DADN: Data Augmentation-driven Dual Network Framework for Node Classification

1st Dengfeng Liu
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
liudengfeng21@nudt.edu.cn

2nd Zhiqiang Pan
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
panzhiqiang@nudt.edu.cn

3rd Siyuan Wang
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
wangsiyuan21@nudt.edu.cn

4th Peihong Li
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
lipeihong21@nudt.edu.cn

5th Shixian Liu
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
lsx@nudt.edu.cn

6th Yanying Mao *
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
maoyanying@nudt.edu.cn

2023 2nd International Conference on Automation, Robotics and Computer Engineering (ICARCE) | 979-8-3503-0834-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICARCE59252.2024.10492585

***Abstract*—Node classification is an important task in graph neural networks (GNNs), aiming to predict the category of nodes in the graph according to their neighbours and their own characteristics. Current methods mainly focus on one level of data augmentation, ignoring huge potential in other underlying levels. Furthermore, existing models cannot fully exploit the information contained in the existing data, which undermines the performance of node classification. In order to solve the above problems, we propose a novel Data Augmentation-driven Dual Network (DADN) framework. Specifically, we first design a composite data augmentation module that comprehensively considers the local neighborhood level, the node level, and the edge level to augment the original features. Then, we copy the original features and the augmented features into the dual network framework with dropout for model training to capture the innermost features. Finally, we experimentally evaluate our method on three real-world public datasets, and the results show that our method outperforms the baseline models in the node classification task, except for individual metrics.**

Keywords—Data Augmentation, Dual Network, Node Classification

I. INTRODUCTION

As a comprehensive learning approach to process graphs, graph neural networks (GNNs) acquire the representation of target node through the amalgamation of structural and attribute features from neighbouring nodes [1]. GNNs find extensive application in fields such as social networks [2].

Recently, there have been many data augmentation methods for GNNs. Despite some progress, most of these methods focus on merely one level of data augmentation, such as node-level data augmentation, edge-level data augmentation, and so on [3]. They still cannot fully mine the existing graph information by only augmenting the original data from one perspective. In addition, the current GNNs frameworks based on representation learning struggle with information extraction in node classification, which indicates they are limited to superficial information and struggle to harness the full potential of the existing graph data.

Postgraduate Scientific Research Innovation Project of Hunan Province No.CX20220042.

Therefore, we propose a Data Augmentation-driven Dual Network (DADN) framework for node classification. Specifically, we first design a composite data augmentation module that considers three-level features, including local neighborhoods, nodes, and edges: (1) we apply the conditional variational autoencoder (CVAE) generative model to learn the distribution of neighboring node features based on the central node features and generate local augmentation features; (2) we employ DropNode and random propagation to output node augmentation features; (3) we utilize DropEdge to randomly delete edges. Then, we copy the augmented and the original features, and input them into classifiers with dropout respectively. Moreover, we use KL divergence loss to minimize the distribution between the two sub-classifiers during training.

We evaluate our approach on three publicly available datasets. The experimental results show that DADN outperforms the best baseline model in terms of Precision, Recall, F1, and Accuracy, except for Recall on the Pubmed dataset. This demonstrates the effectiveness of DADN.

This study's primary contributions can be condensed as follows.

(1) We propose a composite data augmentation module that considers local neighborhoods, nodes, and edges, which effectively extends the original data.

(2) We design a dual network representation learning framework that fully learns from the training data and improves the robustness of the model.

(3) Extensive experiments on three public network datasets verified that our approach, DADN, provides distinct performance improvements over baselines in terms of four evaluation metrics.

II. RELATED WORK

Data augmentation originated from computer vision, aiming to improve the robustness and generalization ability of models by generating additional data, and has achieved significant effects in tasks such as image classification [4]. Subsequently, data augmentation techniques have been extended to other fields such as natural language processing [5]. In recent years, with the enormous potential demonstrated

by GNNs in processing graph-structured data, applying data augmentation techniques to GNNs has become a hot research direction [1]. Graph data augmentation (GDA) can be divided into node-level and edge-level augmentation, depending on the task level [6].

Node-level data augmentation tasks such as GRAND [7] propose the joint use of DropNode and random propagation to generate multiple augmented graphs and consistency loss, effectively alleviating the over-smoothing problem. You et al. [8] propose the NodeDropping technique, which aims to delete a portion of nodes. NodeDropping is similar to DropNode, but focuses on self-supervised graph representation learning. Additionally, adversarial training-based methods have been used to modify node features and improve model robustness [9]. These methods achieve data augmentation by deleting nodes or modifying node features.

Edge-level data augmentation tasks such as DropEdge [10] enhance model generalization and mitigate the over-smoothing problem by randomly deleting edges during training. Subsequent works propose edge augmentation methods considering additional information preservation or employing information filtering. For example, NeuralSparse [11] selectively deletes task-irrelevant edges using a graph sparsification model; PTDNet [12] further incorporates nuclear norm regularization to impose low-rank constraints. TADropEdge [13] selects to delete edges with minimal connectivity impact based on graph spectral analysis. These methods perform data augmentation by deleting or preserving edges.

In summary, although existing GDA research has made certain progress, these GDA methods mainly focus on data augmentation at one level, which makes it difficult to fully exploit the existing graph-structured data information.

III. Approach

Assume $G=(V,E)$ is a graph with S nodes, where $V=\{v_1,v_2,...v_S\}$ represents the set of nodes and $E\subseteq S\times S$ represents the set of edges. The adjacency matrix is denoted as $A\in\{0,1\}^{S\times S}$, with $A_{i,j}=0$ indicating no edge between node v_i and node v_j, otherwise $A_{i,j}=1$. Let $F\in R^{S\times D}$ represent the node feature matrix, where D is the node feature dimension, and F_i represents the feature of node v_j. $l_i\in L$ represents the label of node v_j. The goal of node classification is to learn node representations and apply them to downstream classification tasks, predicting the category for each node within the graph.

The proposed Data Augmentation-Driven Dual Network (DADN) method consists of two main parts: Composite Data Augmentation (Section III-A) and Dual Network Prediction (Section III-B). Fig. 1 displays the comprehensive structure of DADN. Specifically, given the feature matrix F_G of the graph G, we first generate augmented features using the local neighborhood data augmentation module. Then, we apply the DropNode and DropEdge operations on both the original and augmented features, followed by random propagation to construct augmented features. This process is repeated N times to obtain N augmented features. After duplicating the above and original features, they are respectively input into two identical classifiers that incorporate dropout, and the final predicted values are outputted.

A. Composite Data Augmentation

Local Neighborhood Data Augmentation. This sub-module utilizes a conditional variational autoencoder (CVAE) to learn the conditional distribution of the features of neighboring nodes given the feature of a central node. The design of the model is depicted in Fig. 2.

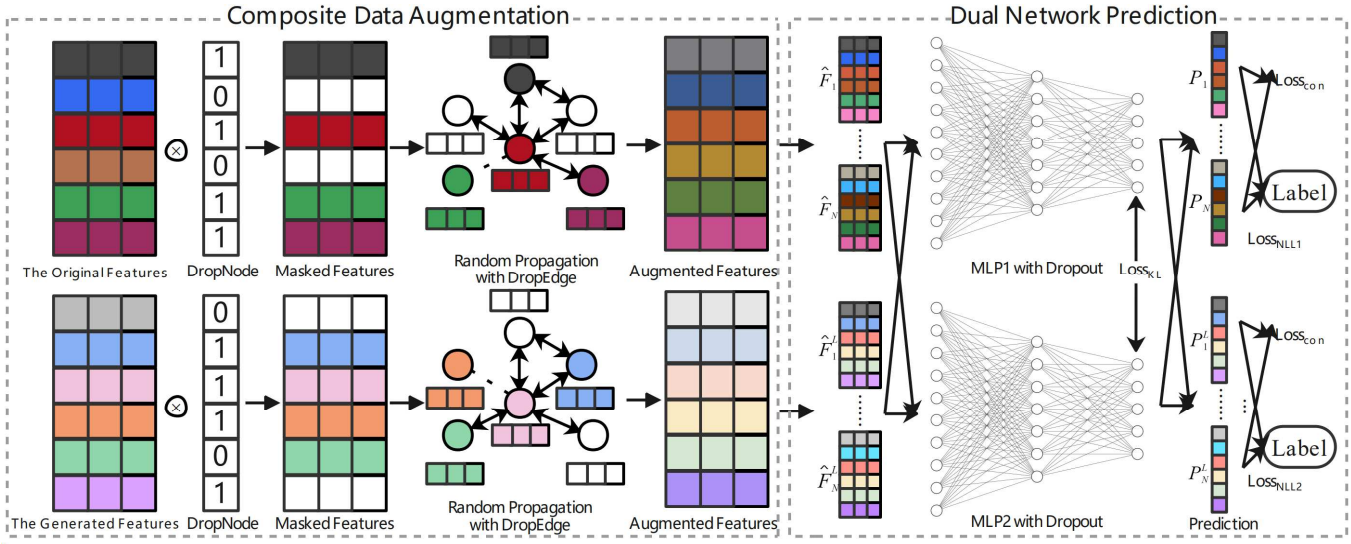

Fig. 1. Overall schematic diagram of data augmentation-driven dual network framework.

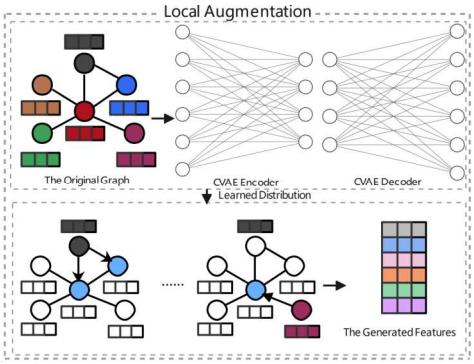

Fig. 2. Schematic diagram of local neighborhood data augmentation.

Since the features F_j of neighboring node v_j are correlated with the features F_i of the central node v_i, we treat the features F_i as a condition. Based on the theory proposed by Sohn [9], let z be the latent variable. When $z \sim p_\theta\left(z|\ F_i\right)$ and $F_j \sim p_\theta\left(F|\ F_i, z_i\right)$ are satisfied, z is generated by $p_\theta(z \mid F_i)$ and F_j is generated by $p_\theta(F \mid F_i, z)$. Assume that ψ is the variational parameter, θ denotes the generation parameter and N is the number of nodes connected to the center node v_i. Suppose $\alpha = q_\psi\left(z \mid F_j, F\right)$, $\beta = p_\theta\left(F_j|\ F_i, z^{(l)}\right)$. The evidence lower bound (ELBO) can be expressed as (1).

$$L\left(F_j, F_i; \theta, \psi\right) = -KL\left(\alpha p_\theta\left(z|\ F_i\right)\right) + \frac{1}{N}\sum_{n=1}^{N}\log\beta, \quad (1)$$

where $z^{(l)} = g_\psi\left(F_i, F_j, \tau^{(n)}\right)$, $\tau^{(n)} \sim \mathrm{N}(0, \mathrm{I})$. During model training, our goal is to maximize (1), where the encoder takes F_i and F_j as inputs. During model inference, the decoder takes z and F_j as inputs to generate the augmented features $\hat{F}_i^L$ for the central node v_i.

Node and Edge Data Augmentation. Given the feature matrix F_G and the adjacency matrix A of the graph G, as well as the augmented features F_G^L generated by local neighborhood data augmentation. We perform random propagation for N times to further augment the above features.

Firstly, we apply DropNode to completely remove the features of certain nodes. This strategy is motivated by the homophily assumption [4], which suggests that neighboring nodes usually have similar features and labels. The removed node information can be compensated by the features of neighboring nodes, forming an approximate representation of the node in the corresponding data augmentation. Let $\kappa_i \sim \mathrm{Bernoulli}(1\text{-}\delta)$ be a binary mask sampled for each target node v_i, the perturbed feature matrix $\overline{F}$ can be represented as (2).

$$\overline{F} = \frac{\kappa_{ij}}{1-\delta}F, \quad (2)$$

Next, we use DropEdge to randomly delete some edges to alleviate over-smoothing. Let A_r be the set of edges deleted from the original edge set E, we can obtain (3).

$$\overline{A} = A - A_r, \quad (3)$$

Finally, we adopt a mixed propagation strategy for feature propagation, as shown in s (4) and (5).

$$\hat{F} = \tilde{A}\overline{F}, \quad (4)$$

$$\tilde{A} = \sum_{t=0}^{T}\frac{1}{T+1}\overline{A}^t, \quad (5)$$

Compared to using $\overline{A}^t$, using (5) introduces more local information to further alleviate over-smoothing.

B. Dual Network Prediction

Prediction. Given the original graph features F_G and the locally augmented features F_G^L. Assume there are h nodes in the original graph, after N rounds of random propagation as described in Section 3.2.2, we generate the augmented feature matrices $\hat{F}^{(n)}$ and $\hat{F}_G^{(n)}$ ($1 \le n \le N$). We then input $\hat{F}^{(n)}$ and $\hat{F}_G^{(n)}$ into the classifiers, i.e., the proposed dual network, where we adopt MLP as the classifier in the experiments. The outputs are the predicted probabilities as shown in s (6) and (7).

$$P_n = f_{mlp}(\hat{F}^{(n)}, \xi), \quad (6)$$

$$P_n^L = f_{mlp}(\hat{F}_G^{(n)}, \xi), \quad (7)$$

where P_n and P_n^L are the predicted probabilities and ξ are the model parameters.

Loss Function. The loss function consists of three parts. The first part is the supervised loss, which is the average cross-entropy loss on the N augmented features. Taking the augmented features of the original graph $\hat{F}^{(n)}$ as an example, it can be expressed as (8).

$$L_{\mathrm{ce}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{h} l_i^{\cdot}\ \log P_i^n, \quad (8)$$

It should be noted that during training, we duplicate the original graph features and the augmented features, and input them into two identical MLP classifiers with dropout. Therefore, the supervised loss comes from two different classifiers, denoted as L_{ce1} and L_{ce2}.

The second part of the loss function comes from the KL divergence loss between the two classifiers, as shown in (9). Suppose $\varpi = l_i | \hat{F}_i^n$, $\theta = l_i | \hat{F}_i^n$.

$$L_{KL} = \frac{1}{2}(KL(D_1 \varpi D_2 \theta) + KL(D_2 \theta D_1 \varpi), \quad (9)$$

where $D_1(l_i | \hat{F}_i^n)$ and $D_2(l_i | \hat{F}_i^n)$ are the distributions predicted by the two classifiers.

In the third part, the loss function is the consistency regularization loss, as shown in (10):

$$L_{\text{con}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{h}\left\|\tilde{P}_i - P_i^n\right\|_2^2, \quad (10)$$

where $\tilde{P}_i$ comes from sharpening techniques [11], and the label is guessed based on the mean of all distributions $\bar{P}_i = \frac{1}{N}\sum_{n=1}^{N} P_i^n$. Specifically, the probability that the i-th node guesses the j-th $(1 \le j \le K)$ category can be calculated using (11):

$$\tilde{P}_{ij} = \bar{P}_{ij}^{\frac{1}{T}} / \sum_{k=1}^{K} \bar{P}_{ik}^{\frac{1}{T}}, \quad (11)$$

where T $(0 < T \le 1)$ is a coefficient controlling the category distribution.

The final loss function we optimize is as shown in (12):

$$L_{all} = \frac{1}{2}(L_{\text{ce1}} + L_{\text{ce2}}) + \lambda L_{KL} + \mu L_{\text{con}}, \quad (12)$$

where λ and μ are both control coefficients.

IV. Experimental Setup

In this section, the overall experimental design of the DADN model will be introduced in detail from three aspects: datasets, baseline models, and evaluation metrics.

A. Datasets

This study conducts experiments on three publicly available citation network datasets: Cora, Citeseer, and Pubmed. In these datasets, each paper is represented as a node, and the citation relationships are represented as edges between nodes. We adopt a fixed training, validation, and test set split method [8]. Table I provides detailed statistics of the three datasets.

TABLE I. The Statistics of Datasets

Statistics	Cora	Citeseer	Pubmed
# Node	2708	3327	19717
# Edge	5429	4732	44338
# Category	7	6	3
# Feature dimension	1433	3703	500
Training set	140	120	60
Validation set	500	500	500
Test set	1000	1000	1000

B. Baseline Models

To evaluate the performance of DADN proposed in this paper, it is compared with the following representative advanced node classification baseline models:

- **Graphsage** [14] is a scalable GNN framework that generates node representations by aggregating local neighborhood information.
- **GCN** [15] is a graph convolutional network framework that learns node representations by aggregating neighborhood information and applying a graph-level activation function.
- **GAT** [16] introduces an attention mechanism to aggregate neighborhood information by assigning different weights to different neighboring nodes.
- **Grand** [7] performs data augmentation on graphs by designing random propagation strategies and enforces consistency regularization to ensure consistent predictions across different augmented features.
- **LA-Grand** [17] builds upon Grand by introducing local augmentation, providing richer graph structure information.

C. Evaluation Metrics

We employ evaluation metrics, namely Precision, Recall, F1 Score, and Accuracy, to comprehensively assess the performance of the models. The detailed specifications of each metric are expounded below.

Precision: This metric gauges the accuracy of the model by measuring the ratio of correctly predicted positive samples to the total predicted positive samples. Higher values indicate more precise predictions.

Recall: By measuring the ratio of correctly predicted positive samples to the actual positive samples, this metric assesses the model's ability to identify true positive samples Higher values indicating more true positive samples are identified.

F1 Score: This metric computes the harmonic mean of precision and recall, taking into account both the accuracy and comprehensiveness of the model. Higher values indicate a superior overall performance of the model.

Accuracy: By measuring the ratio of correctly predicted total samples to the total number of samples, this metric evaluates the overall classification ability of the model. Higher values indicate enhanced overall classification performance of the model.

V. Experiment Analysis

In this section, we investigate the effectiveness of our augmentation-driven dual network framework model DADN. We conduct comparative analysis against state-of-the-art node classification baselines on three popular benchmark datasets, to demonstrate the superior performance of our model. To further validate the contribution of the composite data augmentation module and dual network module, we perform an ablation study. Finally, we explore the impact of our DADN model.

A. Overall Performance

In order to verify the effectiveness of DADN, we conducted experiments on three datasets, namely Cora,

Citeseer, and Pubmed, and compared the results with a series of state-of-the-art baseline models using four evaluation metrics. The experimental results are shown in Tables II, III, IV, which presents the mean and standard deviation of the model's performance over ten runs, and the bold indicates the best performance according to the specified evaluation metric.

TABLE II. THE CLASSIFICATION RESULTS OF CORA DATASET

Method	Cora			
	Precesion	*Recall*	*F1*	*Acc*
Graphsage	78.61±1.00	81.89±0.48	79.57±0.83	80.52±0.77
GCN	78.76±0.83	82.03±0.64	79.96±0.73	80.63±0.73
GAT	80.92±0.75	84.11±0.58	82.15±0.65	82.85±0.74
Grand	83.94±0.71	84.27±0.41	83.94±0.52	85.31±0.57
LA-Grand	83.78±0.44	84.17±0.41	83.83±0.39	85.42±0.32
Ours	**84.13±0.36**	**84.45±0.35**	**84.14±0.34**	**85.72±0.27**

TABLE III. THE CLASSIFICATION RESULTS OF CITESEER DATASET

Method	Citeseer			
	Precesion	*Recall*	*F1*	*Acc*
Graphsage	65.05±0.70	68.95±0.46	66.30±0.58	67.24±0.64
GCN	66.34±0.39	66.48±0.47	65.95±0.42	68.46±0.48
GAT	68.13±0.27	67.48±0.12	67.25±0.17	71.11±0.23
Grand	72.12±0.41	71.45±0.5	71.51±0.46	75.44±0.36
LA-Grand	71.94±0.24	71.57±0.27	71.54±0.25	75.67±0.22
Ours	**72.30±0.41**	**71.80±0.33**	**71.76±0.28**	**76.09±0.41**

TABLE IV. THE CLASSIFICATION RESULTS OF PUBMED DATASET

Method	Pubmed			
	Precesion	*Recall*	*F1*	*Acc*
Graphsage	78.06±1.06	81.78±0.28	79.31±0.74	80.16±0.80
GCN	78.36±0.59	78.99±0.51	78.57±0.39	79.06±0.43
GAT	79.02±0.79	78.40±0.59	78.59±0.60	79.12±0.53
Grand	82.37±0.68	82.14±0.61	82.14±0.53	82.55±0.51
LA-Grand	82.83±0.74	**82.76±0.68**	82.65±0.72	83.21±0.67
Ours	**83.04±0.56**	82.71±0.51	**82.78±0.36**	**83.53±0.42**

In Table II, III, IV, it can be observed that DADN outperforms all baseline models in terms of the performance metrics on the three datasets, except for 0.05% lower recall than LA-Grand on the Pubmed dataset. Taking the Cora dataset as an example, DADN achieves a performance improvement of 0.35%, 0.28%, 0.31%, and 0.30% in terms of precision, recall, F1, and accuracy, respectively, compared to the strongest baseline model, with smaller variances. This indicates that our model has advantages in terms of both accuracy and stability.

B. Ablation Study

To evaluate the contributions of each sub-module in DADN, we conducted an ablation study. Specifically, we removed three sub-modules from the complete model, including (1) w/o Local, which removes the neighborhood data augmentation module; (2) w/o N&E (Node & Edge), which eliminates the node and edge data augmentation module; (3) w/o Dual, which removes the dual network module. The performance of DADN and the three variant models on the three datasets are shown in Figs. 3-5.

The experimental results show that the variant models with removed the corresponding sub-module exhibit decreased performance compared to the complete DADN model on all three datasets. This indicates that each sub-module has a positive impact on improving the performance of the node classification task. Furthermore, except for the recall metric on the Cora dataset, w/o N&E has the largest performance gap with DADN among all other metrics. This may be because the introduced augmented features by this sub-module provide more effective supervision signals to the model, resulting in better representations.

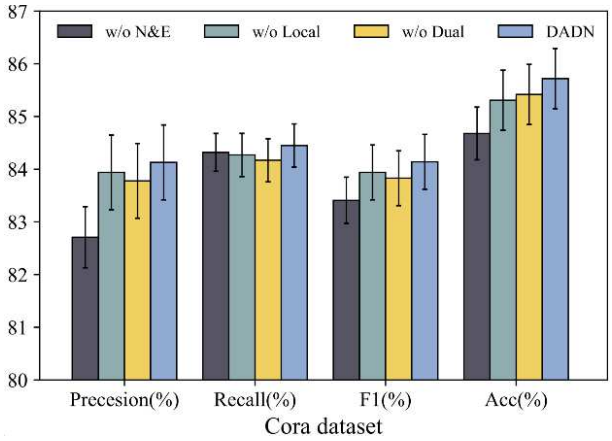

Fig. 3. Results of ablation study on Cora dataset.

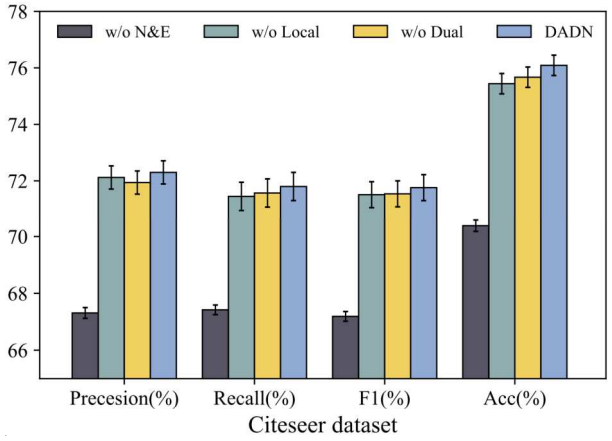

Fig. 4. Results of ablation study on Citeseer dataset.

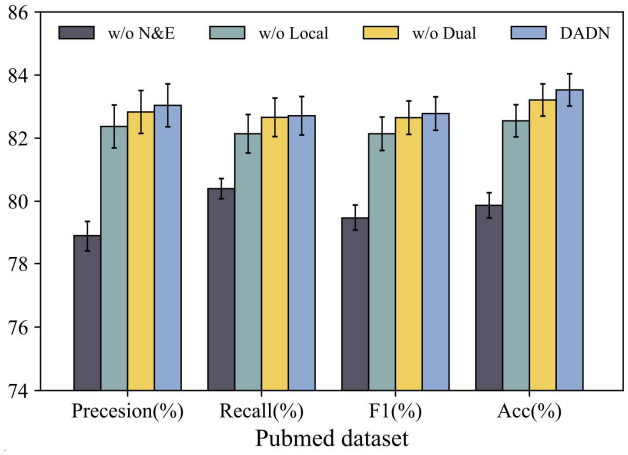

Fig. 5. Results of ablation study on Pubmed dataset.

C. Impact of Parameter λ

Additionally, we investigated the impact of parameter λ, which represents the weight of KL divergence loss on the performance. Specifically, we conducted experiments with different values of {1e-5, 5e-5, 1e-4, 5e-4, 1e-3} and the effects on the Cora, Citeseer, and Pubmed datasets are shown in Figs. 6-8, respectively.

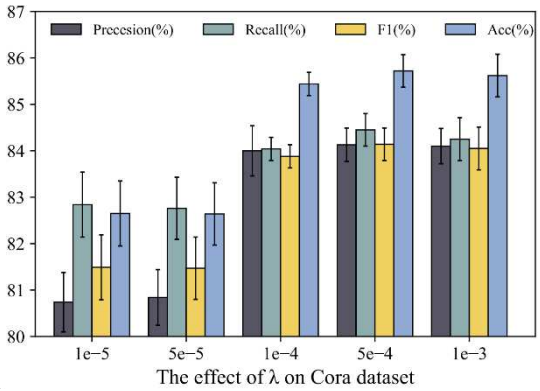

Fig. 6. The impact of Parameter λ on Cora dataset.

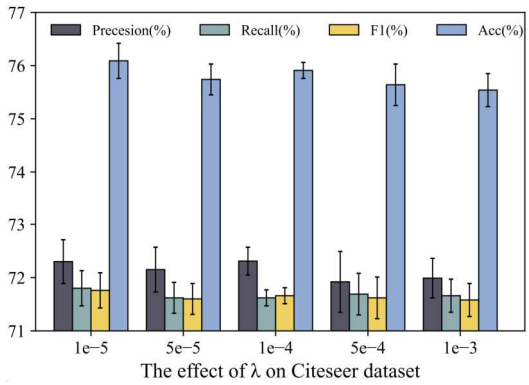

Fig. 7. The impact of Parameter λ on Citeseer dataset.

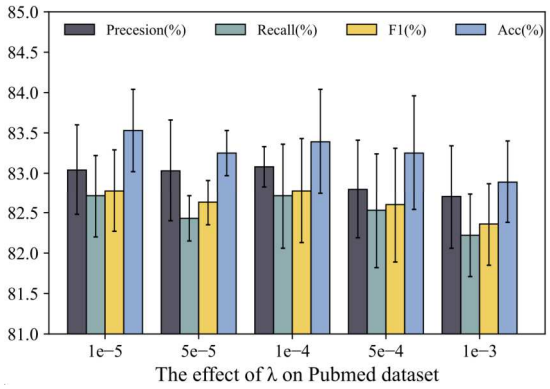

Fig. 8. The impact of Parameter λ on Pubmed dataset.

From Fig. 6, it can be observed that the performance metrics on the Cora dataset show an increasing trend at first and then decrease as λ increases, reaching the optimum at a certain value of 5e-4. Fig. 7 and Fig. 8 reveal that the changing trends of most metrics on the Citeseer and Pubmed datasets are similar, exhibiting a pattern of decreasing, increasing, and then decreasing again, with the optimum at 1e-5. Taking the Cora dataset as an example, 5e-4achieves an average positive gain of 2.68% across the four metrics than1e-5, indicating that KL divergence loss deserves more attention. However, excessively large λ is also unsuitable. For instance, on the Citeseer dataset, 1e-3 can result in an average decrease of 0.42% across the four metrics than1e-5. Therefore, the optimal choice of λ varies depending on factors such as the dataset and model size.

VI. Conclusion

In this paper, we propose a data augmentation-driven dual network framework, namely DADN, for node representation learning and apply it to the node classification task. By generating multiple augmented training samples using a composite data augmentation method and leveraging the augmented samples for node representation learning, our approach DADN is able to better capture the high-order relationships between nodes and features. Furthermore, the introduction of the dual network training framework and KL divergence loss allows the full exploitation of the data information and improves the robustness of the model. The experimental results show that the proposed framework achieves good performance in the node classification task.

In the future, we will explore more effective data augmentation strategies to further improve the quality of the node representation.

References

[1] B. X. Zhang, et al., "The expressive power of graph neural networks: A survey," arXiv preprint arXiv:2308.08235, 2023.

[2] Z. H. Wu, et al., "A comprehensive survey on graph neural networks," IEEE transactions on neural networks and learning systems, vol. 32, no. 1, pp. 4-24, 2020.

[3] T. Zhao, G. Liu, S. Günnemann, and M. Jiang. "Graph data augmentation for graph machine learning: A survey," arXiv preprint arXiv:2202.08871, 2022.

[4] J. Wang, L. Perez. "The effectiveness of data augmentation in image classification using deep learning," Convolutional Neural Networks Vis. Recognit, vol. 11, pp. 1-8, 2017.

[5] M. Bayer, et al., "Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers," International journal of machine learning and cybernetics, vol.14, no. 1, pp. 135-150, 2023.

[6] J. Zhou, et al., "Data Augmentation on Graphs: A Survey," arXiv preprint arXiv:2212.09970, 2022.

[7] W. Z. Feng, et al., "Graph random neural networks for semi-supervised learning on graphs," Advances in neural information processing systems, pp. 22092-22103, 2020.

[8] Y. N. You, et al., "Graph contrastive learning with augmentations," Advances in neural information processing systems, pp. 5812-5823, 2020.

[9] K. Z. Kong, et al., "Flag: Adversarial data augmentation for graph neural networks," arXiv preprint arXiv:2010.09891, 2020.

[10] R. Yu, W. B. Huang, T. Y. Xu, and J. Z. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," arXiv preprint arXiv:1907.10903, 2019.

[11] Z. Cheng, et al., "Robust graph representation learning via neural sparsification," International Conference on Machine Learning, pp. 11458-11468, 2020.

[12] D. S. Luo, et al., "Learning to drop: Robust graph neural network via topological denoising," International conference on web search and data mining, pp. 779-787, 2021.

[13] Z. Gao, et al., "Training robust graph neural networks with topology adaptive edge dropping," arXiv preprint arXiv:2106.02892, 2021.

[14] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," In Advances in Neural Information Processing Systems, pp. 1024-1034, 2017.

[15] T. N. Kipf ,and M. Welling. "Semi-supervised classification with graph convolutional networks," International Conference on Learning Representations, arXiv:1609.02907, 2017.

[16] P. Veličković, et al., "Graph attention networks," In International conference on learning representations, arXiv:1710.10903, 2018.

[17] S. Liu, et al., "Local augmentation for graph neural networks," International Conference on Machine Learning, pp. 14054-14072, 2022.