GraFN: Semi-Supervised Node Classification on Graph with Few Labels via Non-Parametric Distribution Assignment

Junseok Lee[1], Yunhak Oh[1], Yeonjun In[1], Namkyeong Lee[1], Dongmin Hyun[2], Chanyoung Park[1,3*]
[1] Dept. of Industrial and Systems Engineering, KAIST, Daejeon, Republic of Korea
[2] Institute of Artificial Intelligence, POSTECH, Pohang, Republic of Korea
[3] Graduate School of Artificial Intelligence, KAIST, Daejeon, Republic of Korea
{junseoklee, yunhak.oh, yeonjun.in, namkyeong96}@kaist.ac.kr, dm.hyun@postech.ac.kr, cy.park@kaist.ac.kr

ABSTRACT

Despite the success of Graph Neural Networks (GNNs) on various applications, GNNs encounter significant performance degradation when the amount of supervision signals, i.e., number of labeled nodes, is limited, which is expected as GNNs are trained solely based on the supervision obtained from the labeled nodes. On the other hand, recent self-supervised learning paradigm aims to train GNNs by solving pretext tasks that do not require any labeled nodes, and it has shown to even outperform GNNs trained with few labeled nodes. However, a major drawback of self-supervised methods is that they fall short of learning class discriminative node representations since no labeled information is utilized during training. To this end, we propose a novel semi-supervised method for graphs, GraFN, that leverages few labeled nodes to ensure nodes that belong to the same class to be grouped together, thereby achieving the best of both worlds of semi-supervised and self-supervised methods. Specifically, GraFN randomly samples support nodes from labeled nodes and anchor nodes from the entire graph. Then, it minimizes the difference between two predicted class distributions that are non-parametrically assigned by anchor-supports similarity from two differently augmented graphs. We experimentally show that GraFN surpasses both the semi-supervised and self-supervised methods in terms of node classification on real-world graphs. The source code for GraFN is available at https://github.com/LJS-Student/GraFN.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Semi-supervised learning settings**.

KEYWORDS

Semi-Supervised, Graph Neural Networks, Graph Representation Learning, Few Label

*Corresponding author.

SIGIR '22, July 11-15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

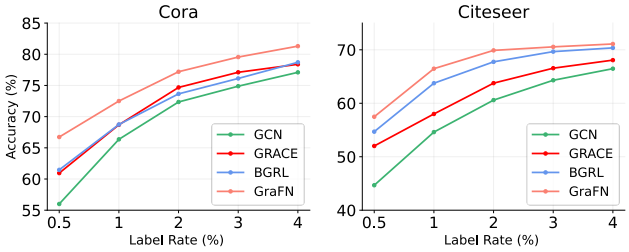


Figure 1: Performance of GCN, GRACE, BGRL and GraFN on Cora and Citeseer datasets over various labeled node rates.

1 INTRODUCTION

Recently, Graph Neural Networks (GNNs) are widely applied in IR applications ranging from recommender system [4, 6, 12, 23, 25], question answering [5, 28] to web search [11, 13]. Specifically, graph convolution-based methods [27, 29] incorporate rich attributes of nodes along with the structural information of graphs by recursively aggregating the neighborhood information. Despite the success, the performance of GNNs on node classification significantly degrades when only few labeled nodes are given as GNNs tend to overfit to the few labeled nodes. To make the matter worse, GNNs suffer from ineffective propagation of supervisory signals due to the well-known over-smoothing issue [9], which makes GNNs not being able to fully benefit from the given labeled nodes. Fig. 1 demonstrates a severe performance degradation of GCN as the rate of the labeled nodes decreases.

To train GNNs given limited labeled information, recent studies mainly focus on leveraging pseudo-labeling techniques. Specifically, co-training [9] uses Parwalks [26] to provide confident pseudo-labels to help train GNNs, and self-training [9] expands the label set by obtaining pseudo-labels provided by GNNs trained in advance. Moreover, M3S [18] leverages a clustering technique to filter out pseudo-labels that do not align with the clustering assignments for improving the pseudo-labeling accuracy. However, pseudo-labeling-based methods suffer from an inherent limitation originated from the incorrect pseudo-labels, which eventually incur confirmation bias [1]. To alleviate this issue, it is crucial to fully benefit from the given label information.

On the other hand, self-supervised methods for graphs [20, 31, 32] learn node representations without any requirements of labeled nodes. In particular, based on graph augmentations, contrastive learning-based methods pull positive pairs of nodes together while pushing negative ones apart, whereas consistency regularization-based methods impose GNNs to consistently output the same node

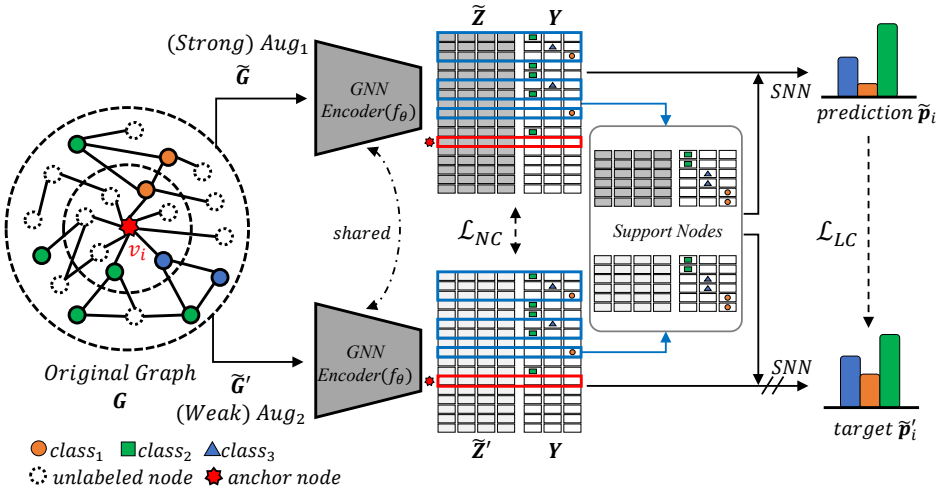


Figure 2: The overall model architecture of GraFN. Given a graph G, we generate two differently augmented views $\tilde{G}$ and $\tilde{G}'$ both of which are fed into a shared encoder f_θ to obtain node-level representation $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{Z}}'$, respectively. Then, GraFN not only minimizes the difference between these two representations obtained from differently augmented graphs, i.e. node-wise consistency ($\mathcal{L}_{\text{NC}}$), but also minimizes the difference between the predicted class distributions computed in a non-parametrically manner by using the similarity between the anchor node and support nodes, which are randomly sampled from labeled nodes, i.e., label-guided consistency ($\mathcal{L}_{\text{LC}}$).

representations over various perturbations of the given graph. Although these methods have achieved the state-of-the-art results in node classification even outperforming the supervised counterparts, they fall short of learning class discriminative node representations since no labeled information is utilized during training. As shown in Fig. 1, although recent self-supervised methods for graphs, i.e., BGRL [20] and GRACE [31], outperform GCN [7] over various rates of labeled nodes, the performance degradation is still severe as the rate decreases.

To this end, we propose a simple yet effective semi-supervised method for graphs that fully leverages a small amount of labeled nodes to learn class discriminative node representations, called GraFN. The main idea is to consistently impose the representations of nodes that belong to the same class to be grouped together on differently augmented graphs. Specifically, we randomly sample support nodes from the labeled nodes and anchor nodes from the entire graph, and non-parametrically compute two predicted class distributions from two augmented graphs based on the anchor-supports similarity. By minimizing the difference between the two class distributions, GraFN not only learns augmentation invariant parameters, but also enforces the representations of nodes that belong to the same class to be grouped together. As shown in Fig. 1, GraFN consistently outperforms both the semi-supervised and self-supervised baselines over various rates of labeled nodes, especially outperforming when the number of labeled nodes is smaller, which demonstrates the robustness of GraFN.

Notations. Let $G = (V, E)$ denote a graph, where V is the set of $|V| = N$ nodes and E is the set of edges between the nodes. The adjacency matrix is defined by $\mathbf{A} \in \mathbb{R}^{N \times N}$ with each element $\mathbf{A}_{ij} = 1$ indicating the existence of an edge between nodes v_i and v_j, otherwise $\mathbf{A}_{ij} = 0$. The node attributes are denoted by $\mathbf{X} \in \mathbb{R}^{N \times F}$, where F is the number of features of each node. Additionally, the label matrix is denoted by $\mathbf{Y} \in \mathbb{R}^{N \times C}$, where C is the number of classes, and each row, i.e., $\mathbf{Y}_i \in \mathbb{R}^C$, is the one-hot label vector for node v_i. We denote V_L and V_U as the set of labeled and unlabeled nodes, respectively. Our goal is to accurately predict the labels of nodes that belong to V_U given few labeled nodes, i.e., $|V_L| \ll |V_U|$.

2 PROPOSED METHOD : GraFN

1) Graph Augmentations and Encoding. Given a graph, we first generate two graph views by applying stochastic graph augmentations, which randomly mask node features and drop partial edges. Two differently augmented views are denoted by $\tilde{G} = (\tilde{\mathbf{A}}, \tilde{\mathbf{X}})$ and $\tilde{G}' = (\tilde{\mathbf{A}}', \tilde{\mathbf{X}}')$. Each augmented view is fed into a *shared* GNN encoder, $f_\theta : \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times F} \rightarrow \mathbb{R}^{N \times D}$, to obtain low dimensional node-level representations $f_\theta(\tilde{\mathbf{A}}, \tilde{\mathbf{X}}) = \tilde{\mathbf{Z}} \in \mathbb{R}^{N \times D}$, and $f_\theta(\tilde{\mathbf{A}}', \tilde{\mathbf{X}}') = \tilde{\mathbf{Z}}' \in \mathbb{R}^{N \times D}$. Note that we adopt GCN as the backbone of the GNN encoder.

2) Node-wise Consistency Regularization. Then, to learn augmentation invariant node representations, we minimize the difference, i.e., cosine distance, between the representations obtained from the two differently augmented graphs in a node-wise manner:

$$\mathcal{L}_{\text{NC}} = -\frac{1}{N} \sum_{i=1}^{N} \frac{\tilde{\mathbf{Z}}_i \cdot \tilde{\mathbf{Z}}'_i}{\|\tilde{\mathbf{Z}}_i\| \|\tilde{\mathbf{Z}}'_i\|} \tag{1}$$

Note that the above loss can be considered as a simplified version of the self-supervised loss proposed in BGRL. The major difference is that BGRL involves two separate encoders, where one encoder is updated by minimizing the distance between the node representations obtained from the two views, while the other one is updated by the exponential moving average of the parameters of the other

Table 1: Test Accuracy on semi-supervised node classification.

Methods	Cora			Citeseer			Pubmed			Am. Comp			Am. Photos		
Label Rate	0.5%	1%	2%	0.5%	1%	2%	0.03%	0.06%	0.1%	0.15%	0.2%	0.25%	0.15%	0.2%	0.25%
MLP	31.24	37.74	44.53	32.07	43.07	46.11	52.50	55.80	61.22	40.30	42.22	49.98	29.76	31.64	38.55
LP	50.77	58.28	64.43	31.15	37.95	41.71	50.93	55.83	62.14	60.46	65.90	68.79	63.67	66.38	70.40
GCN	56.00	66.36	72.35	44.67	54.61	60.59	59.28	64.00	73.74	62.71	66.81	71.75	66.70	70.72	75.74
GAT	58.57	67.75	72.74	48.70	58.73	62.71	63.15	64.11	73.19	66.17	70.18	72.82	73.29	74.46	80.12
SGC	49.19	63.60	69.56	44.02	55.89	63.61	58.58	62.50	71.90	59.69	64.24	68.29	55.96	61.64	69.69
APPNP	62.02	71.45	76.89	41.79	54.70	62.86	63.15	64.11	73.19	68.53	72.47	74.27	75.54	78.49	82.75
GRAND	54.51	70.92	74.90	46.76	58.40	65.31	55.87	61.25	72.42	68.00	72.71	75.77	73.80	75.83	82.33
GLP	56.94	68.28	72.97	41.53	54.84	63.08	56.70	60.83	73.46	62.97	68.56	70.70	63.18	67.96	75.19
IGCN	58.81	70.10	74.34	43.28	57.00	64.62	57.50	62.06	73.13	65.48	70.05	71.03	71.27	73.28	77.93
CGPN	64.21	70.54	72.97	53.90	63.70	65.15	64.55	67.58	71.42	65.37	67.98	70.77	74.14	76.89	81.57
GRACE	60.95	68.69	74.68	52.01	58.00	63.76	64.86	68.35	**75.92**	65.25	67.79	71.79	70.19	71.89	77.32
BGRL	61.74	68.74	73.65	54.69	63.75	67.75	65.77	**68.86**	75.91	68.80	73.04	75.11	74.27	78.25	83.12
Co-training	62.75	68.72	74.05	43.76	54.75	61.13	63.01	68.15	74.24	67.06	71.62	71.34	72.85	74.65	79.92
Self-training	57.28	70.73	75.40	46.26	60.36	66.47	57.34	65.13	72.86	61.32	65.95	68.66	61.92	65.24	71.34
M3S	64.46	**72.93**	76.41	55.07	65.74	67.64	61.53	64.60	73.18	61.51	66.30	68.10	63.93	67.62	73.39
GraFN	**66.73**	72.50	**77.20**	**57.48**	**66.47**	**69.89**	**65.91**	68.41	75.74	**71.73**	**74.26**	**77.37**	**79.25**	**80.87**	**85.36**

encoder to prevent the collapsing of node representations to trivial solutions. On the other hand, GraFN trains only one *shared* encoder, and we find that the supervisory signals incorporated in the next step help avoid the collapse of representations.

3) Label-guided Consistency Regularization. Although the above self-supervised loss has been shown to be effective, the learned node representations are not class discriminative because the node label information is not involved in the training process. We argue that unlabeled nodes can be grouped together according to their classes by enforcing them to be consistently close with a certain class of labeled nodes on differently augmented graphs. Hence, we compute the similarity between few labeled nodes and all the nodes in the two augmented graphs, and maximize the consistency between the two similarity-based class assignments, expecting that this would help nodes that belong to the same class to be grouped together. More precisely, we first randomly sample b labeled nodes per class to construct the support set $\mathcal{S}$, and let $\mathbf{Z}^{\mathcal{S}} \in \mathbb{R}^{(b\times C)\times D}$ denote $(b\times C)$ support node representations[1]. Then, for each anchor node $v_i \in V$, we compute the similarity distribution, i.e., predicted class distribution, $\mathbf{p}_i \in \mathbb{R}^C$ using anchor-supports similarity in a non-parametric manner by applying Soft Nearest Neighbors(SNN) strategy [2, 15] as follows:

$$\mathbf{p}_i = \sum_{(\mathbf{Z}_j^{\mathcal{S}}, \mathbf{Y}_j^{\mathcal{S}}) \in \mathbf{Z}^{\mathcal{S}}} \frac{\exp\left(\mathsf{sim}(\mathbf{Z}_i, \mathbf{Z}_j^{\mathcal{S}})/\tau\right)}{\sum_{\mathbf{Z}_k^{\mathcal{S}} \in \mathbf{Z}^{\mathcal{S}}} \exp\left(\mathsf{sim}(\mathbf{Z}_i, \mathbf{Z}_k^{\mathcal{S}})/\tau\right)} \cdot \mathbf{Y}_j^{\mathcal{S}} \tag{2}$$

where $\mathsf{sim}(\cdot,\cdot)$ computes the cosine similarity between two vectors, and τ is a temperature hyperparameter. $\mathbf{p}_i$ can be considered as the soft pseudo-label vector for node v_i since it is derived based on the labeled nodes in $\mathcal{S}$. Having defined the predicted class distribution as in Eqn. 2, we compute the predicted class distributions for each node $v_i \in V$, i.e., $\tilde{\mathbf{p}}_i$ and $\tilde{\mathbf{p}}_i^{'}$ each of which is obtained from $\tilde{G}$ and $\tilde{G}^{'}$, respectively, where the former is considered as the prediction distribution, and the latter is considered as the target distribution. Then, we minimize the cross-entropy between them:

$$\frac{1}{|V_U|} \sum_{v_i \in V_U} H(\tilde{\mathbf{p}}_i^{'}, \tilde{\mathbf{p}}_i) + \frac{1}{|V_L|} \sum_{v_i \in V_L} H(\mathbf{Y}_i, \tilde{\mathbf{p}}_i) \tag{3}$$

where $H(\mathbf{y}, \hat{\mathbf{y}})$ is the cross-entropy between the target $\mathbf{y}$ and the prediction $\hat{\mathbf{y}}$. Note that for each labeled node, instead of computing the predicted class distribution as in Eqn. 2, we simply assign its one-hot label vector, i.e., $\tilde{\mathbf{p}}_i^{'} = \mathbf{Y}_i, \forall v_i \in V_L$, to fully leverage the label information. However, naively minimizing the above loss would incur confirmation bias [1] due to inaccurate $\tilde{\mathbf{p}}_i^{'}$ computed for unlabeled nodes (i.e., V_U), which is detrimental to the performance of pseudo-labeling-based semi-supervised methods. To this end, we introduce a confidence-based label-guided consistency regularization:

$$\mathcal{L}_{\text{LC}} = \frac{1}{|V_{\text{conf}}|} \sum_{v_i \in V_{\text{conf}}} H(\tilde{\mathbf{p}}_i^{'}, \tilde{\mathbf{p}}_i) + \frac{1}{|V_L|} \sum_{v_i \in V_L} H(\mathbf{Y}_i, \tilde{\mathbf{p}}_i) \tag{4}$$

where $V_{\text{conf}} = \{v_i | \mathbb{1}(\max(\tilde{\mathbf{p}}_i^{'}) > \nu) = 1, \forall v_i \in V_U\}$ is the set of nodes with confident predictions, ν is the threshold for determining whether a node has confident prediction, and $\mathbb{1}(\cdot)$ is an indicator function. $\tilde{\mathbf{p}}_i^{'}$ is considered to be confident if its maximum element is greater than ν. We argue that setting ν to a high value helps alleviate confirmation bias [1, 17] by enforcing only high-quality target distribution $\tilde{\mathbf{p}}_i^{'}$ to be able to contribute to Eqn. 4. It is important to note that we apply a relatively weak augmentation for graph $\tilde{G}^{'} = (\tilde{\mathbf{A}}^{'}, \tilde{\mathbf{X}}^{'})$ that is used to compute $\tilde{\mathbf{p}}_i^{'}$, because aggressive augmentations (e.g., dropping more than half of edges) drastically change the semantics of a graph, which may eventually incur inaccurate $\tilde{\mathbf{p}}_i^{'}$. A further benefit of the label-guided consistency regularization defined in Eqn. 4 is that since the class distributions are computed regardless of the structural information of graphs, the label information can be effectively propagated to distant nodes, whereas existing GNNs suffer from ineffective propagation incurred by the over-smoothing issue [9]. Moreover, to break the symmetry

[1]To make the best use of labeled nodes in the support set S, we fix b to the number of labeled nodes that belong to the class with the fewest nodes, i.e., minority class.

Table 2: Statistics for datasets used for experiments.

	# Nodes	# Edges	# Features	# Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
Am. Comp.	13,752	245,861	767	10
Am. Photos	7,650	119,081	745	8

of the model architecture thereby preventing the collapsing of node representations to trivial solutions, we stop gradient for the target distribution (i.e., $\tilde{\mathbf{p}}'$), and only update the parameters associated with the prediction distribution (i.e., $\tilde{\mathbf{p}}$).

4) Final Objective. Finally, we combine $\mathcal{L}_{\text{NC}}$ and $\mathcal{L}_{\text{LC}}$ with coefficients λ_1 and λ_2 to compute the final objective function as follows:

$$\mathcal{L}_{\text{Training}} = \lambda_1 \mathcal{L}_{\text{NC}} + \lambda_2 \mathcal{L}_{\text{LC}} + \mathcal{L}_{\text{sup}} \tag{5}$$

We add the cross-entropy loss, i.e., $\mathcal{L}_{\text{sup}}$, defined over a set of labeled nodes V_L. The overall pipeline of GraFN is shown in Fig. 2.

3 EXPERIMENTS

Datasets. To verify the effectiveness of GraFN, we conduct extensive experiments on five widely used datasets (Table 2) including three citation networks (Cora, Citeseer, Pubmed) [16] and two co-purchase networks (Amazon Computers, Amazon Photos) [14].

Baselines. We compare GraFN with fifteen baseline methods. (i) *Two conventional methods*: **MLP** and **LP** [30]. (ii) *Five graph convolution-based methods*: **GCN** [7] and **GAT** [21]. **SGC** [24] simplifies GCN by removing repeated feature transformations and nonlinearities. **APPNP** [8] and **GRAND** [3] alleviate the limited receptive field issue of existing message passing models. (iii) *Two self-supervised methods*: **GRACE** [31] and **BGRL** [20] maximize the agreement of the representations of the same nodes from two augmented views[2]. (iv) *Three label-efficient methods*: **GLP** [10] and **IGCN** [10] apply a low-pass graph filter to message propagation to achieve label efficiency. **CGPN** [22] leverages Graph Poisson Network to effectively spread limited labels to the whole graph, and also utilizes contrastive loss to leverage information on unlabeled nodes. (v) *Three pseudo-labeling-based methods*: **Co-training** [9], **Self-training** [9] and **M3S** [18].

Evaluation Protocol. We randomly create 20 splits to evaluate the effectiveness of GraFN on practical few-label setting. For citation networks, we closely follow the evaluation protocol of [9, 19]. For co-purchase networks, we evaluate models with {0.15%, 0.2%, 0.25%} training label rates to match with the average number of labeled nodes per class of the citation networks. The remaining nodes are split into 1:9, and each split is used for validation and test, respectively. We report the averaged test accuracy when the validation accuracy is the highest. For baseline methods including GraFN, we search hidden dimension, learning rate, weight decay, and dropout ratio in {32, 64, 128}, {0.1, 0.01, 0.05, 0.001, 0.005}, {1e-2, 1e-3, 1e-4, 5e-4}, and {0, 0.3, 0.5, 0.8}, respectively, while other hyperparameter configurations are taken from each work. Additionally, we conduct an extensive grid search for unreported hyperparameters for fair comparisons. As mentioned in Sec. 2, for GraFN, we apply relatively weaker augmentations on $\tilde{G}'$ compared with $\tilde{G}$, and thus we search the augmentation hyperparameters of node feature masking and partial edge dropping in {0.2, 0.3, 0.4} for $\tilde{G}'$, and {0.5, 0.6} for $\tilde{G}$. Moreover, the temperature τ is fixed to 0.1, the threshold ν is searched in {0.8, 0.9}, and the balance hyperparameters, i.e., λ_1 and λ_2, are searched in {0.5, 1.0, 2.0}.

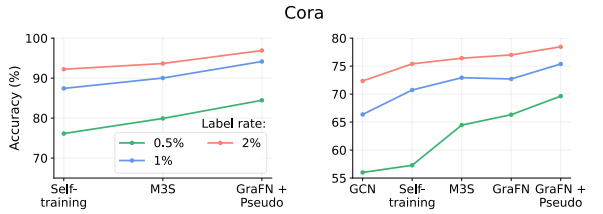
[2]For fair comparisons with GraFN, we extend them to semi-supervised setting by adding the conventional supervised loss, i.e., cross-entropy loss.

Table 3: Performance on similarity search. (Sim@K: Average ratio among K-NNs sharing the same label as the query node.)

		GRACE	BGRL	GraFN
Cora	Sim@5	0.8146	0.8047	**0.8222**
	Sim@10	0.7947	0.7823	**0.7984**
Citeseer	Sim@5	0.6407	0.6623	**0.6810**
	Sim@10	0.6147	0.6396	**0.6621**
Pubmed	Sim@5	0.7571	**0.7815**	0.7110
	Sim@10	0.7493	**0.7733**	0.7015
Am.Comp	Sim@5	0.8091	0.8335	**0.8351**
	Sim@10	0.7965	0.8211	**0.8246**
Am.Photos	Sim@5	0.8831	0.8886	**0.9004**
	Sim@10	0.8761	0.8881	**0.8937**

3.1 Performance Analysis

Table 1 shows the performance of various methods in terms of node classification over various label rates. We have the following observations: **1)** GraFN outperforms methods that use advanced GNN encoders (i.e., APPNP and GRAND), and label efficient GNNs that effectively propagate label information to distant nodes (e.g., GLP and IGCN) and leverage unlabeled nodes with contrastive loss (i.e., CGPN). We argue that the label-guided consistency regularization helps GraFN to effectively propagate the label information to distant nodes, thereby learning class discriminative representations. Note that GraFN especially outperforms other methods when the label rate is small demonstrating the robustness of GraFN (also refer to Fig. 1). **2)** Even though GraFN adopts a simple self-supervised loss with a single shared GNN-based encoder, i.e. node-wise consistency regularization, GraFN outperforms advanced self-supervised methods, i.e. GRACE and BGRL. We attribute this to the label-guided consistency loss that groups nodes that belong to the same class by leveraging the given few labeled nodes. **3)** To empirically verify the benefit of the label-consistency regularization of GraFN, we compare the similarity search results of GRACE, BGRL and GraFN in Table 3. Since GRACE and BGRL consider node labels through conventional supervised loss with advanced self-supervised loss, whereas GraFN does so through the label-guided consistency regularization with a simple self-supervised loss, the superior performance of GraFN implies the benefit of the label-guided consistency regularization of GraFN despite its simple self-supervised loss. We indeed observe that GraFN generally outperforms GRACE and BGRL, which corroborates the benefit of the label-consistency regularization for learning class discriminative node representations. **4)** We observe that the pseudo-labeling-based methods, i.e., Co-training, Self-training and M3S, generally outperform vanilla GCN, which solely relies on the given label information. On the other hand, GraFN outperforms these methods without relying on the



(a) Pseudo-label accuracy **(b) Node classification accuracy**

Figure 3: Accuracy of pseudo-labeling and node classification.

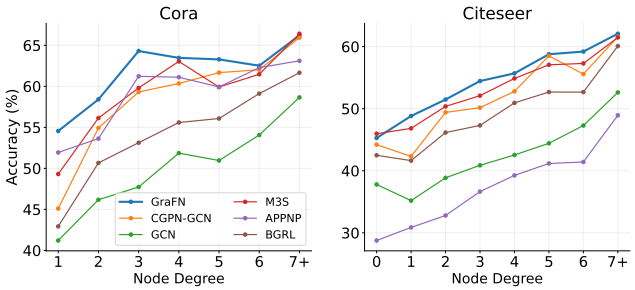


Figure 4: Node classification results on various node degrees.

pseudo-labeling techniques. This implies that the pseudo-labeling techniques should be adopted with particular care as they may introduce incorrect labels. **5)** GraFN shows relatively low performance on Pubmed dataset, which contains a small number of classes, i.e., 3 classes. Since GraFN assigns class distributions based on the similarity with labeled nodes in different classes, having more classes leads to more class discriminative node representations.

Adopting Pseudo-Labeling to GraFN. Since GraFN learns class discriminative node representations with few labels, we hypothesized that the confirmation bias [1] of the pseudo-labeling technique suffered by existing methods would be alleviated when the pseudo-labeling technique is adopted to GraFN. Fig. 3(a) indeed demonstrates that adopting the pseudo-labeling technique to GraFN gives the best pseudo-labeling accuracy, which in turn results in further improvements of GraFN in terms of node classification (Fig. 3(b)).

Performance Comparison on Different Node Degrees. In most real-world graphs, the node degrees follow a power-law distribution, which means that the majority of nodes are of low-degree. Since the training of GNNs is based on the neighborhood aggregation scheme, high-degree nodes receive more information than low-degree nodes, which eventually leads to the models overfit to high-degree nodes and underfit to low-degree nodes [19]. Fig. 4 demonstrates that the node classification accuracy is indeed highly correlated with the node degree, i.e., high-degree nodes tend to result in better classification performance. Furthermore, we observe that GraFN greatly outperforms other methods for low-degree nodes, while showing a comparable performance for high-degree nodes (i.e. degree ≥ 7). We attribute the superior performance of GraFN on low-degree nodes to the label-guided consistency regularization that promotes the supervision information to be evenly spread over the unlabeled nodes regardless of their node degree.

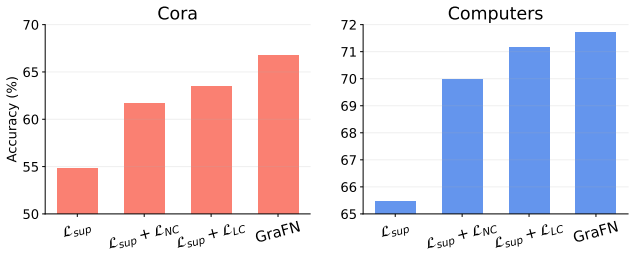


Figure 5: Ablation study on GraFN.

3.2 Ablation Studies

To evaluate each component of GraFN, we conduct ablation studies on Cora and Computers datasets on the lowest label rate, i.e. 0.5%, 0.15%, respectively. In Fig 5, we have the following observations: **1)** Only using the supervised cross-entropy loss shows poor performance because it suffers from overfitting and ineffective propagation of supervisory signal. **2)** Using both node-wise and label-guided consistency regularization (i.e., GraFN) is more beneficial than using either one of them. We argue that this is because these two losses are complementary. More precisely, using only the node-wise loss cannot fully leverage the given label information, whereas using only the label-guided regularization loss can suffer from inaccurate target distribution, which incurs confirmation bias.

4 CONCLUSIONS

In this paper, we present a novel semi-supervised method for graphs that learns class discriminative node representations with only few labeled nodes. GraFN not only exploits the self-supervised loss, i.e. node-wise consistency regularization, but also ensures nodes that belong to the same class to be grouped together by enforcing unlabeled nodes to be consistently close with a certain class of labeled nodes on differently augmented graphs. Through extensive experiments on real-world graphs, we show that GraFN outperforms existing state-of-the art methods given few labeled nodes. Moreover, it is worth noting that GraFN 1) alleviates underfitting problem of low-degree nodes by propagating label information to distant nodes, and 2) enjoys further improvements by adopting the pseudo-labeling technique.

Acknowledgements. This work was supported by the NRF grant funded by the MSIT (No.2021R1C1C1009081), and the IITP grant funded by the MSIT (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).

REFERENCES

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. 2021. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8443–8452.

[3] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. 2020. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems* 33 (2020), 22092–22103.

[4] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for

recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[5] Guangyi Hu, Chongyang Shi, Shufeng Hao, and Yu Bai. 2020. Residual-Duet Network with Tree Dependency Representation for Chinese Question-Answering Sentiment Analysis. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1725–1728.

[6] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–668.

[7] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[8] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).

[9] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.

[10] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. 2019. Label efficient semi-supervised learning via graph filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9582–9591.

[11] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1259–1268.

[12] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing gcn for recommendation. In *Proceedings of the Web Conference 2021*. 1296–1305.

[13] Kelong Mao, Xi Xiao, Jieming Zhu, Biao Lu, Ruiming Tang, and Xiuqiang He. 2020. Item Tagging for Information Retrieval: A Tripartite Graph Neural Network based Approach. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2327–2336.

[14] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[15] Ruslan Salakhutdinov and Geoff Hinton. 2007. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*. PMLR, 412–419.

[16] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[17] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems* 33 (2020), 596–608.

[18] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. 2020. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5892–5899.

[19] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*. 1435–1444.

[20] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. 2022. Large-Scale Representation Learning on Graphs via Bootstrapping. In *International Conference on Learning Representations*.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[22] Sheng Wan, Yibing Zhan, Liu Liu, Baosheng Yu, Shirui Pan, and Chen Gong. 2021. Contrastive Graph Poisson Networks: Semi-Supervised Learning with Extremely Limited Labels. *Advances in Neural Information Processing Systems* 34 (2021).

[23] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[24] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[25] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[26] Xiao-Ming Wu, Zhenguo Li, Anthony So, John Wright, and Shih-Fu Chang. 2012. Learning with partially absorbing random walks. *Advances in neural information processing systems* 25 (2012).

[27] Bingbing Xu, Junjie Huang, Liang Hou, Huawei Shen, Jinhua Gao, and Xueqi Cheng. 2020. Label-consistency based graph neural networks for semi-supervised node classification. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1897–1900.

[28] Wenxuan Zhang, Yang Deng, and Wai Lam. 2020. Answer ranking for product-related questions via multiple semantic relations modeling. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 569–578.

[29] Yunxiang Zhao, Jianzhong Qi, Qingwei Liu, and Rui Zhang. 2021. Wgcn: Graph convolutional networks with weighted structural features. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 624–633.

[30] Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. (2002).

[31] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).

[32] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*. 2069–2080.