Addressing the Impact of Localized Training Data in Graph Neural Networks

Akansha
Department of Mathematics
Manipal Institute of Technology
Manipal Academy of Higher Education - 576104, India.
akansha.agrawal@manipal.edu.

***Abstract*—Graph Neural Networks (GNNs) have achieved notable success in learning from graph-structured data, owing to their ability to capture intricate dependencies and relationships between nodes. They excel in various applications, including semi-supervised node classification, link prediction, and graph generation. However, it is important to acknowledge that the majority of state-of-the-art GNN models are built upon the assumption of an in-distribution setting, which hinders their performance on real-world graphs with dynamic structures. In this article, we aim to assess the impact of training GNNs on localized subsets of the graph. Such restricted training data may lead to a model that performs well in the specific region it was trained on but fails to generalize and make accurate predictions for the entire graph. In the context of graph-based semi-supervised learning (SSL), resource constraints often lead to scenarios where the dataset is large, but only a portion of it can be labeled, affecting the model's performance. This limitation affects tasks like anomaly detection or spam detection when labeling processes are biased or influenced by human subjectivity. To tackle the challenges posed by localized training data, we approach the problem as an out-of-distribution (OOD) data issue by by aligning the distributions between the training data, which represents a small portion of labeled data, and the graph inference process that involves making predictions for the entire graph. We propose a regularization method to minimize distributional discrepancies between localized training data and graph inference, improving model performance on OOD data. Extensive tests on popular GNN models show significant performance improvement on three citation GNN benchmark datasets. The regularization approach effectively enhances model adaptation and generalization, overcoming challenges posed by OOD data[1].**

***Index Terms*—Graph neural networks (GNNs), Out-of-distribution (OOD) graph data, Semi-supervised learning, Distributional discrepancy in graph data**

I. Introduction

Graph Neural Networks (GNNs) have emerged as a powerful framework for learning from and reasoning over graph-structured data. With the ability to capture complex dependencies and relationships between nodes, GNNs have achieved remarkable success in a wide range of applications, including social network analysis [1], [2], recommendation systems [3]–[5], drug discovery [6]–[8], and knowledge graph reasoning [9]–[11]. GNNs are versatile and can be used for various tasks, such as graph generation, community detection, and anomaly detection. However, node classification [12], [13], link prediction and graph classification are some of the most prevalent and well-studied applications of GNNs.

Despite the remarkable success of Graph Neural Networks (GNNs) in tasks such as node classification, link prediction, and graph generation, it is important to acknowledge that the majority of state-of-the-art GNN models are built upon the assumption of an in-distribution setting [14]–[17]. In this scenario, the training and test data are assumed to be drawn from the same distribution. However, in practice, it becomes challenging to meet this assumption when dealing with real-world graphs, such as social networks, citation networks, or population graphs. These graphs are often characterized by dynamic and evolving structures, diverse node attributes, and complex interactions. As a result, the performance of traditional GNN models tends to degrade when confronted with out-of-distribution (OOD) data.

Effectively handling OOD data is vital for ensuring the practical usability of GNNs in real-world scenarios with high-stakes applications. These applications include molecule prediction [18], financial analysis [19], criminal justice [20], autonomous driving [21], and pandemic prediction [22]. To address the OOD problem in GNNs, several attempts have made, for instance, data or graph augmentation [23]–[25]: these techniques involve introducing variations to the graph structure or node attributes during the training phase to simulate OOD scenarios. By training GNN models on augmented data that covers a broader range of graph variations, the models can learn to be more robust and generalize better to OOD data. Disentanglement-Based Graph Models [26]–[29]: this approach focuses on designing novel GNN architectures that explicitly disentangle the propagation step from the non-linear transformations. These models aim to separate the graph structure and the node attributes to better capture and model the complex relationships within the graph. Learning Strategies [30]–[37]: various learning strategies are proposed to improve GNN performance on OOD data. These strategies involve adapting the training process or loss functions to account for the challenges posed by OOD scenarios.

In this article, we aim to measure the impact of training the model on only a localized subset of the graph. This restricted training data may result in a model that performs well on the specific region it was trained on but fails to generalize and make accurate predictions for the entire graph.

[1]Codes are available at https://github.com/Akanshaaga/Reg_APPNP.

The inability of a GNN model trained on localized data to generalize effectively raises concerns in various domains. For instance, anomaly detection or spam detection [38], [39] tasks can suffer greatly when the labeling process is biased or influenced by human subjectivity. In such scenarios, the model's performance may be inflated within the labeled region, but its predictions outside that region may be erroneous. This limitation undermines the reliability and practical applicability of the GNN model. To tackle the challenges posed by localized training data, we approach the problem as an OOD data issue. We recognize the need to align the distributions between the training data, which represents a small portion of labeled data, and the graph inference process that involves making predictions for the entire graph. By minimizing the distributional shift between these two components, we aim to enhance the model's ability to generalize effectively and make accurate predictions beyond the limited training subset.

To bridge the gap between the localized training data and the whole-graph inference, we propose a regularization method. Our regularization approach focuses on minimizing the distributional discrepancy between the two distributions, ensuring that the model can accurately predict beyond the localized training subset. By simulating a shift in the training data, we introduce bias during the labeling process as done in [40], creating a more significant difference between the two distributions. The regularization is then performed by minimizing the distributional discrepancy between the entire graph distribution and the training data distribution obtained with the introduced bias. This regularization technique guides the model to adapt and generalize effectively, mitigating the limitations posed by localized training data.

Key observation. Our observations reveal that the degradation in accuracy in traditional state-of-the-art Graph Neural Networks (GNNs) [14]–[16] can be attributed to the propagation of mismatched probability distributions. This issue becomes exacerbated with each layer, resulting in poor accuracy. Consequently, in the context of out-of-distribution (OOD) graph data, improving accuracy necessitates minimizing the discrepancy between domain-specific features before propagation in the classification task. In light of this, we find that models [25], [41], [42] which treat propagation and nonlinear transformation operations as separate steps tend to perform better on OOD graph data.

Our contributions. Our contributions in this research are threefold. Firstly, the proposed technique demonstrates stability by consistently improving the average prediction accuracy while maintaining a small standard deviation. This indicates that our regularization method reliably enhances the model's performance on out-of-distribution (OOD) data. Secondly, the proposed regularization approach can be applied to any given graph or Graph Neural Network (GNN) model, making it a versatile solution for addressing OOD data challenges across various domains. Lastly, in the experimental section, we conducted extensive tests on popular GNN models, applying our regularization strategy. The results showed a significant improvement in performance when handling OOD data. Specifically, we evaluated our approach on a well-known citation network and observed a significant increase in accuracy compared to the previously reported results in [40]. These findings highlight the efficacy of our regularization method in enhancing model performance and addressing the challenges posed by OOD data.

II. Related Work

Various techniques have been proposed to address the problem of out-of-distribution (OOD) data in graph-based scenarios. These techniques can be broadly categorized into three categories.

The first category involves data or graph augmentation techniques. These techniques introduce variations to the graph structure or node attributes during the training phase to simulate OOD scenarios. By training Graph Neural Network (GNN) models on augmented data that covers a broader range of graph variations, the models can learn to be more robust and generalize better to OOD data. Several studies have explored this approach and demonstrated its effectiveness [23]–[25].

The second category focuses on disentanglement and causality-based GNNs. These approaches aim to design novel GNN architectures that explicitly separate the propagation step from the non-linear transformations. By disentangling the graph structure and node attributes, these models can better capture and model the complex relationships within the graph, leading to improved performance on OOD data [26]–[29].

The third category comprises learning strategies specifically tailored for OOD scenarios. Various learning strategies have been proposed to enhance GNN performance on OOD data. These strategies include graph invariant learning, graph adversarial learning, and graph self-supervised learning. Graph invariant learning aims to learn representations that are invariant to graph isomorphisms, while graph adversarial learning focuses on training GNNs that are robust against adversarial attacks. Graph self-supervised learning leverages unlabeled data to pretrain GNN models and learn meaningful representations. These learning strategies adapt the training process or loss functions to address the challenges posed by OOD scenarios [30]–[37].

However, our work closely follows the line of shift-robust GNNs and employs a size shift regularization method [31], [40]. Inspired by these approaches, we aim to tackle the challenges of OOD data by introducing regularization that simulates a shift in the training data. Our regularization method is designed to make the model more robust to this shift and improve its performance on OOD data scenarios.

III. Preliminaries

Notation. In our notation, the graph is represented as $G = (V, E)$, where $|V| = n$ denotes the number of nodes and $|E| = m$ denotes the number of edges in the graph. Specifically, V represents the set of nodes, and E represents the set of edges in the graph. To denote the 1-hop neighborhood of a node u in the graph, we use $N_u \subset V$, which represents the set of nodes that are directly connected to node u within one edge. Furthermore,

(u, v) is used to denote an edge between node u and node v, indicating a connection between the two nodes in the graph. In the context of the graph, we employ the adjacency matrix $A \in \{0,1\}^{n\times n}$, where each entry in the matrix corresponds to the presence or absence of an edge between two nodes. A value of 1 indicates the presence of an edge, while a value of 0 indicates the absence of an edge. This matrix captures the initial features $x_u \in \mathbb{R}^d$ associated with each node $u \in V$ in the graph, allowing for the representation of various node attributes or properties. Additionally, we utilize the feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, where n represents the number of nodes and d denotes the dimensionality of the feature vectors.

Biased data generation. To investigate the impact of localized training data on the node classification task, it is necessary to generate biased training data with controllable bias. In order to achieve this, we utilize the personalized PageRank vector of each node, denoted as $\Pi = (I - (1-\alpha)\tilde{A})^{-1}$. Here, $\tilde{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}$ represents the normalized adjacency matrix, where $D = \sum_{i=1}^{n} a_{ii}$ is the degree matrix of $A+I = (a_{ij})$ [41]. Following the methodology described in [40], we employ this personalized PageRank vector to generate biased training data.

Central Moment Discrepancy. Central Moment Discrepancy (CMD) is a the metric measures the direct distance between two distributions p and q using their central moments, given by

$$d_{CMD}(p,q) = \frac{1}{|b-a|}\|E(p)-E(q)\|_2 + \sum_{k=2}^{\infty}\|c_k(p)-c_k(q)\|_2,$$

where a, b denotes the joint distribution support, $E(\cdot)$ is the expected values of the distributions and $c_k(\cdot)$ are their respective central moments, considering different orders k of moments.

Maximum Mean Discrepancy. Maximum Mean Discrepancy (MMD) is a metric used to measure the difference between means of distributions in a high-dimensional feature space defined by a kernel function. It quantifies the dissimilarity between two probability distributions by comparing their expected values in the rich Hilbert space. The formula for MMD is:

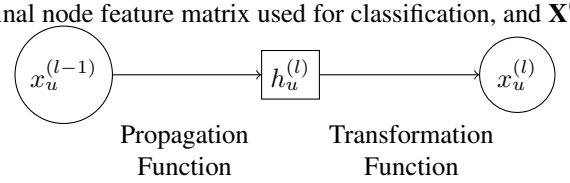
$$d_{MMD}(p,q) = \|\mu_p - \mu_q\|_H,$$

where p and q are two probability distributions, μ_p and μ_q are their means in the Hilbert space H, and $\|\cdot\|_H$ represents the norm in that space.

Graph Neural Network (GNN) Layer: A typical l-th layer of a Graph Neural Network can be defined as follows:

$$h_u^{(l)} = propagation\left\{x_u^{(l-1)}, \{x_v^{(l-1)} \text{ where } v \in N_u\}\right\}$$
$$x_u^{(l)} = transformation\{(h_u^{(l-1)})\},$$

where $x_u^{(l)}$ represents the feature vector of node u at layer l, and $x_u^{(0)}$ is the initial feature representation of node u. The propagation step updates $x_u^{(l)}$ based on feature vectors of neighboring nodes in N_u, and the transformation step further processes the updated feature vector $h_u^{(l)}$. The process is repeated for a total of K layers, with $\mathbf{X}^{(K)}$ representing the final node feature matrix used for classification, and $\mathbf{X}^{(0)} = \mathbf{X}$.



IV. Problem setup.

In this paper, we tackle the challenge of distributional shift in the context of semi-supervised learning (SSL) on a single graph where only a small portion is labeled. In traditional SSL, a common approach is to use the cross-entropy loss function, which calculates the difference between the predicted labels $\hat{y}_i$ generated by a graph neural network for each node i, and the true labels y_i. The overall loss l is then computed as the average of individual losses over all training examples M:

$$l = \frac{1}{M}\sum_{i=1}^{M} l(y_i, \hat{y}_i).$$

When training and testing data come from the same distribution, that is $P_{\text{train}}(X,Y) = P_{\text{test}}(X,Y)$, optimizing the cross-entropy loss on the training data ensures that the classifier is well-calibrated for making accurate predictions on the testing data. However, a significant challenge in machine learning arises when there is a mismatch between the distributions of the training and testing datasets, i.e., $P_{\text{train}}(X,Y) \neq P_{\text{test}}(X,Y)$. To address this issue, we focus on the distributional shift specifically in the output of the last hidden layer, which is denoted as $\hat{Y}$. Standard learning theory assumes that the distribution of labels given the representations $\hat{Y}$ is the same for both training and test data, that is $P_{\text{train}}(Y|\hat{Y}) = P_{\text{test}}(Y|\hat{Y})$.

V. Regularized Approximate personalized propagation of neural prediction (Reg-APPNP)

In this section, we tackle the issue of distributional shift in graph neural networks, where the probability distributions of predicted labels differ between the training and testing data (i.e., $P_{train}(\hat{Y}) \neq P_{test}(\hat{Y})$). Since we have access to both the graph structure and unlabeled data, we propose a regularization technique that utilizes two discrepancy metrics: Central Moment Discrepancy (CMD) and Maximum Mean Discrepancy (MMD). These metrics allow us to quantify the dissimilarity between the probability distributions of the training and testing data.

A. Reg-APPNP framework.

We introduce our regularization method designed to minimize the discrepancy in the latent representations induced by distributional shift:

$$\mathcal{L}_{Reg-APPNP} = \frac{1}{M}\sum_{i=1}^{M} l(y_i, \hat{y}_i) + \lambda d_{CMD}(\hat{Y}_{train}, \hat{Y}_{test}) + \beta d_{MMD}(\hat{Y}train, \hat{Y}test)$$

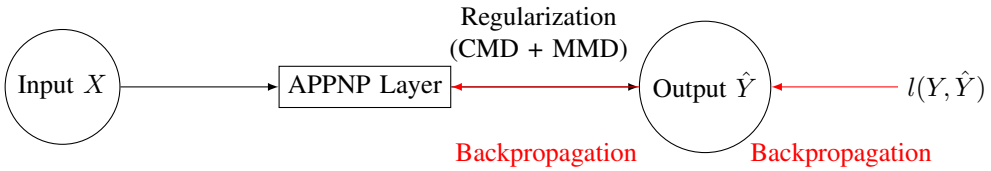


Fig. 1. Illustration of Regularized APPNP (Reg-APPNP) Architecture. The regularization technique mitigates the impact of distributional shift in predicted labels. The model is trained end-to-end, ensuring comprehensive optimization.

In this formulation, the regularization term includes the cross-entropy loss (l) between true labels y_i and predicted labels $haty_i$, and two discrepancy metrics, CMD and MMD. The parameters λ and β control the importance of each regularization component. Empirical results demonstrate that incorporating both CMD and MMD metrics as regularization techniques potentially enhances the regularization's effectiveness because they capture different aspects of distributional shift. CMD assesses the direct distance between distributions, focusing on their central moments that convey shape and variability information. On the other hand, MMD emphasizes differences in the means of distributions, capturing more global distinctions. Our approach employs the APPNP model [41] on biased data with the proposed regularization term. Extensive experiments on real-world citation networks reveal that in some cases, one metric may dominate in capturing distributional shift, while in others, both metrics contribute equally. We fine-tune the parameters λ and β to achieve improved accuracy in the semi-supervised learning (SSL) classification task.

VI. Experiments.

In this section, we apply our proposed regularization technique to biased sample data, as generated in the work of Zhu et al. [40]. We use the same validation and test splits as in the GCN paper by Kipf and Welling [14]. Our goal is to evaluate the effectiveness of our framework, Reg-APPNP, in minimizing distributional shifts compared to other baseline methods. Furthermore, we present a comprehensive study on hyperparameter optimization specifically for the CORA dataset. Hyperparameters play a crucial role in the performance of our regularization technique, and we thoroughly investigate and optimize these parameters to achieve the best results.

Datasets. In our experiments, we focus on the semi-supervised node classification task using three widely-used benchmark datasets: Cora, Citeseer, and Pubmed. For the biased training samples, we utilize the same data as generated for SR-GNN in Zhu et al.'s work [40]. To ensure fair comparison, we adopt the validation and test splits used in the GCN paper by Kipf and Welling [14]. In order to obtain unbiased data, we perform random sampling from the remaining nodes after excluding those assigned to the validation and test sets.

Baselines. To investigate the performance under distributional shift, we employ two types of GNN models. Firstly, we utilize traditional GNNs that incorporate message passing and transformation operations, including GCN [14], GAT [15], and GraphSage [16]. Secondly, we explore another branch of GNNs that treats message passing and nonlinear transformation as separate operations. This includes models such as APPNP [41] and DAGNN [42].

Our Method. Unless specified otherwise, we consider the instance of APPNP [41] with our proposed regularization technique, referred to as Reg-APPNP, as our base model. Additionally, we provide two ablations of Reg-APPNP for validation purposes. The first ablation includes only the CMD metric as regularization, and the second ablation includes only the MMD metric. By conducting these ablations, we aim to assess the effectiveness of our proposed Reg-APPNP method in addressing distributional shifts and enhancing the performance of the graph neural network models.

Hyperparameters. The primary parameters in our methods are the penalty parameters λ and β, which correspond to the regularization metrics d_{CMD} and d_{MMD}, respectively. Based on empirical results, we have determined the optimal values for these parameters for the SSL classification task on the three benchmark datasets. For the Cora dataset, we found that $\lambda = 0.5$ and $\beta = 1$ achieve the highest accuracy. In the case of Citeseer, the optimal values are $\lambda = 0.1$ and $\beta = 1$, while for the Pubmed dataset, $\lambda = 0.1$ and $\beta = 0.1$ yield the best performance. Regarding the APPNP model, we set $\alpha = 0.1$, and we utilize the DGL (Deep Graph Library) layer for implementing the APPNP model within our experimental setup. The results, including the impact of different values of λ on the Cora dataset, are depicted in Figure 2.

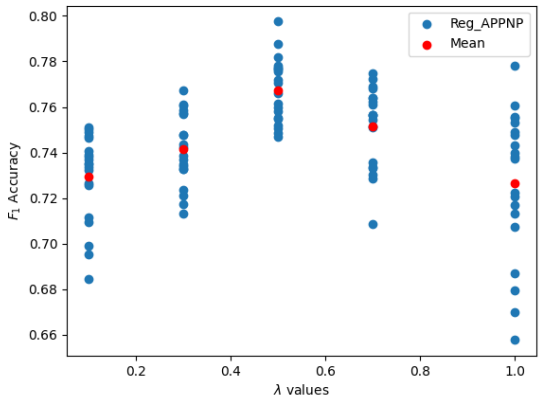


Fig. 2. Cumulative Impact of Penalty Parameter λ (corresponding to d_{CMD}) on Accuracy.

A. Experimental Results

We begin by presenting a performance comparison between our proposed regularization technique and the shift-robust technique proposed by Zhu et al. [40], when combined with popular GNN models trained on biased training data. The F1-accuracy results for semi-supervised node classification for CORA data set are tabulated in Table I. The table demonstrates that our regularization technique consistently enhances the performance of each GNN model when applied to biased data. Notably, our method, Reg-APPNP, outperforms other baselines, including SR-GNN, on biased training data.

This comparison confirms the efficacy of our proposed regularization technique in mitigating the impact of distributional shifts, leading to enhanced performance across various GNN models. Reg-APPNP emerges as the top-performing method, showcasing its effectiveness in handling biased training data and improving the accuracy of the semi-supervised node classification task.

TABLE I
PERFORMANCE COMPARISON OF OUR PROPOSED REGULARIZATION TECHNIQUE WHEN COMBINED WITH POPULAR GNN MODELS. THE F_1-MICRO ACCURACY IS REPORTED AND COMPARED WITH THE SHIFT-ROBUST TECHNIQUE [40].

Model	F_1-score	F_1-score w. Reg	F_1-score w. SR
GCN	65.35 ± 3.6	71.41 ± 3.6	70.45 ± 3.5
GraphSAGE	67.56 ± 2.1	72.10 ± 2.0	71.79 ± 2.4
APPNP	70.7 ± 1.9	**75.14** ± 1.8	72.33 ± 5.9
DAGNN	71.74 ± 0.32	72.71 ± 0.31	71.79 ± 0.36
SR-GNN	73.5 ± 3.3 (**[40]**)	70.21 ± 0.87	70.42 ± 1.1

In their work [40], the authors demonstrated that as the level of bias increases, popular GNN models experience a decline in performance. In Figure 3, we present the behavior of our Reg-APPNP model as we increase the bias in the training data. The results clearly show that our proposed regularization technique consistently improves the performance in the SSL node classification task compared to the baseline APPNP model trained on biased data.

We applied our Reg-APPNP method to three popular citation GNN benchmarks and report the F_1-micro accuracy in Table II. To assess the impact of biased training data, we compare each model with the baseline APPNP trained on in-distribution IID data. As shown in the second row of Table II, APPNP's performance drops significantly when trained on biased data.

However, our proposed Reg-APPNP demonstrates significant improvement in performance on biased data, outperforming SR-GNN [40] when we executed their code on our system. It's worth noting that the results for SR-GNN in the table differ from the reported ones for Citeseer and Pubmed. The values in the table correspond to the results we reproduced using their code on our personal system.

Finally, we observe that combining both metrics, d_{CMD} and d_{MMD}, is more effective than using only one of them during the entropy loss step. This finding further highlights the benefits of our comprehensive regularization approach in addressing distributional shifts and improving the performance of graph neural networks on biased training data.

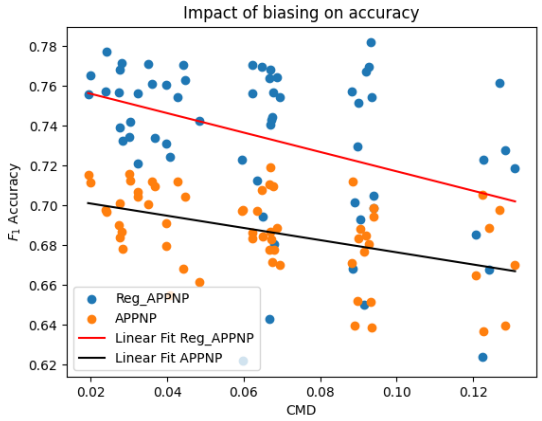


Fig. 3. Cumulative Impact of Biasing on Reg-APPNP and APPNP Models.

TABLE II
PERFORMANCE COMPARISON OF REG-APPNP ON CITATION NETWORK BENCHMARKS. WE REPORT F_1-MICRO ACCURACY AND COMPARE WITH APPNP TRAINED ON IN-DISTRIBUTION IID DATA.

Model	Cora	Citeseer	Pubmed
APPNP (Unbiased)	85.09 ± 0.25	75.73 ± 0.30	79.73 ± 0.31
APPNP	70.7 ± 1.9	60.78 ± 1.6	53.42 ± 1.3
Reg-APPNP w.o. CMD	71.37 ± 1.2	60.46 ± 1.9	53.35 ± 1.0
Reg-APPNP w.o. MMD	75.03 ± 1.2	62.72 ± 1.9	66.13 ± 1.2
Reg-APPNP(ours)	**75.14** ± 1.8	**65.01** ± 2.0	**70.71** ± 2.9
SR-GNN	73.5 ± 3.3	62.60 ± 0.6	68.78 ± 2.2

VII. CONCLUSION

In this work, we observed that as the level of bias increases, popular GNN models experience a decline in performance. We emphasized that biased training data is prevalent in real-world scenarios, and it can arise due to several reasons, such as challenges in labeling a large volume of data, the application of various heuristics or inconsistent techniques for node selection during labeling, delayed label assignment, and other constraints inherent in real-world problems. To address this challenge, we incorporate CMD and MMD as regularization terms in the APPNP model. This regularization approach encourages the model to minimize the discrepancy in the latent representations induced by the distributional shift. The proposed regularization approach can be applied to any given graph or Graph Neural Network (GNN) model, making it a versatile solution for addressing such challenges across various domains. This flexibility allows researchers and practitioners to leverage our method to improve their models' performance and generalization capabilities when dealing with OOD scenarios.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Karmvir Singh Phogat for the valuable insights and critical feedback on the research problem addressed in this article. The thoughtful comments greatly contributed to the improvement and clarity of this work.

REFERENCES

[1] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[2] A. Davies and N. Ajmeri, "Realistic synthetic social networks with graph neural networks," *arXiv preprint arXiv:2212.07843*, 2022.

[3] C. Gao, X. Wang, X. He, and Y. Li, "Graph neural networks for recommender system," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1623–1625.

[4] Y. Chu, J. Yao, C. Zhou, and H. Yang, "Graph neural networks in modern recommender systems," *Graph Neural Networks: Foundations, Frontiers, and Applications*, pp. 423–445, 2022.

[5] Y. Wang, Y. Zhao, Y. Zhang, and T. Derr, "Collaboration-aware graph convolutional networks for recommendation systems," *arXiv preprint arXiv:2207.06221*, 2022.

[6] T. Gaudelet, B. Day, A. R. Jamasb, J. Soman, C. Regep, G. Liu, J. B. Hayter, R. Vickers, C. Roberts, J. Tang *et al.*, "Utilizing graph machine learning within drug discovery and development," *Briefings in bioinformatics*, vol. 22, no. 6, p. bbab159, 2021.

[7] M. Pandey, M. Fernandez, F. Gentile, O. Isayev, A. Tropsha, A. C. Stern, and A. Cherkasov, "The transformational role of gpu computing and deep learning in drug discovery," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 211–221, 2022.

[8] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, and T. Langer, "A compact review of molecular property prediction with graph neural networks," *Drug Discovery Today: Technologies*, vol. 37, pp. 1–12, 2020.

[9] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," *arXiv preprint arXiv:2104.06378*, 2021.

[10] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, and J. Wang, "A comprehensive survey of graph neural networks for knowledge graphs," *IEEE Access*, vol. 10, pp. 75 729–75 741, 2022.

[11] Y. Zhang and Q. Yao, "Knowledge graph reasoning with relational digraph," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 912–924.

[12] S. Xiao, S. Wang, Y. Dai, and W. Guo, "Graph neural networks in node classification: survey and evaluation," *Machine Vision and Applications*, vol. 33, pp. 1–19, 2022.

[13] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," *arXiv preprint arXiv:1905.10947*, 2019.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[16] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[17] A. Wijesinghe and Q. Wang, "Dfnets: Spectral cnns for graphs with feedback-looped filters," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[18] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *Advances in neural information processing systems*, vol. 33, pp. 22 118–22 133, 2020.

[19] S. Yang, Z. Zhang, J. Zhou, Y. Wang, W. Sun, X. Zhong, Y. Fang, Q. Yu, and Y. Qi, "Financial risk analysis for smes with graph-based supply chain mining," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 4661–4667.

[20] C. Agarwal, H. Lakkaraju, and M. Zitnik, "Towards a unified framework for fair and stable graph representation learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 2114–2124.

[21] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.

[22] G. Panagopoulos, G. Nikolentzos, and M. Vazirgiannis, "Transfer graph neural networks for pandemic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 6, 2021, pp. 4838–4845.

[23] J. Yu, J. Liang, and R. He, "Mind the label shift of augmentation-based graph ood generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 620–11 630.

[24] Y. Sui, X. Wang, J. Wu, A. Zhang, and X. He, "Adversarial causal augmentation for graph covariate shift," *arXiv preprint arXiv:2211.02843*, 2022.

[25] W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, Q. Yang, E. Kharlamov, and J. Tang, "Graph random neural networks for semi-supervised learning on graphs," *Advances in neural information processing systems*, vol. 33, pp. 22 092–22 103, 2020.

[26] H. Li, Z. Zhang, X. Wang, and W. Zhu, "Disentangled graph contrastive learning with independence promotion," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[27] S. Fan, X. Wang, Y. Mo, C. Shi, and J. Tang, "Debiasing graph neural networks via learning disentangled causal substructure," *arXiv preprint arXiv:2209.14107*, 2022.

[28] H. Li, X. Wang, Z. Zhang, Z. Yuan, H. Li, and W. Zhu, "Disentangled contrastive learning on graphs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 872–21 884, 2021.

[29] Y. Zhou, G. Kutyniok, and B. Ribeiro, "Ood link prediction generalization capabilities of message-passing gnns in larger test graphs," *arXiv preprint arXiv:2205.15117*, 2022.

[30] H. Li, Z. Zhang, X. Wang, and W. Zhu, "Learning invariant graph representations for out-of-distribution generalization," in *Advances in Neural Information Processing Systems*, 2022.

[31] D. Buffelli, P. Liò, and F. Vandin, "Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks," *arXiv preprint arXiv:2207.07888*, 2022.

[32] Y. Wu, A. Bojchevski, and H. Huang, "Adversarial weight perturbation improves generalization in graph neural network," *arXiv preprint arXiv:2212.04983*, 2022.

[33] A. Sadeghi, M. Ma, B. Li, and G. B. Giannakis, "Distributionally robust semi-supervised learning over graphs," *arXiv preprint arXiv:2110.10582*, 2021.

[34] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2493–2504, 2019.

[35] Y. Wang, C. Li, W. Jin, R. Li, J. Zhao, J. Tang, and X. Xie, "Test-time training for graph neural networks," *arXiv preprint arXiv:2210.08813*, 2022.

[36] H. Liu, B. Hu, X. Wang, C. Shi, Z. Zhang, and J. Zhou, "Confidence may cheat: Self-training on graph neural networks under distribution shift," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1248–1258.

[37] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and S. Y. Philip, "Graph self-supervised learning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5879–5900, 2022.

[38] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 2077–2085.

[39] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, and Y. Qi, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 598–607.

[40] Q. Zhu, N. Ponomareva, J. Han, and B. Perozzi, "Shift-robust gnns: Overcoming the limitations of localized graph training data," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[41] J. Gasteiger, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *International Conference on Learning Representations (ICLR)*, 2019.

[42] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 338–348.