood information corresponding to the L-hop of each node, denoted by H_i, through the number of network L-layers. This process serves to update the feature information of each node H_i. Using an L-layer similar to the convolution operation, the final output is $Z \in \mathbb{R}^{N \times d'}$, where Z is the output obtained by the node after passing through the GNN and d' is the output dimension. For instance, the output of the L-layer of a Graph Convolutional Neural Network (GCN) [37] is represented as:

$$\begin{aligned} H^{(l)} =& \sigma(\hat{A} H^{(l-1)} W^{(l-1)}) \\ H^{(0)} =& X \end{aligned} \tag{1}$$

The normalized adjacency matrix $\hat{A} = \tilde{D}^{-\frac{1}{2}} \hat{A} \tilde{D}^{-\frac{1}{2}}$, $\tilde{A} = A + I$, A is the adjacency matrix, I is the unitary matrix (with the addition of its own node identity), and $\hat{D}$ is the diagonal matrix. $W^{(l)}$ is the learnable weight matrix of the l^{th} layer. After L-layer GCN processing, the output node is represented as $Z = H^{(L)}$. The σ is a nonlinear function (e.g., ReLU). In the MLP decoder section, for any target node $T = \{u, v\}$, the probability of the existence of links between them can be calculated:

$$\begin{aligned} q_{ij} &= z_{v_i} \otimes z_{v_j} \\ p(i, j) &= \Omega(q_{ij}) \end{aligned} \tag{2}$$

where z_{v_i}, z_{v_j} are the representations of the nodes u and v after GNN learning, and $\otimes$ denotes the feature representations of the node pairs that are aggregated using the product operation. Then, the link probability $p(u, v)$ of the target is given by $\Omega(q_{uv})$, where Ω is the learned predictive model (e.g., MLP) that converts the relationship between u, v into the link probability between pairs of entities.

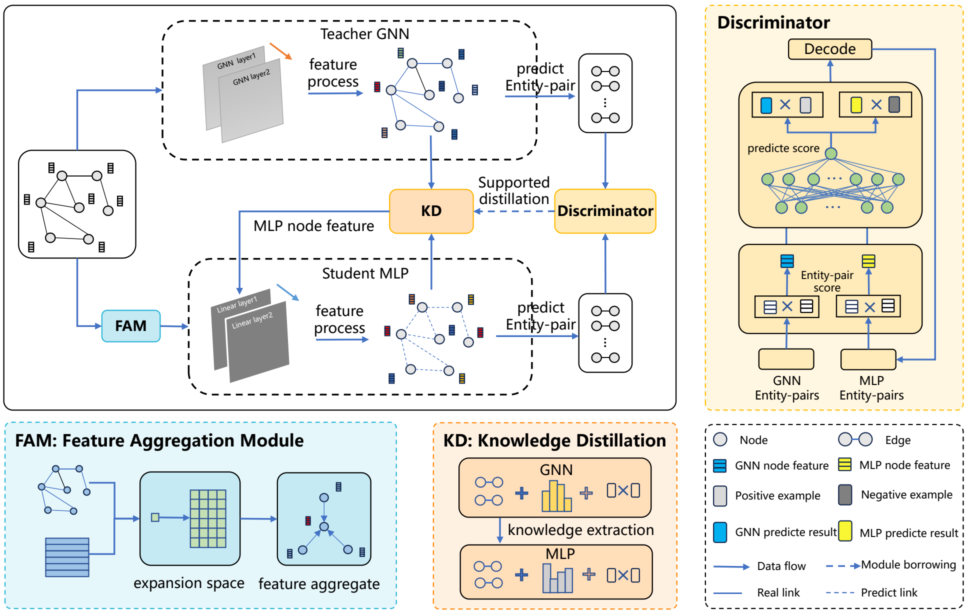

Fig. 1: The upper left quadrant depicts the overarching architectural framework of our model, encompassing GNN, MLP, adversarial generative discriminator module, feature aggregation module, and knowledge distillation module. The yellow section in the upper right quadrant represents the comprehensive structure of the discriminator module, which functions as an antagonistic generation module and undergoes updates as the model undergoes training. The blue section in the lower left quadrant represents our feature aggregation module, which facilitates enhanced feature information acquisition for MLP. The orange section represents the knowledge distillation algorithm module, which facilitates the student MLP in acquiring more pertinent information. The lower right-hand corner contains a legend.

2.3 Knowledge Distillation with GNN and MLP

The purpose of KD is to transfer from complex and low-speed models to smaller models that are more amenable to realistic deployment. As shown by Ghani [38], this is a model compression method. In link prediction tasks, knowledge usually refers to node features and graph structure information. Due to the slow neighbor aggregation caused by the strong data dependency in existing GNN methods, graph-based distillation methods usually extract GNN into MLP. This way is often used in a variety of model deployments because it can significantly improve model efficiency and have good scalability. For example, Zheng [39] proposed a KD-based framework to compress GNN.

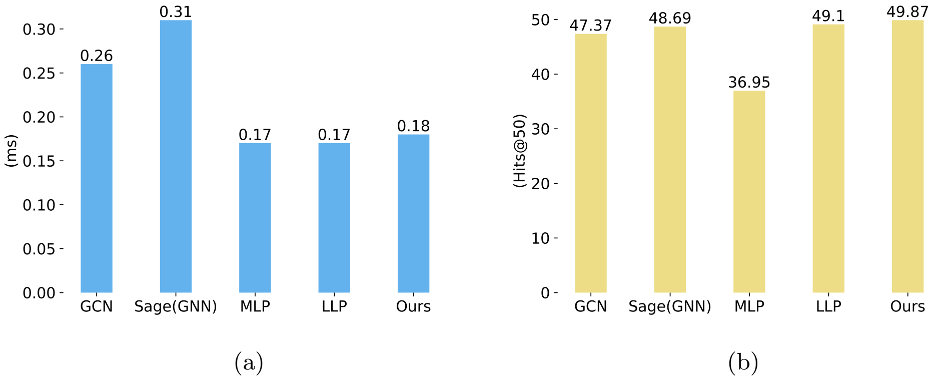

Fig. 2: Inference time and prediction performance comparison on Collab datasets , the figure (a) is the inference time result, and the figure (b) is the prediction performance result.

3 Model Methods

The model is divided into five modules: Teacher GNN, Student MLP, Adversarial generative discriminator Module, Knowledge Distillation Module, and Feature Aggregation Module. Initially, a GNN model is pre-trained and node information aggregation processing is performed on the FAM. Subsequently, the pre-trained GNN model is employed as the teacher model to guide student MLP to train. The AGD and KD were designed to enable the model to learn as much structural and node information as possible from the teacher GNN. The two modules work together so that our student MLP model can better learn sufficient information and knowledge. The specific model structure is shown in Figure 1.

3.1 Feature Aggregation Module

The majority of existing GNN refinement models have demonstrated satisfactory outcomes; however, there are some limitations associated with node-level feature alignment within the refinement process. The conventional approach to node feature information alignment utilizes representation matching, thereby enabling the MLP to learn more node feature information from the GNN. Nevertheless, during the experimental phase, although the student MLP acquired a certain degree of feature information, a considerable number of missing features resulted in a lack of sufficient node information within the student MLP. Consequently, this model incorporates a more efficient feature aggregation module to process the node feature information required by the student MLP. In contrast to the aggregation approach employed in GNN, this module employs the space-for-time method and the operation of scatter summing by index, which enables the efficient aggregation of feature information. Finally, the aggregated node information is utilized as the node feature input for the student MLP. Following the MLP aggregation operation, it is evident during the distillation process that the student MLP can more effectively learn the teacher's node feature representation, as well as enhance its ability to learn graph topology information.

$$\begin{aligned} h_i^k =& h_i^{k-1} + Scatter\{(h_j^{k-1} | e_{i,j} \in \varepsilon), \\ & unsqueeze(h_j^{k-1} \in R^{N \times F})\} \end{aligned} \tag{3}$$

Where *unsqueeze* is to create a blank array with dimensions $N \times F$, *Scatter* is to aggregate the local neighborhood features, and $(h_j^{k-1} | e_{i,j} \in \varepsilon)$ is to index to the node links of neighboring nodes, h_i^{k-1} is to get the node feature information of the previous hop, and h_i^k is to aggregate the node features after aggregating the k-hop neighborhood information, resulting in the final node features with more information.

3.2 Adversarial Generative Discriminator

Traditional distillation methods replicate the teacher network by compelling the student network to utilize a manually created distance function. Besides, the optimal formula is challenging to ascertain, we devise a discriminator to assist the student MLP in acquiring node pair information from the teacher GNN in an adversarial manner. To distinguish between the teacher node pair matching and the student node pair matching, we employ the adversarial generative approach in link prediction. The generator (student MLP) acts as a spoofing discriminator in adversarial learning. The aim was to identify discrepancies between MLP and GNN, thereby enabling the student MLP to learn as much new feature information as possible from the teacher GNN. As the objective is to analyze the link layer rather than the node layer, it is necessary to compare the link probabilities between two nodes. In the end, entity pairing is employed to train the discriminator. The discriminator is divided into two distinct sections: the self-training section and the auxiliary distillation model section.

In the self-training phase, two sets of information must be obtained: the GNN-trained entity pairs and the MLP-generated entity pairs. These two sets of information must be updated continuously throughout the refining process. Furthermore, it is essential to ensure that the information of the two sets of entity pairs is consistent in terms of dimensionality. The specific definitions and processes during the training of the discriminator are as follows: *Real* denotes 1, *Fake* denotes 0, and $D(V_{i,j}^T)$ denotes the teacher GNN positive sample node pair information, $D(V_{i,j}^S)$ denotes the negative sample node pair information generated by the student MLP. Here, the input dimensions are all $N \times F$.

$$\begin{aligned} Max_N \frac{1}{|V|} \sum_{v_{i,j} \in V} & (\log P(Real | N(v_{i,j}^T)) \\ & + \log P(Fake | N(v_{i,j}^S))) \end{aligned} \tag{4}$$

In the auxiliary distillation model phase, the discriminator is employed to facilitate distillation model training. It is postulated that the discriminator trained in an alternating manner possesses a superior filtering function, which can facilitate the student MLP model in acquiring more comprehensive node and structure information from the GNN.

3.3 Knowledge Distillation Module

3.3.1 Entity-Pair Matching

Entity pair matching is a strategy that employs trained adversarial generative discriminators to assist distillation. During model training, the discriminators are continuously updated with training, and the student MLP node features are constantly changing. This promotes the student MLP to continuously learn new structural feature information from the teacher's GNN. This methodology is more effective in developing student MLP generalization abilities. In the distillation process, the loss method is as follows. To handle the v_i and v_j student node entity pairs, the BCE is a cross-entropy loss function.

$$\mathcal{L}_{AD} = BCE(Discriminator(v_i, v_j), 1) \tag{5}$$

3.3.2 Logical Matching

Logic matching is a strategy for extracting knowledge directly from the GNN to the MLP. It aims to impart to the MLP better generalization skills as the teacher model, while still allowing for good summarisation in downstream tasks. It was proposed by [21] and is a very effective method of knowledge distillation that can work well in a variety of distillation models. In addition to theoretical proof of its effectiveness, empirical evidence also demonstrates the efficiency of this approach in the knowledge transfer of graph data. In many recent studies [40–43], soft logs generated by various teacher GNN have been used to train student MLP models, resulting in satisfactory performance in node tasks and link prediction tasks.

$$\mathcal{L}_{sup} = BCE(\hat{y}_{i,j}, a_{i,j}) \tag{6}$$

3.3.3 Distribution-Based Matching

Distribution-based matching is also a strategy used to aid model distillation. In logical matching, the matching strategy has the same learning ability for all links, which precludes the possibility of reflecting the relative importance of different links. Therefore, distribution-based matching plays an important role. Its main role is to rank the contribution to the presence of links with nodes centered on each node, which allows some more important link information to be passed on to the student model. The Kullback-Leibler divergence between the GNN prediction $y_{v,i}$ and the MLP prediction $\hat{y}_{v,i}$ centered on each node v is used. This is defined as follows:

$$\begin{aligned}\mathcal{L}_{DB} = &\sum_{v \in \mathcal{V}} \sum_{i \in C_v} \frac{exp(y_{v,i}/\tau)}{\sum_{j \in C_v} exp(y_{i,j}/\tau)} \\ &\log \frac{exp(\hat{y}_{v,i}/\tau)}{\sum_{j \in C_v} exp(\hat{y}_{v,j}/\tau)}\end{aligned} \tag{7}$$

where τ is a temperature hyperparameter that controls the normalized soft distribution. We match the values of the probability distribution of entity-pair links centered on each node.

3.3.4 Loss

When training our model, we combine three types of losses to better assist in model distillation, including discriminative loss, logical matching loss, and distribution-based matching loss. As a result, our overall loss for MLP is as follows:

$$L = \alpha \mathcal{L}_{AD} + \beta \mathcal{L}_{sup} + \gamma \mathcal{L}_{DB} \tag{8}$$

where α, β, and γ are hyper-parameters, which can be adjusted according to the actual situation in practical applications, and can balance the impact of each loss on the model.

4 Experiments

4.1 Experimental Dataset

We used eight common benchmark datasets in the link prediction task (see Table 1 for specific information), including datasets such as Cora. We set Hits@20,Hits50 and AUC (Area Under Curve) three evaluation metrics for straight talk in datas. We selected the same evaluation metrics as baseline to show the effect for better comparison with Baseline. For the OGB dataset collab, due to the small number of node features in the collab dataset, it will be more difficult in the prediction, and also for a better comparison with baseline, so we use its official metrics after public ranking on the collab dataset ($Hits$@50). For the other datasets, we use $Hits$@20 as the main evaluation metric, and $Hits$@20 is also one of the main evaluation metrics for no-OGB data. For all experiments on the 8 datasets, we used a random initialisation condition and averaged the test performance over 10 runs for each.

Table 1: Datasets Information.

Dataset	Nodes	Features	Edges
Cora	2708	1433	5278
Citeseer	3327	3703	4552
Pubmed	19717	500	44324
CS	18333	6805	163788
physics	34493	8415	495924
Computers	13752	767	491722
Photos	7650	745	238162
Collab	235868	128	1285465

Table 2: Link prediction performance under transductive and production setting. For Collab, we report Hits@50. For other datasets, we report Hits@20. Best performances are marked with bold.

	Datasets	GNN	MLP	LLP(baseline)	Ours	$\triangle_{\mathbf{MLP}}$	$\triangle_{\mathbf{LLP(baseline)}}$
Transductive	**Cora**	$74.38_{\pm 1.54}$	$78.06_{\pm 1.50}$	$78.82_{\pm 1.74}$	$\mathbf{79.28_{\pm 1.25}}$	+1.22	+0.46
	Citeseer	$73.89_{\pm 0.95}$	$71.21_{\pm 3.22}$	$77.32_{\pm 2.42}$	$\mathbf{77.74_{\pm 1.81}}$	+6.53	+0.42
	Pubmed	$51.98_{\pm 5.25}$	$42.89_{\pm 1.67}$	$57.33_{\pm 2.42}$	$\mathbf{61.88_{\pm 1.30}}$	+18.99	+4.55
	Cs	$59.51_{\pm 7.34}$	$34.01_{\pm 9.37}$	$68.62_{\pm 1.46}$	$\mathbf{74.23_{\pm 1.56}}$	+40.22	+5.61
	Physics	$66.74_{\pm 1.53}$	$31.26_{\pm 9.12}$	$72.01_{\pm 1.89}$	$\mathbf{76.20_{\pm 2.16}}$	+44.94	+4.19
	Computer	$31.66_{\pm 3.08}$	$20.19_{\pm 1.58}$	$35.32_{\pm 2.28}$	$\mathbf{40.29_{\pm 3.21}}$	+20.10	+4.97
	Photos	$51.50_{\pm 4.48}$	$27.83_{\pm 4.90}$	$49.32_{\pm 2.64}$	$\mathbf{55.51_{\pm 2.76}}$	+27.68	+6.19
	Collab	$48.69_{\pm 0.87}$	$36.95_{\pm 1.37}$	$49.10_{\pm 0.57}$	$\mathbf{49.87_{\pm 1.07}}$	+12.92	+0.77
Production	**Cora**	$27.80_{\pm 2.11}$	$22.90_{\pm 2.22}$	$27.87_{\pm 1.24}$	$\mathbf{38.76_{\pm 5.34}}$	+15.86	+10.89
	Citeseer	$38.78_{\pm 2.59}$	$31.21_{\pm 3.75}$	$34.70_{\pm 2.45}$	$\mathbf{43.95_{\pm 5.73}}$	+12.74	+9.25
	Pubmed	$52.71_{\pm 1.81}$	$38.01_{\pm 1.67}$	$53.48_{\pm 1.52}$	$\mathbf{55.89_{\pm 2.99}}$	+17.88	+2.09
	Cs	$60.69_{\pm 3.17}$	$38.15_{\pm 10.78}$	$60.74_{\pm 1.41}$	$\mathbf{60.94_{\pm 2.67}}$	+22.79	+0.24
	Physics	$\mathbf{55.82_{\pm 2.43}}$	$29.99_{\pm 1.96}$	$52.83_{\pm 1.50}$	$53.54_{\pm 3.36}$	+23.55	+0.71
	Computer	$\mathbf{34.38_{\pm 1.41}}$	$19.43_{\pm 0.82}$	$24.58_{\pm 3.33}$	$27.17_{\pm 4.09}$	+7.74	+2.59
	Photos	$\mathbf{51.03_{\pm 6.05}}$	$34.29_{\pm 2.49}$	$43.79_{\pm 1.27}$	$46.71_{\pm 7.12}$	+12.42	+2.92

Table 3: Comparison of experimental results for AUC (the probability that the score of a positive sample is greater than the score of that negative sample) in the Tranductive environment.

Datasets	GCN	GIC	SEAL	WalkPool	S3GRL	LLP(baseline)	Ours
Cora	$89.14_{\pm 1.20}$	$91.42_{\pm 1.24}$	$90.29_{\pm 1.89}$	$92.24_{\pm 0.65}$	$94.77_{\pm 0.68}$	$94.90_{\pm 0.53}$	$\mathbf{94.92_{\pm 0.21}}$
Citeseer	$87.89_{\pm 1.48}$	$92.99_{\pm 1.14}$	$88.12_{\pm 0.85}$	$89.97_{\pm 1.01}$	$95.76_{\pm 0.59}$	$95.42_{\pm 0.47}$	$\mathbf{95.87_{\pm 0.25}}$
Pubmed	$92.72_{\pm 0.64}$	$91.04_{\pm 0.61}$	$97.82_{\pm 0.28}$	$98.36_{\pm 0.11}$	$\mathbf{99.00_{\pm 0.08}}$	$97.26_{\pm 0.06}$	$98.39_{\pm 0.06}$

4.2 Evaluation Settings

In contrast to conventional GNN link prediction techniques, our approach involves training MLP with enhanced predictive capabilities through GNN, resulting in notable improvements in inference speed and parameter efficiency. While this approach may not rival the performance of some existing GNN models in terms of prediction accuracy, its enhanced efficiency is a notable advantage.To exemplify the predictive power of our model in different realistic scenarios, we conducted experiments in both Transductive and Production environments. In the transductive environment, all nodes(including training, validating, and testing) in the undirected graph will appear in the training part of the model, which draws on previous research [44–47]. We randomly select 5%-15% of entity node pairs from the graph, define them as valid sets and test sets, and block the link information related to these entity node pairs in the training phase. For the OGB dataset (collab), we directly used the officially provided segmentation data. In the production environment, the information about the nodes in the valid and

Table 4: Transductive and production ablation experiment results.

	Datasets	**GNN**	**MLP**	**LLP**	**LLP+FAM**	**LLP+FAM+AGD**
Transductive	**Cora**	74.38$_{\pm 1.54}$	78.06$_{\pm 1.50}$	78.82$_{\pm 1.74}$	78.86$_{\pm 1.19}$	**79.28**$_{\pm \mathbf{1.25}}$
	Citeseer	73.89$_{\pm 0.95}$	71.21$_{\pm 71.21}$	77.32$_{\pm 2.42}$	77.60$_{\pm 5.40}$	**77.74**$_{\pm \mathbf{1.81}}$
	Pubmed	51.98$_{\pm 5.25}$	42.89$_{\pm 1.67}$	57.33$_{\pm 2.42}$	59.91$_{\pm 2.12}$	**61.88**$_{\pm \mathbf{1.30}}$
Production	**Cora**	27.80$_{\pm 2.11}$	22.90$_{\pm 2.22}$	27.87$_{\pm 1.24}$	37.02$_{\pm 3.23}$	**38.76**$_{\pm \mathbf{5.34}}$
	Citeseer	38.78$_{\pm 2.59}$	31.21$_{\pm 3.75}$	34.70$_{\pm 2.45}$	37.30$_{\pm 4.47}$	**43.95**$_{\pm \mathbf{5.73}}$
	Pubmed	52.71$_{\pm 1.81}$	38.01$_{\pm 1.67}$	53.48$_{\pm 1.52}$	54.53$_{\pm 2.27}$	**55.89**$_{\pm \mathbf{2.99}}$

test is separated from the information about the training nodes in order to simulate a more realistic situation. We therefore perform a more detailed segmentation of the dataset. It is worth noting that we only performed the production setup experiments on the non-OGB dataset. Since the OGB dataset (collab) has already been temporally segmented in its publicly available dataset, we do not have a better way to perform the segmentation in this experimental setup for the time being.

4.3 Experimental Results

4.3.1 Transductive Setting:

In the transductive experimental environment, experiments were conducted on seven non-OGB data sets and one OGB data set. In Table 2, we show the link prediction performance of GNN, MLP, baseline, and our model methods in the transduction setup. Our proposed model outperforms MLP and baseline experimentally on the evaluation metric $Hits@20$ on either dataset. Specifically, our model demonstrated an improvement of 18.18% and 10.59% points over the MLP and baseline, respectively, on average across the entire dataset. In the physical dataset, our model exhibited an improvement of 44.94% and 4.19% points over MLP and baseline, respectively. The greatest improvements in our model over the baseline were observed in the CS and Photos datasets where they improved by 5.61% and 6.19% points, respectively. Furthermore, our model outperforms the teacher GNN and baseline models on eight datasets. This indicates that our proposed MLP node aggregation module and AGD can be more effective in helping student MLP to better learn more information from GNN, which can effectively improve the prediction of connective links in MLP. We can observe that on some of the datasets, our model has a more significant improvement compared to GNN, which may be mainly due to two reasons: The first reason is that the student MLP obtains enough node features that its prediction performance can be improved well. For example, on the Photos dataset, our model has a great improvement over baseline, which is not as effective as GNN in the same situation, so the model has a better prediction ability when MLP gets enough features. The second reason is due to the adoption of the KD framework, where MLP can obtain information about the structure of relationships in GNN through KD, which largely helps to improve the generalization ability of the MLP. The recent study [30, 44] on knowledge distillation also found a similar conclusion that the distillation strategy can effectively improve the generalization ability of the model and help the MLP to better perform the prediction task. We have compared our model with other models of the same magnitude in the Transductive environment using the AUC metrics, and our model also has great advantages, as shown in table 3. We compare our models with 6 baselines belonging to four different categories of link prediction models. For message-passing GNNs (MPGNNs), we select GCN [37]. The autoencoder (AE) method is Graph InfoClust (GIC) [48]. Besides, for the SGRLs (Subgraph Representation Learning) includes SEAL [49], WalkPool [50] and S3GRL [51]. Finally, there is the LLP [44] method that uses a distillation strategy. Compared to other models, our approach has good results on the vast majority of data sets

4.3.2 Production Setting:

In the production experimental environment. In the table 2, we show the link prediction performance of GNN, MLP, baseline, and our model methods. In this environment setting, the OGB dataset has already been time-segmented in its publicly available datasets. Therefore, we only conducted experimental designs on seven non-OGB datasets. From the table, we observe that our model maintains a huge advantage over MLP and baseline in all benchmark tests. Specifically, our model improves 16.14% and 4.13% points on $Hits@20$ over the MLP and baseline methods, respectively. Our model outperforms the teacher GNN on four of the seven datasets, suggesting that our model can achieve effective link prediction in realistic deployments. We also observe that our model exhibits greater fluctuations across datasets, which is a recurring challenge in the process of refining a large model into a smaller one [39, 52], particularly in this prediction for unknown node information. In a production setting, the model can learn to generalize by learning existing node and structure information. However, it is still particularly difficult to predict unknown node information. In this case, for the unknown node features, there is a need for better methods to deal with them, which will largely help us in link prediction.

4.4 Testing Speeder

In Figure 2(a), we compare with other common GNN and baseline inference acceleration methods. The inference acceleration tests were performed on multiple datasets, using the collab dataset as the standard. Our model significantly improves the inference speedup compared to traditional GNN models, by 0.08ms on average. The inference speed is also kept at a very desirable level when compared to MLP, baseline, and both GNN.

In Figure 2(b), in the experiments with $Hits@50$ as the evaluation metric, our model improves on average by 1.84% compared to the two traditional GNN methods, and our model also improves more significantly compared to MLP and baseline, especially by 12.92% compared to MLP. Taken together, our proposed method and module can effectively improve the performance of the link prediction task with very satisfactory time efficiency.

4.5 Ablation Experiment

In the ablation experiments, we evaluated the efficacy of two new modules (the feature aggregation module and the adversarial generative discriminator) by adding them to the baseline model. In table 4, we compare the performance of the full model, standalone MLP, GNN, and baseline on the three datasets, from which it can be seen that the baseline model with the addition of the FAM module improves the results over the datasets in both the MLP and the baseline settings. However, the results in the Citeseer dataset in the production setting are not as effective as the independent GNN. Therefore, we conjecture that although MLP gets sufficiently adequate node information, enriching only the node information does not generalize the model well, and does not allow good judgment when targeting unknown node information. So by adding the AGD module, the results are better in the three datasets in both settings. Sets are further improved, especially in the Production setting, there is a more significant improvement, which reinforces our conjecture that AGD allows the student MLP to learn better generalization in distillation and show more power when it should be for unknown nodes. The results show that both the FAM and AGD contribute significantly to the model performance. In the transductive environment, the FAM module alone outperforms the instructor's GNN, MLP, and baseline. In the Production setting, both modules combined will outperform GNN, MLP, and baseline, when FAM and AGD complement each other, they both refine the relational knowledge more effectively and get a great result.

5 Conclusion

Our work mainly focuses on the related issues of using GNN for large-scale link prediction. To combine the advantages of MLP and GNN, The distillation modeling approach is employed to distill the information obtained from GNN into MLP. To resolve limited node information, we designed a feature aggregation module to enrich the student MLP nodes' features. Additionally, we designed an adversarial generative discriminator, to continuously correct the MLP so that it can have strong learning and generalization capabilities like the GNN. All experimental results also make known that our model improves in terms of inference speed, approaching the baseline and MLP (much faster than GNN). In both experimental environment settings, our model shows very good results on $Hits@20$,$Hits@50$ and AUC, with a very significant improvement over both MLP and baseline, which confirms the validity of our idea.

6 Limitations

Firstly, the distillation model is very dependent on the performance of the teacher's GNN, and the student model performs poorly when the teacher's GNN performs poorly.

Second, the predictive ability of the model in the less-node feature information dataset leaves much to be desired.

7 Acknowledgement

This work was supported in part by the National Key R&D Program of China (2023YFF1105102, 2023YFF1105105), the Major Project of the National Social Science Foundation of China (No. 21&ZD166), the National Natural Science Foundation of China (61876072) and the Natural Science Foundation of Jiangsu Province (No. BK20221535).

References

[1] Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: a survey. ACM Computing Surveys **55**(5), 1–37 (2022)

[2] Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., *et al.*: A survey of graph neural networks for recommender systems: Challenges, methods, and directions. ACM Transactions on Recommender Systems **1**(1), 1–51 (2023)

[3] Lakshmi, T.J., Bhavani, S.D.: Link prediction approach to recommender systems. Computing, 1–27 (2023)

[4] Zhong, T., Wang, T., Wang, J., Wu, J., Zhou, F.: Multiple-aspect attentional graph neural networks for online social network user localization. IEEE Access **8**, 95223–95234 (2020)

[5] Kumar, S., Mallik, A., Khetarpal, A., Panda, B.S.: Influence maximization in social networks using graph embedding and graph neural network. Information Sciences **607**, 1617–1636 (2022)

[6] Badiy, M., Amounas, F., El Allaoui, A., Bayane, Y.: Neural network for link prediction in social network. In: Farhaoui, Y., Hussain, A., Saba, T., Taherdoost, H., Verma, A. (eds.) Artificial Intelligence, Data Science and Applications, pp. 58–63. Springer, Cham (2024)

[7] Bove, P., Micheli, A., Milazzo, P., Podda, M., *et al.*: Prediction of dynamical properties of biochemical pathways with graph neural networks. In: Bioinformatics, pp. 32–43 (2020)

[8] Li, Y., Zhang, G., Wang, P., Yu, Z.-G., Huang, G.: Graph neural networks in biomedical data: A review. Current Bioinformatics **17**(6), 483–492 (2022)

[9] Zhang, X.-M., Liang, L., Liu, L., Tang, M.-J.: Graph neural networks and their current applications in bioinformatics. Frontiers in genetics **12**, 690049 (2021)

[10] Wang, K., Shen, Z., Huang, C., Wu, C.-H., Dong, Y., Kanakia, A.: Microsoft academic graph: When experts are not enough. Quantitative Science Studies **1**(1), 396–413 (2020)

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

[12] Zhang, W., Sheng, Z., Jiang, Y., Xia, Y., Gao, J., Yang, Z., Cui, B.: Evaluating deep graph neural networks. arXiv preprint arXiv:2108.00955 (2021)

[13] Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J.S.: Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE **105**(12), 2295–2329 (2017)

[14] Gallicchio, C., Micheli, A.: Fast and deep graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3898–3905 (2020)

[15] Wang, K., Liu, Y., Ma, Q., Sheng, Q.Z.: Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings. In: Proceedings of the Web Conference 2021, pp. 1716–1726 (2021)

[16] Liu, J., Wang, P., Shang, Z., Wu, C.: Iterde: an iterative knowledge distillation framework for knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 4488–4496 (2023)

[17] Tu, K., Cui, P., Wang, D., Zhang, Z., Zhou, J., Qi, Y., Zhu, W.: Conditional graph attention networks for distilling and refining knowledge graphs in recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1834–1843 (2021)

[18] Hahn, S., Choi, H.: Self-knowledge distillation in natural language processing. arXiv preprint arXiv:1908.01851 (2019)

[19] Yang, Z., Cui, Y., Chen, Z., Che, W., Liu, T., Wang, S., Hu, G.: Textbrewer: An open-source knowledge distillation toolkit for natural language processing. arXiv preprint arXiv:2002.12620 (2020)

[20] Fu, H., Zhou, S., Yang, Q., Tang, J., Liu, G., Liu, K., Li, X.: Lrc-bert: latent-representation contrastive knowledge distillation for natural language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 12830–12838 (2021)

[21] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)

[22] Gu, Y., Dong, L., Wei, F., Huang, M.: Minillm: Knowledge distillation of large language models. In: The Twelfth International Conference on Learning Representations (2023)

[23] Agarwal, R., Vieillard, N., Stanczyk, P., Ramos, S., Geist, M., Bachem, O.: Gkd:

Generalized knowledge distillation for auto-regressive sequence models. arXiv preprint arXiv:2306.13649 (2023)

[24] Li, S., Chen, J., Shen, Y., Chen, Z., Zhang, X., Li, Z., Wang, H., Qian, J., Peng, B., Mao, Y., et al.: Explanations from large language models make small reasoners better. arXiv preprint arXiv:2210.06726 (2022)

[25] Ho, N., Schmid, L., Yun, S.-Y.: Large language models are reasoning teachers. arXiv preprint arXiv:2212.10071 (2022)

[26] Fu, Y., Peng, H., Ou, L., Sabharwal, A., Khot, T.: Specializing smaller language models towards multi-step reasoning. In: International Conference on Machine Learning, pp. 10421–10430 (2023). PMLR

[27] Wang, Y., Xu, C., Xu, C., Tao, D.: Adversarial learning of portable student networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[28] Wang, X., Zhang, R., Sun, Y., Qi, J.: Kdgan: Knowledge distillation with generative adversarial networks. Advances in neural information processing systems **31** (2018)

[29] Wang, X., Zhang, R., Sun, Y., Qi, J.: Adversarial distillation for learning with privileged provisions. IEEE transactions on pattern analysis and machine intelligence **43**(3), 786–797 (2019)

[30] He, H., Wang, J., Zhang, Z., Wu, F.: Compressing deep graph neural networks via adversarial knowledge distillation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 534–544 (2022)

[31] Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)

[32] Berg, R.v.d., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017)

[33] Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, pp. 593–607 (2018). Springer

[34] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)

[35] Davidson, T.R., Falorsi, L., De Cao, N., Kipf, T., Tomczak, J.M.: Hyperspherical

variational auto-encoders. arXiv preprint arXiv:1804.00891 (2018)

[36] Zhu, Z., Zhang, Z., Xhonneux, L.-P., Tang, J.: Neural bellman-ford networks: A general graph neural network framework for link prediction. Advances in Neural Information Processing Systems **34**, 29476–29490 (2021)

[37] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

[38] Ghani, R., Senator, T.E., Bradley, P., Parekh, R., He, J.: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, ??? (2013)

[39] Zheng, W., Huang, E.W., Rao, N., Katariya, S., Wang, Z., Subbian, K.: Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. arXiv preprint arXiv:2111.04840 (2021)

[40] Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11953–11962 (2022)

[41] Chen, D., Mei, J.-P., Zhang, Y., Wang, C., Wang, Z., Feng, Y., Chen, C.: Cross-layer distillation with semantic calibration. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 7028–7036 (2021)

[42] Huang, T., Zhang, Y., Zheng, M., You, S., Wang, F., Qian, C., Xu, C.: Knowledge diffusion for distillation. Advances in Neural Information Processing Systems **36** (2024)

[43] Huo, C., Jin, D., Li, Y., He, D., Yang, Y.-B., Wu, L.: T2-gnn: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 4339–4346 (2023)

[44] Guo, Z., Shiao, W., Zhang, S., Liu, Y., Chawla, N.V., Shah, N., Zhao, T.: Linkless link prediction via relational distillation. In: International Conference on Machine Learning, pp. 12012–12033 (2023). PMLR

[45] Zhang, M., Chen, Y.: Link prediction based on graph neural networks. Advances in neural information processing systems **31** (2018)

[46] Chami, I., Ying, Z., Ré, C., Leskovec, J.: Hyperbolic graph convolutional neural networks. Advances in neural information processing systems **32** (2019)

[47] Cai, L., Li, J., Wang, J., Ji, S.: Line graph neural networks for link prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(9), 5103–5113 (2021)

[48] Mavromatis, C., Karypis, G.: Graph infoclust: Maximizing coarse-grain mutual information in graphs. In: Karlapalem, K., Cheng, H., Ramakrishnan, N., Agrawal, R.K., Reddy, P.K., Srivastava, J., Chakraborty, T. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 541–553. Springer, Cham (2021)

[49] Zhang, M., Chen, Y.: Link prediction based on graph neural networks. Advances in neural information processing systems **31** (2018)

[50] Pan, L., Shi, C., Dokmanić, I.: Neural Link Prediction with Walk Pooling (2022). https://arxiv.org/abs/2110.04375

[51] Louis, P., Jacob, S.A., Salehi-Abari, A.: Simplifying subgraph representation learning for scalable link prediction. arXiv preprint arXiv:2301.12562 (2023)

[52] Zhang, S., Liu, Y., Sun, Y., Shah, N.: Graph-less neural networks: Teaching old mlps new tricks via distillation. arXiv preprint arXiv:2110.08727 (2021)