T2-GNN: Graph Neural Networks for Graphs with Incomplete Features and Structure via Teacher-Student Distillation

Cuiying Huo[1], Di Jin[1,2], Yawen Li[3], Dongxiao He[1,*], Yu-Bin Yang[2], Lingfei Wu[4]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, P.R. China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, P.R. China
[3]School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing, P.R. China
[4]Pinterest, New York, USA
{huocuiying, jindi, hedongxiao}@tju.edu.cn, warmly0716@bupt.edu.cn, yangyubin@nju.edu.cn, lwu@email.wm.edu

Abstract

Graph Neural Networks (GNNs) have been a prevailing technique for tackling various analysis tasks on graph data. A key premise for the remarkable performance of GNNs relies on complete and trustworthy initial graph descriptions (i.e., node features and graph structure), which is often not satisfied since real-world graphs are often incomplete due to various unavoidable factors. In particular, GNNs face greater challenges when both node features and graph structure are incomplete at the same time. The existing methods either focus on feature completion or structure completion. They usually rely on the matching relationship between features and structure, or employ joint learning of node representation and feature (or structure) completion in the hope of achieving mutual benefit. However, recent studies confirm that the mutual interference between features and structure leads to the degradation of GNN performance. When both features and structure are incomplete, the mismatch between features and structure caused by the missing randomness exacerbates the interference between the two, which may trigger incorrect completions that negatively affect node representation. To this end, in this paper we propose a general GNN framework based on teacher-student distillation to improve the performance of GNNs on incomplete graphs, namely T2-GNN. To avoid the interference between features and structure, we separately design feature-level and structure-level teacher models to provide targeted guidance for student model (base GNNs, such as GCN) through distillation. Then we design two personalized methods to obtain well-trained feature and structure teachers. To ensure that the knowledge of the teacher model is comprehensively and effectively distilled to the student model, we further propose a dual distillation mode to enable the student to acquire as much expert knowledge as possible. Extensive experiments on eight benchmark datasets demonstrate the effectiveness and robustness of the new framework on graphs with incomplete features and structure.

Introduction

Graph Neural Networks (GNNs), as a powerful technique for modeling and analysis of graphs, have gained widespread attention in recent years as the graph-structured

---

data become ubiquitous (Wu et al. 2022). GNNs achieve remarkable results across a wide range of graph analysis tasks such as node classification (Wang et al. 2020; Yu et al. 2021), link prediction (You, Ying, and Leskovec 2019; Chen et al. 2020), and recommendation (Xu et al. 2019; Liu et al. 2020), and are continuously being applied in many fields such as computer vision and natural language processing (Wu et al. 2021; Zhou et al. 2020; Jin et al. 2021b). The essence of most existing GNNs is the process of message propagation and aggregation guided by graph structure (neighbors), such as GCN (Kipf and Welling 2017), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GAT (Velickovic et al. 2018) and APPNP (Klicpera, Bojchevski, and Günnemann 2019). Therefore, a key prerequisite for the success of GNNs is complete and trustworthy initial graph descriptions (i.e., node features and graph structure).

However, in real-world scenarios, there are many graph data with incomplete descriptions for various reasons. For example, some features or edges are missing due to technical or human errors during data collection. The large scale and potential resource constraints of complex real-world systems make collecting complete data prohibitively expensive or impossible. And we cannot obtain some sensitive feature information due to confidential data or privacy security (e.g., nationality, age). Furthermore, the addition of a new node leads to the absence of links between the node and other nodes in the graph, and even becomes an isolated node. All these cases lead to incomplete features or structure in the initial graph, which brings great challenges to GNNs.

Based on the types of missing descriptions in the graphs, descriptions missing graphs can be classified into three categories: 1) feature-incomplete graph where features of some nodes are partially (or entirely) missing, as shown in Fig. 1(a); 2) structure-incomplete graph where edges of some nodes are partially (or entirely) missing (Fig. 1(b)); 3) incomplete graph where features and (or) edges of some nodes are partially (or entirely) missing (Fig. 1(c)). For GNN-based methods for feature-incomplete graphs, some specific methods have been proposed. SAT (Chen et al. 2022) proposes a structure-feature matching GNN model to complete the missing features by assuming that the graph structure and node features come from the same latent space. GCNMF (Taguchi, Liu, and Murata 2021) integrates the
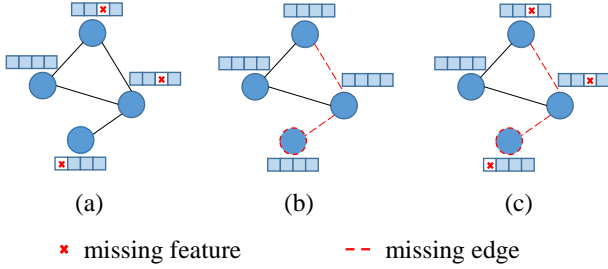
Figure 1: Three types of incomplete data. (a) Feature-incomplete graph. (b) Structure-incomplete graph. (c) Incomplete graph.

completion of missing features and graph learning within the same neural network architecture by representing the missing data with a Gaussian mixture model. HGNN-AC (Jin et al. 2021a) uses graph structure information as a priori guidance and proposes an end-to-end attribute completion framework for heterogeneous graphs. There are also many attempts to design GNN algorithms to model the structure-incomplete graphs by jointly learning optimized graph structure and node representations (Chen, Wu, and Zaki 2020; Zhu et al. 2022, 2021). But to our best knowledge, no method has been proposed for incomplete graphs, which is more challenging and relevant to many real-world applications.

In addition, these existing feature (or structure) completion methods usually rely on the matching relationship between features and structure to guide feature (or structure) completion, or employ the joint learning of node representation and information completion in the hope of achieving reciprocity between the two. However, recent studies have confirmed that the mutual interference between features and structure in graphs is an important reason for the performance degradation of GNNs (Yang et al. 2022; Wang et al. 2020). Due to the randomness of missing, features and structure usually cannot be completely matched in the graph, which makes features and structure interfere with each other in the message passing of GNN. In incomplete graphs, both features and structure are incomplete, which may make the mutual interference between the two more serious. Therefore, the methods based on the above mechanisms cannot be effectively applied to incomplete graphs, and even aggravate the propagation of error information, resulting in completed information having a negative impact on node representation.

To this end, in this paper, we introduce knowledge distillation to address the above problems, and propose a new general GNN framework (T2-GNN) for incomplete graphs. The new T2-GNN framework addresses three key limitations: (i) How to improve the performance of the base GNN (e.g., GCN) model in the case of incomplete features and graph structure, and efficiently avoid potential negative interference between features and structure? (ii) Given incomplete node features and graph structure, how can we effectively model them in favor of learning better node representations? (iii) How to distill the knowledge of the teacher model into

the student model to achieve effective knowledge transfer?

For the first limitation, we design teacher models based on node features and graph structure respectively to avoid potential interference between the two. Then, based on knowledge distillation, the well-trained teacher models are used to provide feature-level and structure-level targeted guidance for the message passing process of the student model (base GNN) to improve the performance of the student model on incomplete graphs. For the second limitation, in the feature teacher, we design parameterized imputation for feature completion, and adopt joint learning to capture effective information at the feature level. In the structure teacher, we construct a structure-enhanced graph in a personalized way to capture effective information at the structure level. Finally, for the third limitation, we propose a dual distillation mode to achieve comprehensive guidance from the teacher model to the student model.

Note that, compared with traditional knowledge distillation, our aim is not to distill knowledge from a cumbersome teacher model into a lightweight student model so that the student can hold a similar performance as the teacher's. Instead, our aim is for the issues existing in the student model, the teacher models can provide targeted and effective guidance, i.e., suit the medicine to the illness. In short, we summarize the main contributions as follows:

- We investigate a novel problem that existing GNNs fail to perform well when both features and structure of the graph are incomplete. And we find that the mutual interference between features and structure makes it more challenging to effectively model incomplete graphs.

- We propose a general GNN framework (T2-GNN) for incomplete graphs through teacher-student distillation, which introduces feature-level and structure-level teacher models to provide targeted guidance for student (base GNN), and designs a dual distillation mode to enhance the effective transfer of knowledge. It can improve the performance of the base GNN while avoiding the mutual interference of features and structure, and is easy to be combined with an arbitrary base GNN.

- By combining with four classical base GNNs, our framework consistently achieves performance improvements across eight benchmark datasets. Experiments comparing with different kinds of baselines further verify the robustness of the new framework.

Preliminaries

Notations

Given an undirected graph $G = (V, X)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of n nodes, $X \in \mathbb{R}^{n \times d}$ denotes the feature matrix and d is the feature dimension. We let $A = [a_{ij}] \in \{0, 1\}^{n \times n}$ represents the adjacency matrix, where $a_{ij} = 1$ denotes there exists an edge between nodes v_i and v_j, or 0 otherwise. An incomplete graph means that some elements x_{ij} in the feature matrix X are missing, and a_{ij} in the adjacency matrix A should be set to 1 and set to 0 by mistake. In this paper, we focus on the semi-supervised node classification task, which means that only a subset of

nodes ($V_L, |V_L| \ll n$) have ground truth labels Y. Our task is to predict the labels of $V \backslash V_L$.

Graph Neural Networks

Graph Neural Networks (GNNs) have shown great power in tackling various analysis tasks on graph-structured data. Most of the existing GNNs follow the message-passing mechanism, that is, propagating and aggregating information guided by the graph structure (neighbors), such as GCN, GAT, etc. Formally, in the process of GNN learning node representations, the representation of node u at the lth layer can be formalized as follows:

$$p_v^{(l)} = \text{Prop}^{(l)} \left(x_v^{(l-1)} \right), \quad v \in \mathcal{N}(u), \qquad (1)$$

$$x_u^{(l)} = \text{Agg}^{(l)} \left(x_u^{(l-1)}, \{ p_v^{(l)} : v \in \mathcal{N}(u) \} \right), \qquad (2)$$

where $\mathcal{N}(u)$ is the set of the neighbors of node u, $\text{Prop}^{(k)}(\cdot)$ is the node representation transformation in the propagation process and $\text{Agg}^{(k)}(\cdot)$ aggregates information of its neighborhood and itself to obtain the final node representation. Both $\text{Prop}^{(k)}(\cdot)$ and $\text{Agg}^{(k)}(\cdot)$ are functions implemented by neural networks in the lth convolutional layer of GNN. Given u's input features x^0 and its neighborhoods $\mathcal{N}(u)$, we can use Eq.(1) and Eq.(2) to obtain its final representation.

However, the initial features and graph structure of real-world graphs are often incomplete, which will make GNNs using the above propagation and aggregation functions learn the suboptimal node representations. In this paper, we focus on improving the performance of GNNs when both features and graph structure are incomplete.

Knowledge Distillation

Knowledge distillation aims to distill knowledge from a cumbersome teacher model into a lightweight student model so that the student can hold a similar performance as the teacher's. Vanilla knowledge distillation was first proposed by Hinton (Hinton, Vinyals, and Dean 2015), which provides softened softmax labels through the teacher model to guide the training of the student model. Given the output logits Z_T and Z_S of the teacher model and the student model, the distillation loss is as follows:

$$\mathcal{L}_{KD} = \mathcal{H}(\sigma_S(Z_S; \rho), \sigma_T(Z_T; \rho)), \qquad (3)$$

where $\mathcal{H}$ is negative cross-entropy loss or KL-divergence, σ denotes the softmax with temperature ρ. The larger the ρ, the smoother the output probability, the information carried by the negative labels will be relatively amplified, and the model training will pay more attention to the negative labels.

In addition, some distillation models based on intermediate representations are constantly being proposed. They achieve knowledge distillation by aligning intermediate representations (such as attention map (Zagoruyko and Komodakis 2017), intermediate features (Huang and Wang 2017)) between the teacher model and the student model. Common alignment functions such as L_2 loss:

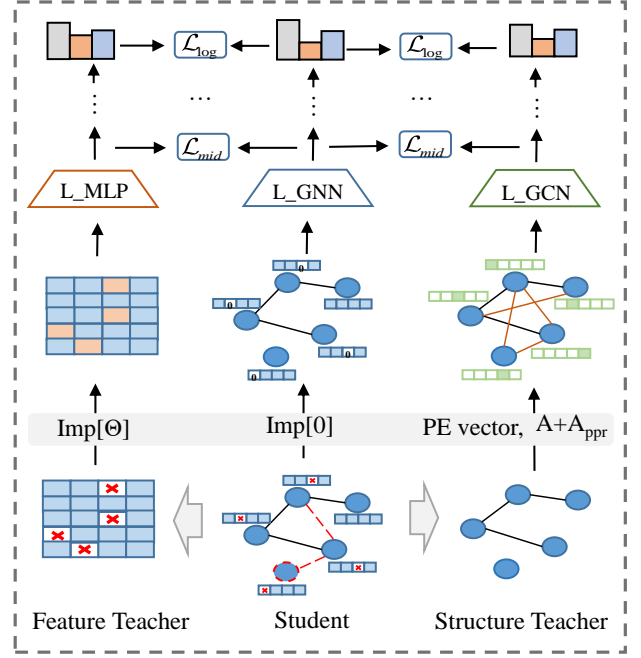$$\mathcal{L}_2 = ||R_S - R_T||_2^2, \qquad (4)$$

Figure 2: Overview of the T2-GNN framework.

where R_S and R_T are the intermediate representations of the student and teacher models, respectively.

To the best of our knowledge, we employ knowledge distillation for the first time to improve the robustness of GNNs on incomplete graphs, effectively avoiding inaccurate feature completion or structure enhancement that is exacerbated by the potential interference between features and graph structure.

T2-GNN Framework

Overview

To address the problem that GNNs fail to perform well on incomplete graphs, we propose a novel and general GNN framework based on teacher-student distillation that can effectively model both incomplete features and graph structure simultaneously and improve the robustness of GNNs, namely T2-GNN. The new framework aims to design personalized feature teacher and structure teacher models on incomplete graphs, and provide targeted and effective guidance for student model (such as GCN) and avoid interference between features and structure through knowledge distillation. It has three components: feature teacher with parameterized completion, structure teacher with personalized enhancement, as well as a dual distillation module (Fig. 2). The feature teacher with parameterized completion is designed to capture valuable information in incomplete features while avoiding the potential interference of graph structure on its learning process. It assigns parameterized imputation to missing features, uses multi-layer perceptrons to model node features, and conducts joint training for feature completion and node representation. The structure teacher with personalized enhancement is designed to ef-

fectively model the incomplete graph structure while avoiding the potential interference of node features during model learning. It uses personalized PageRank to enhance the incomplete initial graph structure, and replaces the incomplete initial node features with positional encoding vector to further enhance the personalized structure information of nodes. Then it feeds the new structure augmented graph into GCN for model training. The dual distillation module is designed to maximize the distillation of targeted teachers' expert knowledge into the student model to effectively improve its performance on incomplete graphs. It adopts both contrast-based intermediate representation alignment and logit-based soft label-guided dual distillation mode to achieve effective and comprehensive transfer of teacher's knowledge to student. Finally, the student model is jointly optimized by the distillation losses and the original classification loss.

Feature Teacher with Parameterized Completion

Previous GNN methods for modeling incomplete node features usually rely on pre-defined assumptions for feature completion. For example, SAT (Chen et al. 2022) assumes that the graph structure is complete and use the matching relationship between structure and feature for feature completion, GCNMF (Taguchi, Liu, and Murata 2021) assumes that all features follow a same data distribution and use distribution parameters for feature imputation. However, such strong assumptions are not universal because real-world graph data are usually complex and diverse, and may also be accompanied by incomplete graph structure. In addition, the potential mutual interference between features and graph structure in the learning process of GNN has also been confirmed in recent studies (Yang et al. 2022; Wang et al. 2020).

To this end, we propose to use a parameterized completion mechanism to complete the missing features through the learning ability of the model itself, so as to achieve the reciprocity of feature completion and model learning. Specifically, given an incomplete feature matrix $X \in \mathbb{R}^{n \times d}$, we use learnable neural network parameters to impute missing features. The feature matrix for parameterized imputation can be formalized as:

$$\overline{X}_{ij} = \begin{cases} \Theta_{ij} & \text{if } X_{ij} \text{ is missing;} \\ X_{ij} & \text{otherwise,} \end{cases} \tag{5}$$

where $\Theta \in \mathbb{R}^{n \times d}$ is a randomly initialized neural network parameter matrix.

In order to fully capture the valuable information in the features while avoiding the potential interference caused by the graph structure, we only utilize node features to construct feature teacher model. We employ a multi-layer perceptron (MLP) to model the features. Thus we can obtain the output logits Z_{fea} and intermediate representations R_{fea} of the feature teacher for distillation:

$$Z_{fea}, R_{fea} = f_{mlp}(\overline{X}), \tag{6}$$

where $f_{\text{mlp}}(\cdot)$ is the MLP with the ReLU activation function. Its training objective function can be written as:

$$\min_{\Theta_{fea}} = \sum\nolimits_{v_i \in \mathcal{T}_\mathcal{V}} \mathcal{J}_c\left(\sigma(z_{fea,i}), y_i\right), \tag{7}$$

where Θ_{fea} is all parameters of the feature teacher including imputation parameter Θ, $\sigma(\cdot)$ the softmax function, $\mathcal{T}_V$ the nodes in training set, $\mathcal{J}(\cdot)$ the cross entropy loss, and y_i the true one-hot label of node v_i.

Structure Teacher with Personalized Enhancement

Neighbor nodes are crucial for the target node to learn representations, but the real-world graph structure is usually incomplete, so that some neighbors of the target node are missing, and even becomes an isolated node. Therefore, we first adopt personalized PageRank, which is very effective in expanding important neighbors in the graph (Klicpera, Bojchevski, and Günnemann 2019), to augment the incomplete graph structure.

Personalized PageRank (PPR) (Haveliwala 2002) takes the graph structure as input, adopts a restarted random walk strategy to sample neighbors for the target node v_i, and calculates the ranking score p_{ij} between v_i and the sampled node v_j. A larger p_{ij} indicates that the two nodes are more related. Similar to (Bojchevski et al. 2020), we use a push iteration method to compute ranking scores, which can be approximated effectively even for very large networks. Formally, the PPR matrix P is calculated as:

$$P = (1 - \alpha)\hat{A}P + \alpha I, \tag{8}$$

where α is the reset probability, $\hat{A} = A + I$. Then, since there are inevitably some unimportant scores in P, we only keep the top k important scores for each node in the P and set the other elements to zero to get the PPR-based structure matrix A_{ppr}.

Considering that the initial graph structure still carries rich and useful information, we combine the initial adjacency matrix A with the PPR-based matrix A_{ppr} to construct the enhanced graph structure: $\overline{A} = A + A_{ppr}$.

Different from the feature teacher only uses node features for model learning, we adopt positional encoding (PE) vectors to replace the initial incomplete features to construct a complete personalized enhancement graph for structure teacher. PE can effectively model position information in sequence data (Vaswani et al. 2017). Here we intend to design the PE vectors to further extract the personalized (location) information of nodes in the graph structure. We use one-hot encoding to compute PE vectors, and in order to improve the scalability of PE vectors in large-scale graph data, we use a linear transformation to transform the vectors into the same dimension as the initial node feature. Formally, the PE vector of node v is:

$$PE(v) = W \cdot x_v + b, \tag{9}$$

where W and b are learnable parameter matrix and bias vector, x_v the one-hot vetor of node v.

Then we feed the complete enhancement graph into multi-layer GCN to obtain the output logit and intermediate representation of the structure teacher for distillation.

$$Z_{str}, R_{str} = f_{gcn}(\overline{A}, PE), \tag{10}$$

where $f_{\text{gcn}}(\cdot)$ is the GCN with multiple convolutional layers. Its training objective function is the same as that of the feature teacher (Eq.(7)).

Dual Distillation and Model Training

After obtaining the output logits and intermediate representations of the structure and feature teachers, we design a dual distillation mode to provide targeted and effective guidance for the student model to the greatest extent. The output logits are demonstrated to contain rich information (Hinton, Vinyals, and Dean 2015), so we first use a logit-based distillation to achieve the knowledge transfer from the teacher model to the student model. Without loss of generality, we adopt KL-divergence as the distillation loss:

$$\mathcal{L}_{log} = \mathcal{H}_{KL}\left(softmax(\frac{Z_s}{\rho})||softmax(\frac{Z_t}{\rho})\right), \quad (11)$$

where Z_s is the output logits of the student model, Z_t the output logits of the feature or structure teacher model, and ρ the temperature parameter that controls the smoothness of the output.

The logit-based distillation can be considered to provide more label supervision information for student model by generating soft labels. Considering the negative effects of missing features and edges on the training process of student model, we further introduce intermediate representation distillation to guide the training of student model, which can also be considered as an indirect completion to missing features or edges in the training process of student model. We adopt a contrast-based intermediate representation alignment mechanism to achieve targeted guidance from teachers to student training by maximizing the consistency of intermediate embedding from the teacher model and the student model for the same node i (i.e., positive samples (r_i^s, r_i^t)), and minimizing the consistency of the intermediate representation of i in the student model and the intermediate representations of other nodes in the teacher model (i.e., negative samples (r_i^s, r_j^t)). The alignment loss is formalized as:

$$\mathcal{L}_{mid}(R_s, R_t) = \frac{1}{|V|}\sum_{v_i \in V}\log\frac{e^{\langle r_i^s, r_i^t\rangle}}{e^{\langle r_i^s, r_i^t\rangle} + \sum_{j\neq i}e^{\langle r_i^s, r_j^t\rangle}}, \quad (12)$$

where $\langle \cdot, \cdot \rangle = \cos(\cdot, \cdot)/\rho'$, and ρ' is a scaler temperature parameter. Then, we combine the above two losses as the final distillation loss between the student model and an arbitrary teacher model: $\mathcal{L}_{dual} = \mathcal{L}_{log} + \mathcal{L}_{mid}$.

Considering that the missing rates of features and edges in real-world graphs may be different, we adopt the parameter λ to balance the two distillation losses. The overall training loss of the student model is as follows:

$$\mathcal{L}_{final}^{stu} = \mathcal{J}_c^{stu} + \lambda\mathcal{L}_{dual}^{fea} + (1-\lambda)\mathcal{L}_{dual}^{str}, \quad (13)$$

where $\mathcal{J}_c^{stu}$ is standard cross-entropy loss, λ the balance hyperparameter.

Experiments

Experimental setup

Datasets. We use eight real-world datasets of different scales for model evaluation (Table 1). They are two

Datasets	Nodes	Edges	Features	Classes
Texas	183	295	1703	5
Cornell	183	295	1703	5
Wisconsin	251	499	1703	5
Chameleon	2,277	31,421	2,325	5
Cora	2,708	5,278	1,433	7
Citeseer	3,327	4,676	3,703	6
Squirrel	5,201	198,493	2,089	5
Pubmed	19,717	44,327	500	3

Table 1: Statistics of datasets.

Wikipedia networks Chameleon and Squirrel (Rozemberczki, Allen, and Sarkar 2021), three webpage networks Cornell, Texas and Wisconsin[1], and three citation networks Cora, CiteSeer and PubMed (Sen et al. 2008; Namata et al. 2012).

Baselines. We first plug four GNNs as student models into T2-GNN, including GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and APPNP (Klicpera, Bojchevski, and Günnemann 2019), to verify whether T2-GNN can boost the performance of node classification.

Then we compare our T2-GNN with two GNNs for features completion (SAT (Chen et al. 2022) and GC-NMF (Taguchi, Liu, and Murata 2021)) and two GNNs for structure enhancement (IDGL (Chen, Wu, and Zaki 2020) and PTDNet (Luo et al. 2021)). For a fair comparison, all methods use GCN as the backbone.

Finally, we use different training ways to construct two new variant models to further verify the effectiveness of T2-GNN: (1) combining feature completion and structure enhancement into a single-teacher model to guide student' training, named singleT; (2) conducting a joint end-to-end training of feature teacher, structure teacher and student model, instead of training the teacher models alone, named T2-GNN (online).

Implementation Details. For all datasets, we randomly split nodes of each class into 60%, 20%, and 20% for training, validation and testing, and generate 10 random splits for all methods. Since the missing degree of features and edges in real-world datasets is agnostic, we randomly mask 30% of features and edges in each dataset for model evaluation. All parameters in the baselines adopt the original parameters in their paper. In our new framework T2-GNN, for feature teacher, we use a two-layer MLP with 512 units in the hidden layer. For structure teacher, we adopt GCN with two graph convolutional layers, and search the top k for PPR-based structure enhancement strategy in $\{5, 10, \cdots, 25\}$. We set learning rate to 0.001 and weight decay to 0.0005 for both teacher models. For student model, we use its original parameter settings in the paper. We search the temperature parameter ρ in the distillation loss in $\{1, 2, \cdots, 5\}$ and the balance parameter λ in $\{0.1, 0.2, \cdots, 0.9\}$.

---

[1]http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb

Datasets	Texa.	Corn.	Wisc.	Cham.	Cora	Cite.	Squi.	Pubm.	avg
GCN	46.48	48.90	50.39	58.37	82.77	68.84	40.09	83.51	59.90
T2-GCN	65.40	67.29	73.39	60.81	83.84	72.32	44.33	84.85	69.02
+Impv.	18.92	18.39	23.00	2.44	1.07	3.58	4.24	1.34	9.12
GAT	50.27	55.67	53.72	57.56	83.54	70.00	36.58	84.33	61.45
T2-GAT	61.35	64.86	62.35	58.39	84.46	72.42	38.87	85.07	65.96
+Impv.	11.08	9.19	8.63	0.83	0.92	2.42	2.33	0.74	4.51
GraphSAGE	70.81	64.59	69.41	59.05	82.93	70.96	41.86	84.96	68.07
T2-SAGE	78.10	75.40	82.54	62.89	84.62	71.39	45.78	85.56	73.28
+Impv.	7.29	10.81	13.13	3.84	1.69	0.43	3.92	0.60	5.21
APPNP	47.83	55.94	53.52	44.18	83.48	70.61	28.57	78.77	57.86
T2-APPNP	57.83	66.48	66.27	50.41	84.08	71.77	36.93	83.36	64.63
+Impv.	10.00	10.54	12.75	6.23	0.60	1.16	8.36	4.59	6.77

Table 2: Comparisons on node classification (Percent). "+Impv" is the performance improvement of T2-GNN for base GNN.

Datasets	Texa.	Corn.	Wisc.	Cham.	Cora	Cite.	Squi.	Pubm.	avg
GCN	46.48	48.90	50.39	58.37	82.77	68.84	40.09	83.51	59.90
SAT	63.89	**76.68**	67.49	56.57	64.59	45.87	35.23	-	58.61
GCNMF	57.28	55.43	53.02	47.75	83.08	71.75	30.56	67.08	58.24
IDGL	62.16	54.05	60.78	48.68	79.88	65.80	33.11	82.60	60.88
PTDNet	61.62	64.87	73.14	48.42	74.93	72.29	30.45	82.49	63.52
singleT	53.81	54.76	57.10	55.93	80.19	67.22	40.32	81.03	61.29
T2-GCN (online)	60.53	74.01	69.42	54.22	80.11	65.31	**45.19**	83.04	66.47
T2-GCN	**65.40**	67.29	**73.39**	**60.81**	**83.84**	**72.32**	44.33	**84.85**	69.02

Table 3: Comparisons on node classification (Percent). The best results are in bold and the second best is underlined. "-": OOM.

Missing rate	0%	20%	40%	60%	80%
GCN	62.33	60.82	57.85	52.43	45.17
T2-GCN	71.30	69.93	66.54	60.44	54.60

Table 4: Comparisons of the average accuracy on eight datasets under different missing rates (Percent).

Node Classification

We report here the accuracy on the node classification task (Table 2 and Table 3), and compare the average accuracy of node classification on eight datasets under different features and edges missing rates (Table 4). We have the following observations.

In Table 2, T2-GNN consistently improves the performance of the four backbones on all datasets. The average improvement is 9.12%, 4.51%, 5.21% and 6.77% on GCN, GAT, GraphSAGE and APPNP, respectively. In particular, T2-GNN has achieved significant performance improvement on small datasets (Texas, Cornell and Wisconsin). This is because it is well known that small datasets are more prone to over-smoothing problems than large datasets under the same network depth. However, the missing of features or edges may be accompanied by the missing of important and individualized features or structure of the node, thus aggravating the over-smoothing problem. T2-GNN individually models incomplete features and structure, and enhances the personalized nature of nodes in features and structure, respectively, which alleviates above problem to some extent.

T2-GNN outperforms the baseline methods in most cases (Table 3), which validates the necessity of GNN modeling incomplete features and graph structure simultaneously. Furthermore, GCN outperforms feature completion methods (SAT and GCNMF) and structure-enhanced methods (IDGL and PTDNet) in many cases. This is because these methods generally believe that the potential relationship between features and structure is beneficial to solve the problem of incomplete features (or structure). However, when both features and structure are incomplete, this mechanism may have a negative influence on model performance due to potential interference between features and structure. T2-GNN effectively avoids this negative influence by completing features and structure separately.

T2-GNN outperforms singleT and T2-GNN (online) in most cases, and even GCN can perform better than these two variant models on 4 out of 8 datasets. Although T2-GNN (online) also models incomplete features and structure separately, the mechanism of joint training still cannot effectively avoid the potential (indirect) interference between the two during training. T2-GNN trains each model in an individually optimized manner so that it is free from this interferences.

In addition, as shown in Table 4, T2-GCN has a stable and significant performance improvement against GCN, with 8.97%, 9.11%, 8.69%, 8.01% and 9.43% improvement (on average on all the datasets) under different masking ratios, which further validates the effectiveness and robustness of T2-GNN.
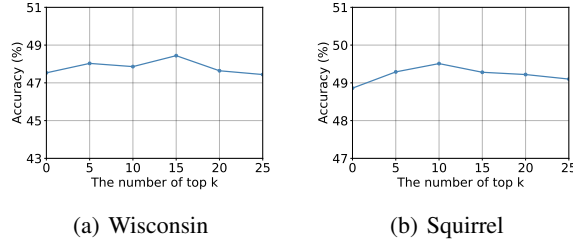
(a) Wisconsin   (b) Squirrel

Figure 3: Parameter sensitivity analysis of top k.

(a) Wisconsin   (b) Squirrel

Figure 5: Parameter sensitivity analysis of λ and ρ.

Ablation Study

We compare T2-GNN with its four variants on all datasets to intuitively understand the effectiveness of each component.

- w/o_Teacher_str: T2-GNN without structure teacher.
- w/o_Teacher_fea: T2-GNN without feature teacher.
- w/o_Distill_log: T2-GNN without logit distillation.
- w/o_Distill_mid: T2-GNN without intermediate representation distillation.

From the results in Fig. 4, we can draw the following conclusions: (1) T2-GNN consistently outperforms all variants, indicating the effectiveness of using both feature and structure teachers as well as the dual distillation. (2) w/o_Teacher_fea outperforms w/o_Teacher_str on the Chamelon and Squirrel datasets, but it is opposite on other datasets. This indicates that the degree of credibility of expert knowledge from features and structure in different datasets is different, so we need to weigh the weights of feature teacher and structure teacher to better provide targeted guidance (which is solved in our paper by balancing parameter λ). (3) Both w/o_Distill_log and w/o_Distill_mid achieve good performance but are lower than T2-GNN. This indicates that the dual distillation mechanism more effectively and comprehensively realizes the knowledge transfer from teacher model to student model.
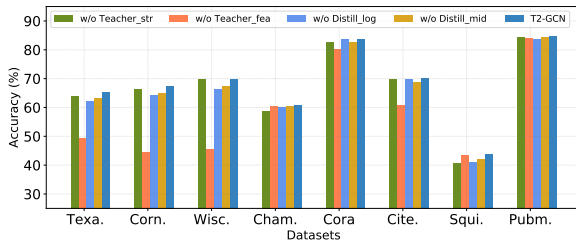

Figure 4: Comparisons of the new T2-GNN with its four variants on node classification.

Parameter Analysis

Taking datasets Wisconsin and Squirrel of different scales as examples, we investigate the sensitivity of hyperparameters in T2-GNN.

Analysis of top k. The parameter k in our framework refers to the number of PPR-based neighbor nodes per node
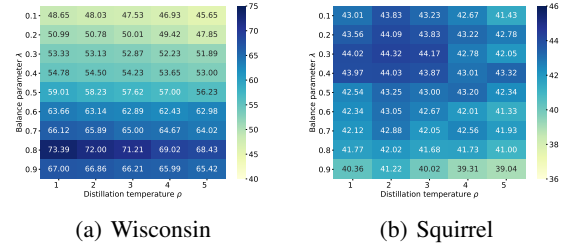
in the structure enhancement. We vary it from 0 to 25. As shown in Fig. 3, the performance degraded when k was large or small. This observation indicates that a small number of structurally enhancing relationships is not sufficient to obtain informative node embeddings, whereas too many may introduce noise and weaken the information propagation.

Analysis of balance parameter λ. In our framework, λ is used to balance the weight of distillation loss from feature and structure teachers. Here we investigate the influence of λ on node classification by varying it from 0.1 to 0.9 with a step size of 0.1. As shown in Fig. 5, T2-GNN achieves the best performance when $\lambda = 0.8$ on Wisconsin and $\lambda = 0.4$ on Squirrel. Its performance shows a trend of first rising and then descending and reaches its best at a certain value. This observation indicates that node features and graph structure may have different degrees of trustworthiness when modeling incomplete graphs, because we cannot determine which missing information is more important to the node, as is the case in the real-world graph.

Analysis of distillation temperature ρ. In our framework, ρ is a temperature parameter in logit-based distillation loss. The larger the ρ, the model training will pay more attention to the negative labels. We also investigate the influence of ρ on node classification by varying it from 1 to 5. As Fig. 5 shows, ρ is relatively insensitive and a small ρ (e.g., 1 or 2) has a slight advantage. This observation indicates that due to the incompleteness of the graph itself, in order to avoid the influence of noise in the negative label, we need a relatively small distillation temperature.

Conclusion

In this paper, we propose a general GNN framework based on teacher-student distillation, which can effectively model incomplete graphs and avoid the interference between features and structure. In specific, we design feature-based and structure-based teacher models to capture effective information in features and structure, respectively. Then with the well-trained teacher model, targeted guidance is provided for the student model (base GNN, such as GCN) from the feature level and the structure level respectively. Finally, the dual distillation mode is adopted to achieve the comprehensive and effective transfer of knowledge from teachers to students. Extensive experiments demonstrate that the proposed new framework can effectively improve the performance of GNN on incomplete graphs, and further comparisons with baselines demonstrate its robustness.

Acknowledgments

References

Bojchevski, A.; Klicpera, J.; Perozzi, B.; Kapoor, A.; Blais, M.; Rózemberczki, B.; Lukasik, M.; and Günnemann, S. 2020. Scaling Graph Neural Networks with Approximate PageRank. In *Proceedings of SIGKDD*, 2464–2473.

Chen, H.; Yin, H.; Sun, X.; Chen, T.; Gabrys, B.; and Musial, K. 2020. Multi-level Graph Convolutional Networks for Cross-platform Anchor Link Prediction. In *Proceedings of SIGKDD*, 1503–1511.

Chen, X.; Chen, S.; Yao, J.; Zheng, H.; Zhang, Y.; and Tsang, I. W. 2022. Learning on Attribute-Missing Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(2): 740–757.

Chen, Y.; Wu, L.; and Zaki, M. J. 2020. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings. In *Proceedings of NeurIPS*.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of NeurIPS*, 1024–1034.

Haveliwala, T. H. 2002. Topic-sensitive PageRank. In *Proceedings of WWW*, 517–526.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Huang, Z.; and Wang, N. 2017. Like What You Like: Knowledge Distill via Neuron Selectivity Transfer. *arXiv preprint arXiv:1707.01219*.

Jin, D.; Huo, C.; Liang, C.; and Yang, L. 2021a. Heterogeneous Graph Neural Network via Attribute Completion. In *Proceedings of WWW*, 391–400.

Jin, D.; Yu, Z.; Jiao, P.; Pan, S.; Yu, P. S.; and Zhang, W. 2021b. A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Trans. Knowl. Data Eng.*

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of ICLR*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *Proceedings of ICLR*.

Liu, Z.; Wan, M.; Guo, S.; Achan, K.; and Yu, P. S. 2020. BasConv: Aggregating Heterogeneous Interactions for Basket Recommendation with Graph Convolutional Neural Network. In *Proceedings of SDM*, 64–72.

Luo, D.; Cheng, W.; Yu, W.; Zong, B.; Ni, J.; Chen, H.; and Zhang, X. 2021. Learning to Drop: Robust Graph Neural Network via Topological Denoising. In *Proceedings of WSDM*, 779–787.

Namata, G.; London, B.; Getoor, L.; and Namatag, B. H. and BLONDON and GETOOR and BERT@CS.UMD.EDU. 2012. Query-driven Active Surveying for Collective Classification. In *Mining and Learning with Graphs*.

Rozemberczki, B.; Allen, C.; and Sarkar, R. 2021. Multi-Scale attributed node embedding. *J. Complex Networks*, 9(2).

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Mag.*, 29(3): 93–106.

Taguchi, H.; Liu, X.; and Murata, T. 2021. Graph convolutional networks for graphs containing missing features. *Future Gener. Comput. Syst.*, 117: 155–168.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Proceedings of NeurIPS*, 5998–6008.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of ICLR*.

Wang, X.; Zhu, M.; Bo, D.; Cui, P.; Shi, C.; and Pei, J. 2020. AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. In *Proceedings of SIGKDD*, 1243–1253.

Wu, L.; Cui, P.; Pei, J.; and Zhao, L. 2022. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Singapore.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1): 4–24.

Xu, F.; Lian, J.; Han, Z.; Li, Y.; Xu, Y.; and Xie, X. 2019. Relation-Aware Graph Convolutional Networks for Agent-Initiated Social E-Commerce Recommendation. In *Proceedings of CIKM*, 529–538.

Yang, L.; Zhou, W.; Peng, W.; Niu, B.; Gu, J.; Wang, C.; Cao, X.; and He, D. 2022. Graph Neural Networks Beyond Compromise Between Attribute and Topology. In *Proceedings of WWW*, 1127–1135.

You, J.; Ying, R.; and Leskovec, J. 2019. Position-aware Graph Neural Networks. In *Proceedings of ICML*, 7134–7143.

Yu, Z.; Jin, D.; Liu, Z.; He, D.; Wang, X.; Tong, H.; and Han, J. 2021. AS-GCN: Adaptive Semantic Architecture of Graph Convolutional Networks for Text-Rich Networks. In *Proceedings of ICDM*, 837–846.

Zagoruyko, S.; and Komodakis, N. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *Proceedings of ICLR*.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.

Zhu, Y.; Xu, W.; Zhang, J.; Du, Y.; Zhang, J.; Liu, Q.; Yang, C.; and Wu, S. 2022. A Survey on Graph Structure Learning: Progress and Opportunities. *arXiv preprint arXiv:2103.03036v2*.

Zhu, Y.; Xu, W.; Zhang, J.; Liu, Q.; Wu, S.; and Wang, L. 2021. Deep Graph Structure Learning for Robust Representations: A Survey. *arXiv preprint arXiv:2103.03036*.