

SELF-SUPERVISED GUIDED HYPERGRAPH FEATURE PROPAGATION FOR SEMI-SUPERVISED CLASSIFICATION WITH MISSING NODE FEATURES

Chengxiang Lei, Sichao Fu^{*,†}, Yuetian Wang, Wenhao Qiu, Yachen Hu, Qinmu Peng[‡] and Xinge You

School of Electronic Information and Communications, Huazhong University of Science and Technology

ABSTRACT

Graph neural networks (GNNs) with missing node features have recently received increasing interest. Such missing node features seriously hurt the performance of the existing GNNs. Some recent methods have been proposed to reconstruct the missing node features by the information propagation among nodes with known and unknown attributes. Although these methods have achieved superior performance, how to exactly exploit the complex data correlations among nodes to reconstruct missing node features is still a great challenge. To solve the above problem, we propose a self-supervised guided hypergraph feature propagation (SGHFP). Specifically, the feature hypergraph is first generated according to the node features with missing information. And then, the reconstructed node features produced by the previous iteration are fed to a two-layer GNNs to construct a pseudo-label hypergraph. Before each iteration, the constructed feature hypergraph and pseudo-label hypergraph are fused effectively, which can better preserve the higher-order data correlations among nodes. After then, we apply the fused hypergraph to the feature propagation for reconstructing missing features. Finally, the reconstructed node features by multi-iteration optimization are applied to the downstream semi-supervised classification task. Extensive experiments demonstrate that the proposed SGHFP outperforms the existing semi-supervised classification with missing node feature methods.

Index Terms— Missing node features, Graph neural networks, Semi-supervised classification

1. INTRODUCTION

Graphs are widely applied for many real-world scenarios, such as social networks [1], citation networks [2], traffic networks [3], molecular networks [4]. In recent years, with the rapid development of deep learning, graph neural networks (GNNs) [5, 6] have achieved remarkable success in graph-structured data with complex data correlations. GNNs

typically operate by a message propagation scheme, where each node propagates its feature representations along the constructed edges. Meanwhile, the feature representation of each node is updated by aggregating the representations from its neighbors.

However, the existing GNNs variants [7] typically assume that all node feature attributes are fully observed during the training process. In fact, in many real-world applications, some feature attributes are unobserved due to resource limits or privacy concerns [8]. For example, in social networks, some users are unwilling to provide their ages and gender. Thus, when directly utilizing the existing GNNs variants to deal with the above tasks with missing node features, their performance will have a big deterioration.

To solve the above issue, many matrix factorization-based methods [9, 10] have been proposed to reconstruct missing node features. However, these methods cannot make full use of the topological information between data. Recently, graph signal processing [11] provides several methods for interpolating signals on graphs, they generalize Fourier analysis to graphs to reconstruct missing signals on graphs. Such methods are too computationally intensive so they are infeasible for graphs with thousands of nodes. Very recently, several methods extend GNNs to tackle missing node features directly. For example, SAT [12] assumes that the structure and feature information on the graph share the same latent space and develops a distribution-matching strategy to reconstruct missing features. GCNMF [13] adapts GCN [14] to graphs with missing features by representing the missing features with a Gaussian mixture model. PaGNN [15] develops partial aggregation-based GNNs that only propagate the observed features. FP [8] propagates the known features to the nodes with unknown features iteratively to reconstruct missing features. However, FP only considers pairwise connection relationships between data and FP also assumes that each node has the same influence on all neighbors, which cannot accurately describe the local geometric distribution between data.

In this paper, we propose a self-supervised guided hypergraph feature propagation (SGHFP) for semi-supervised classification with missing node features. Specifically, the feature and pseudo-label hypergraph are first computed according to the node features with missing information and reconstructed node features generated by the previous iteration in turn. Be-

This work was supported in part by the National Natural Science Foundation of China under Grant 62172177, in part by the Fundamental Research Funds for the Central Universities under Grant 2022JYCXJJ034 and YCJJ202204016. *Work was done when Sichao Fu was interning at JD Retail POMC. †Sichao Fu and Qinmu Peng are the corresponding authors.

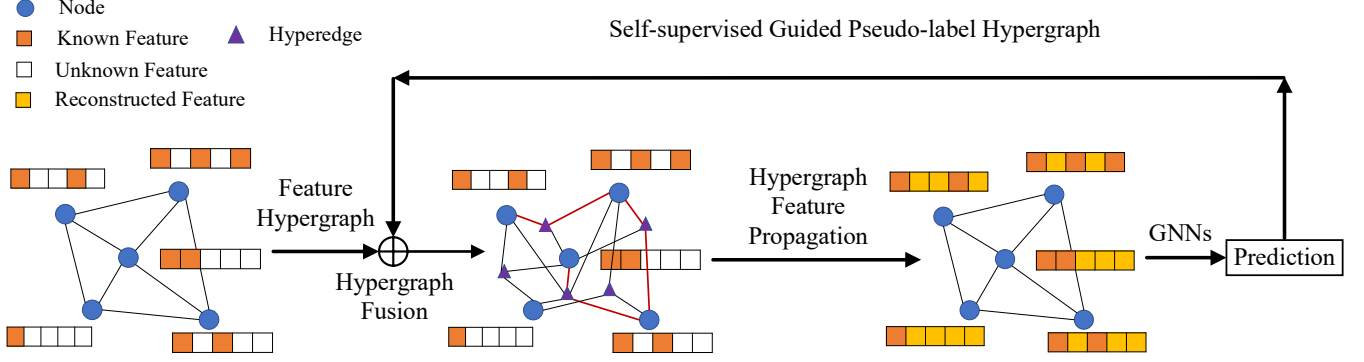


Fig. 1. A diagram illustrating our Self-supervised Guided Hypergraph Feature Propagation framework.

fore each iteration, the obtained feature and pseudo-label hypergraph are further fused into an effective hypergraph. Compared to the single feature or pseudo-label hypergraph, the fused hypergraph can better describe the complex high-order structure information between data. Following, the fused hypergraph and FP are combined to reconstruct missing features in each iteration. After multi-iterations optimization, the reconstructed node features can be applied to downstream semi-supervised classification tasks. To validate the effectiveness of SGHFP, we conduct extensive experiments on four benchmarks. Experiment results demonstrate that our approach outperforms many state-of-the-art methods. The main contributions are summarized as follows:

- Compared with the traditional graph, the proposed hypergraph can simultaneously utilize the higher-order correlations from feature and pseudo-label hypergraphs to update the inaccurate connection relationships.
- Our proposed SGHFP is an independent module, which can combine any GNNs variants for any graph representation learning tasks.
- Extensive experiments show that the proposed SGHFP outperforms many existing semi-supervised classifications with missing node feature methods.

2. SELF-SUPERVISED GUIDED HYPERGRAPH FEATURE PROPAGATION

2.1. Problem Definition and Notation Description

Give a simple graph $\mathcal{G} = (V, E)$, where $V = \{v_i | i = 1, \dots, n\}$ is the set of nodes, and $E = \{e_{ij} | i = 1, \dots, n, j = 1, \dots, n\}$ is the set of edges. Let $A \in \mathbb{R}^{n \times n}$ denotes the adjacency relationship matrix, where $A_{ij} = 1$ if $e_{ij} \in E$, and $A_{ij} = 0$ if $e_{ij} \notin E$. $X \in \mathbb{R}^{n \times d}$ denote the nodes features matrix, where d is the dimension of node features.

Different from a simple graph, a hyperedge can connect two or more nodes. Let $H = (\tilde{V}, \tilde{E}, W)$ denotes a hyper-

graph, where W denotes the weight matrix of all hyperedges, and $H \in \mathbb{R}^{|\tilde{V}| \times |\tilde{E}|}$ is a incidence matrix, i.e.

$$h(\tilde{v}, \tilde{e}) = \begin{cases} 1, & \text{if } \tilde{v} \in \tilde{e} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

For a node $\tilde{v} \in \tilde{V}$, its degree is defined as $d(\tilde{v}) = \sum_{\tilde{e} \in \tilde{E}} w(\tilde{e}) h(\tilde{v}, \tilde{e})$. For a hyperedge $\tilde{e} \in \tilde{E}$, its degree is defined as $\delta(\tilde{e}) = \sum_{\tilde{v} \in \tilde{V}} h(\tilde{v}, \tilde{e})$. Let D_e denote all hyperedges' degree matrix and D_v denote all nodes' degree matrix. In addition, we let $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$ and $\Delta = I - \Theta$, where Δ is called as hypergraph Laplacian.

$\tilde{V}_k \subseteq \tilde{V}$ denotes the set of nodes where the features are known, and $\tilde{V}_u = V_k^c = \tilde{V} \setminus \tilde{V}_k$ denote the unknown ones. In this paper, the problem we focus on is how to better reconstruct the unknown features \tilde{x}_u , given the known features \tilde{x}_k and the graph structure \mathcal{G} .

2.2. Hypergraph Fusion

To accurately describe the local geometric distribution among nodes, We first generate a feature hypergraph G_f according to the node features with missing information. Each time one node is selected as the centroid, and all its neighbors are linked as a hyperedge. Second, the reconstructed node features produced by the previous iteration are fed to a two-layer GNNs to construct a pseudo-label hypergraph G_{pl} .

Denote v_a and v_b are two nodes, from [16], we can know that the label influence of v_a on v_b equals to the to the cumulative normalized feature influence of v_a on v_b after k iterations of propagation:

$$\mathbb{E}[I_l(v_a, v_b; k)] = \sum_{j=1}^k \tilde{I}_f(v_a, v_b, j). \quad (2)$$

Equation 2 shows that pseudo-label hypergraph is beneficial to increase the intra-class feature influence. To highlight the accuracy of connection relationships, the constructed feature hypergraph G_f and pseudo-label hypergraph G_{pl} are further combined to generate a fused hypergraph before each iteration. To reduce complexity, we fuse the two graphs into a

sparse matrix. G_f is used as the indices of specified elements, and G_{pl} is used as the corresponding values. As shown in Fig.1, hyperedges connecting two nodes of the same class are bold so that features can be more easily propagated among nodes along hyperedges with stronger connections.

2.3. Hypergraph Feature Propagation

Similar to FP [8], we reconstruct the unknown node features \tilde{x}_u through interpolation that minimizes *Dirichlet energy*: $\ell(x, \mathcal{G}) = \frac{1}{2} x^\top \Delta x = \frac{1}{2} \sum_{ij} \theta_{ij} (x_i - x_j)^2$, where θ_{ij} are the individual entries of the normalized incidence matrix Θ . Dirichlet energy, which represents how much a function changes in a certain area, is widely used as a smoothness criterion.

For the convenience of derivation, the node feature matrix \tilde{x} is split into two sub-matrices. H and Δ can be divided into four sub-matrices.

$$\tilde{x} = \begin{bmatrix} \tilde{x}_k \\ \tilde{x}_u \end{bmatrix} H = \begin{bmatrix} H_{kk} & H_{ku} \\ H_{uk} & H_{uu} \end{bmatrix} \Delta = \begin{bmatrix} \Delta_{kk} & \Delta_{ku} \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix}. \quad (3)$$

Let $\dot{x}(t) = -\nabla \ell(\tilde{x}(t))$ denotes the associated *gradient flow*, and the known features $\tilde{x}_k = \tilde{x}_k(t)$ is the boundary condition. Therefore, the solution at the missing nodes: $\tilde{x}_u = \lim_{n \rightarrow \infty} \tilde{x}_u(t)$ is the interpolation. From [8], we can get a diffusion equation:

$$\begin{bmatrix} \dot{x}_k(t) \\ \dot{x}_u(t) \end{bmatrix} = - \begin{bmatrix} 0 & 0 \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} \begin{bmatrix} \tilde{x}_k \\ \tilde{x}_u(t) \end{bmatrix} = - \begin{bmatrix} 0 \\ \Delta_{uk} \tilde{x}_k + \Delta_{uu} \tilde{x}_u(t) \end{bmatrix}. \quad (4)$$

and its solution in an iterative scheme:

$$\tilde{x}^{(k+1)} = \begin{bmatrix} \mathbf{I} & 0 \\ \Theta_{uk} & \Theta_{uu} \end{bmatrix} \tilde{x}^{(k)}. \quad (5)$$

Algorithm 1 SGHFP

```

1: Input: feature vector  $x$ , graph structure  $G$ , train epochs  $T$ 
2: for  $x = 1 \rightarrow T$  do
3:   construct  $G_f$  and  $G_{pl}$  by  $x$  and  $G$ 
4:    $\Theta \leftarrow fuse(G_f, G_{pl})$ 
5:    $x \leftarrow \Theta x$  ▷ Propagate features
6:    $x_k \leftarrow y_k$  ▷ Reset known features
7: end for

```

The update procedure in equation 4 is equivalent to the following two steps. First, the feature vector x is multiplied by the propagation matrix Θ . Second, the known features are reset to their original true values. This update procedure provides an iterative algorithm to reconstruct the unknown node features, as shown in Algorithm 1. Specifically, the feature hypergraph G_f and pseudo-label hypergraph G_{pl} are fused to generate a propagation matrix Θ (in the first iteration, only feature hypergraph G_f is used). At each iteration, features are propagated among nodes by the propagation matrix Θ . After that, we clamp the known features by resetting them to their original true values.

3. EXPERIMENT

3.1. Datasets

We fed the reconstructed node features into many downstream semi-supervised node classification tasks, their classification performance can reflect the accuracy of feature reconstruction intuitively. In this paper, we evaluate our proposed SGHFP on four benchmark datasets including Cora, CiteSeer, PubMed [17], and Photo (Amazon) [18].

Table 1. Dataset statistics.

Dataset	Nodes	Edges	Features	Classes
Cora	2485	5069	1433	7
Citeseer	2120	3679	3703	6
PubMed	19717	44324	500	3
Photo	7487	119043	745	8

3.2. Experimental Setup

In all experiments, We randomly select 20 nodes per class as the training set, 1500 nodes for validation, and the rest for testing. In this paper, we use a two-layers GCN with a dropout rate of 0.5 as the downstream classifier. The Adam optimizer [19] with a learning rate of 0.005 is introduced to optimize the model parameters. Empirically, hypergraph feature propagation diffuses the features over 50 iterations. For all the baselines, the hyperparameters are the same as mentioned in the respective papers [8, 13, 15, 20, 21].

Table 2. Classification performance of our proposed SGHFP and FP under different rates of missing features. The best results are highlighted.

Dataset	Method	Missing Node Features Rate			
		0	50%	90%	99%
Cora	FP	80.39%	79.7% (-0.86%)	79.77% (-0.77%)	78.22% (-2.70%)
	SGHFP	81.23%	80.9% (-0.41%)	80.41% (-1.01%)	79.41% (-2.25%)
Citeseer	FP	67.48%	65.74% (-2.57%)	65.57% (-2.82%)	65.4% (-3.08%)
	SGHFP	67.45%	66.86% (-0.87%)	66.76% (-1.02%)	66.5% (-1.41%)
PubMed	FP	77.36%	76.68% (-0.89%)	75.85% (-1.96%)	74.29% (-3.97%)
	SGHFP	77.45%	77.15% (-0.37%)	76.63% (-1.06%)	75.17% (-2.94%)
Photo	FP	91.73%	91.29% (-0.48%)	89.48% (-2.46%)	87.73% (-4.36%)
	SGHFP	92.08%	91.46% (-0.67%)	90.04% (-2.22%)	88.40% (-3.99%)

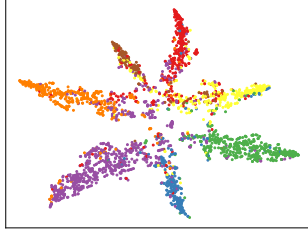
3.3. Results and Discussion

3.3.1. Comparison with State-of-the-art Methods

We report the mean classification accuracy with 10 runs for all methods. Each run has a different mask of missing features. Table 2 illustrates the classification performance of our proposed SGHFP and FP (combined with a downstream GCN model) under full features, 50%, 90%, and 99% missing node features. In addition, we also report the relative drop rate when features are partially missing compared to all features that are present. Table 2 shows that the proposed SGHFP outperforms FP under any missing rate and also has a lower

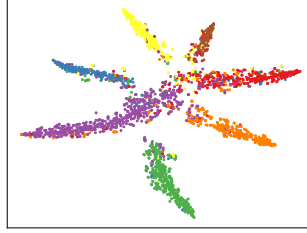
Table 3. Ablation experiments on Cora, Citeseer, PubMed, and Photo dataset with 99% missing node features.

Module	Cora				Citeseer				PubMed				Photo			
FP	✓															
FH		✓				✓				✓				✓		
SGPH			✓				✓				✓				✓	
Accuracy	78.22±0.32	78.77±0.24	79.41±0.21	65.40±0.54	65.92±0.28	66.50±0.11	74.29±0.55	74.56±0.36	75.17±0.10	87.73±0.27	88.02±0.18	88.40±0.10				



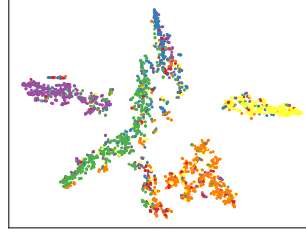
Silhouette_score:0.290

(a) FP on Cora



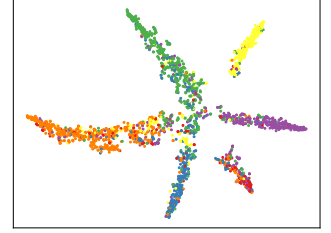
Silhouette_score:0.368

(b) SGHFP on Cora



Silhouette_score:0.354

(c) FP on Citeseer



Silhouette_score:0.438

(d) SGHFP on Citeseer

Fig. 2. The t-SNE visualization and Silhouette score of the node embeddings with reconstructing features. Each color represents one class.

relative drop rate. On average, SGHFP with 99% missing features loses only 2.67% of relative accuracy compared to the same GNNs model with no missing features.

Table 4. Classification performance of GCNMF, PaGNN, LP, GPE and SGHFP under 99% of features missing. The best results are highlighted.

Dataset	GCNMF	PaGNN	LP	GPE	SGHFP
Cora	34.54±2.07	58.03±0.57	74.68±0.36	76.33±0.26	79.41±0.21
Citeseer	30.65±1.12	46.02±0.58	64.60±0.40	65.87±0.37	66.50±0.11
PubMed	39.80±0.25	54.25±0.70	73.81±0.56	73.70±0.29	75.17±0.10
Photo	29.64±2.78	85.41±0.28	83.45±0.94	83.45±0.26	88.40±0.10

In this part, we compare the proposed SGHFP with many state-of-the-art GNNs methods when 99% of the node features are missing. We additionally compare to feature-agnostic baselines: Label Propagation (LP) [20], which only exploits the structure information of graphs by propagating labels iteratively, and Graph Positional Encodings (GPE) [21], which treats the first k eigenvector matrices of the Laplacian as node features. Table 4 shows that our proposed SGHFP outperforms the existing methods on all experimental datasets. For example, GCNMF and PaGNN have a large drop in relative accuracy when the feature missing rate is high. In comparison, SGHFP has only a 2.67% drop.

3.3.2. Ablation experiments

In this part, we investigate the impact of the self-supervised guided pseudo-label hypergraph (SGPH) and feature hypergraph (FH) module for feature reconstruction on Cora, Citeseer, PubMed, and Photo datasets. Table 3 shows that our proposed SGPH and FH modules all improve the classification performance of downstream tasks with missing node fea-

tures. For example, on the Cora dataset, FP with the FH module obtains gains of 0.55%. FP with SGPH and FH module (SGHFP) achieve 1.19% improvements in comparison to FP.

3.3.3. t-SNE visualization

To better demonstrate that the proposed SGHFP can better reconstruct the missing features, we use t-SNE and Silhouette scores to visualize the embedding of graphs with reconstructed features in 2-D space on Cora and Citeseer datasets. Nodes in the same class are expected to be clustered together and have higher Silhouette scores. Fig.2 shows that our proposed SGHFP can separate different categories and the nodes in the same class are clustered more compactly. All experiments are performed with a 90% missing rate of node features.

4. CONCLUSION

In this paper, we present a novel approach for semi-supervised classification with missing node features. The feature hypergraph and pseudo-label hypergraph are constructed to describe the local geometric distribution between data in turns. Then, a fused hypergraph generated by an effective strategy is further applied to the feature propagation model for reconstructing the missing features. Experimental results on several datasets demonstrate that the proposed SGHFP is useful to reconstruct the missing features and also outperforms many existing state-of-the-art methods. While SGHFP is designed for homogeneous graphs, it does not perform well on heterogeneous graphs. Learning heterogeneous feature-missing graphs with a more general learnable diffusion matrix could also be an interesting problem.

5. REFERENCES

- [1] Ping Hu, Virginia Bordinon, Stefan Vlaski, and Ali H Saye, "Optimal combination policies for adaptive social learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 5842–5846.
- [2] Artun Bayer, Arindam Chowdhury, and Santiago Segarra, "Label propagation across graphs: Node classification using graph neural tangent kernels," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 5483–5487.
- [3] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li, "Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *Proceedings of the International Conference on Machine Learning*, 2022, pp. 11906–11917.
- [4] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò, "3d infomax improves gnns for molecular property prediction," in *Proceedings of the International Conference on Machine Learning*, 2022, pp. 20479–20502.
- [5] Ashish Tiwari, Richeek Das, and Shanmuganathan Raman, "Exploring deeper graph convolutions for semi-supervised node classification," in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 5463–5467.
- [6] Sichao Fu, Senlin Wang, Weifeng Liu, Baodi Liu, Bin Zhou, Xinhua You, Qinmu Peng, and Xiao-Yuan Jing, "Adaptive graph convolutional collaboration networks for semi-supervised classification," *Information Sciences*, vol. 611, pp. 262–276, 2022.
- [7] Sichao Fu, Weifeng Liu, Kai Zhang, Yicong Zhou, and Dapeng Tao, "Semi-supervised classification by graph p-laplacian convolutional networks," *Information Sciences*, vol. 560, pp. 92–106, 2021.
- [8] Emanuele Rossi, Henry Kenlay, Maria I Gorinova, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael Bronstein, "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features," *arXiv preprint arXiv:2111.12128*, 2022.
- [9] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, Jianping Yin, and Wen Gao, "Multiple kernel k -means with incomplete kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1191–1204, 2019.
- [10] Jinsung Yoon, James Jordon, and Mihaela Schaar, "Gain: Missing data imputation using generative adversarial nets," in *Proceedings of the International Conference on Machine Learning*, 2018, pp. 5689–5698.
- [11] Sunil K Narang, Akshay Gadde, and Antonio Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 5445–5449.
- [12] Xu Chen, Siheng Chen, Jiangchao Yao, Huangjie Zheng, Ya Zhang, and Ivor W Tsang, "Learning on attribute-missing graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [13] Hibiki Taguchi, Xin Liu, and Tsuyoshi Murata, "Graph convolutional networks for graphs containing missing features," *Future Generation Computer Systems*, vol. 117, pp. 155–168, 2021.
- [14] Max Welling and Thomas N Kipf, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, 2017.
- [15] Bo Jiang and Ziyang Zhang, "Incomplete graph representation and learning via partial graph neural networks," *arXiv preprint arXiv:2003.10130*, 2022.
- [16] Hongwei Wang and Jure Leskovec, "Unifying graph convolutional neural networks and label propagation," *arXiv preprint arXiv:2002.06755*, 2020.
- [17] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–93, 2008.
- [18] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia, "Microsoft academic graph: When experts are not enough," *Quantitative Science Studies*, vol. 1, pp. 396–413, 2020.
- [19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.
- [20] Zhu Xiaojin and Ghahramani Zoubin, "Learning from labeled and unlabeled data with label propagation," *Technical Report CMU-CALD-02-107*, Carnegie Mellon University, 2002.
- [21] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson, "Benchmarking graph neural networks," *arXiv preprint arXiv:2003.00982*, 2020.