GRAPH NETWORKS STAND STRONG: ENHANCING ROBUSTNESS VIA STABILITY CONSTRAINTS

Zhe Zhao[1],*Pengkun Wang*[12†], *Haibin Wen*[3], *Yudong Zhang*[1], *Binwu Wang*[1],*Yang Wang*[12†]

[1] University of Science and Technology of China
[2] Suzhou Institute for Advanced Research, USTC
[3] Shaoguan University

ABSTRACT

Graph neural networks (GNNs) have achieved great success in graph classification tasks across many domains. However, the varying quality of real-world graph data leads to stability and reliability issues for real-world applications of graph neural networks (GNNs). Improving the robustness of GNNs would help enhance the quality and safety of GNNs in real-world applications. Recently, there have been studies that incorporate insights from information theory, causal theory, etc. into graph classification tasks to improve robustness. However, these strategies rely on extensive task-specific designs that increase model complexity and limit the scope of the methods. In this work, we leverage the interdependence between model stability and robustness by introducing stability constraints to graph neural network models through two different consistency regularization methods. To balance the trade-off between stability constraints and classification performance, we adaptively adjust the strength of the constraints dynamically using multi-objective optimization, making our method applicable to graph classification tasks of varying scales and domains. Extensive experiments on graph datasets from different domains demonstrate the superiority of our proposed method. The code is available in https://github.com/haibin65535/temp.

***Index Terms*—** Graph neural networks, Robustness, Stability

1. INTRODUCTION

Graph classification is crucial for many real-world applications, such as predicting molecular properties, analyzing social networks, and detecting fake news [1, 2, 3, 4, 5]. Thanks to abundant datasets and diverse advanced neural network architectures, graph neural networks have achieved state-of-the-art results on graph classification tasks across many domains. However, graph neural networks (GNNs) are vulnerable to adversarial attacks, improving the robustness of GNNs would help ensure quality and security for many GNN-dependent real-world applications. For example, enhancing GNN robustness could aid in detecting fake news or misinformation in social networks. There have been studies on improving GNN learning robustness, such as GIB [6] which compresses irrelevant graph classification information in representations to improve robustness by information bottleneck. In addition, DIR [7] tries to reduce non-causal information extraction to improve robustness through causal relevance relations. These methods improve neural network robustness by incorporating additional insights, but the introduction of different insights relies on extensive task-specific designs (e.g. mutual information estimators) unrelated to graph classification itself, thus limiting the applicability of the methods across domains. To achieve a more concise and generalizable robust graph classification learning framework, we try to leverage the connection between model stability and robustness. Model stability refers to how much a machine learning model's outputs vary in response to small perturbations or noise in the inputs. Papers [8, 9, 10] explore the close ties between model stability and adversarial attack robustness, that is, models with high stability tend to be more robust to corrupted inputs like noisy examples, adversarial examples, occlusions, etc. Inspired by this, we propose a new framework that improves GNN graph classification robustness by introducing stability constraints during graph classification learning. We implement these stability constraints through different consistency regularization methods and dynamically trade off imposing stability constraints and classification performance, making our method applicable to graph classification tasks of varying complexity across domains.

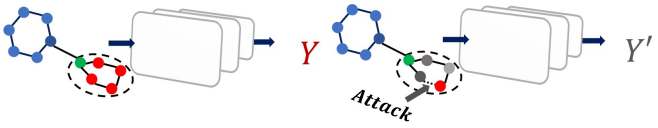


Fig. 1. The left subfigure shows the traditional graph classification process. The right subfigure shows that an attack on the graph data structure changes the GNN's prediction of the graph properties..

†:Corresponding Author.

Our contributions can be summarized as:

- *New insight and framework*: Unlike previous lines of work controlling irrelevant information, we propose a graph classification framework with stability constraints to improve graph learning robustness.
- *New strategy*: We propose two consistency regularization strategies to enforce stability constraints, and adaptively optimize the trade-off between stability maintenance and performance improvement through adaptive optimization strategies, making our method widely applicable.
- *Compelling empirical results*: Extensive experiments on graph datasets from multiple domains demonstrate the effectiveness of our method in improving graph classification robustness and its applicability across domains.

2. RELATED WORK

2.1. Graph Neural Networks

Graph neural networks (GNNs) have achieved great success in graph-based learning tasks [11, 12]. By propagating and transforming node features, GNNs can learn high-level representations of graphs. Given a graph $G = (V, E)$ with node features $\mathbf{X} \in R^{N\times F}$ where N is the number of nodes and F is the feature dimension, graph convolution operates as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{A}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right) \tag{1}$$

where $\mathbf{A} \in R^{N\times N}$ is the adjacency matrix, $\mathbf{H}^{(l)} \in R^{N\times F}$ is the hidden representation in the l^{th} layer, $\mathbf{W}^{(l)} \in R^{F\times F'}$ is a trainable weight matrix, and $\sigma(\cdot)$ denotes an activation function. By stacking multiple convolutional layers, GNNs can learn hierarchical representations of the input graph. However, existing GNNs still face challenges regarding robustness and can be vulnerable to adversarial attacks or subtle perturbations [13, 14, 15].

2.2. Enhancing Robustness of Graph Classification

Recent works have sought to improve GNN robustness through information bottlenecks [6], capturing predictive substructures. By filtering out non-predictive signals, bottleneck-based methods enhance model invariance to perturbations. Another line of work leverages causal discovery to identify robust predictive subgraphs [16, 7], By focusing on invariant causal mechanisms, these methods can debias GNNs from spurious correlations. However, they rely on restrictive assumptions and lack scalability. Different from these works, we approach the problem of enhancing robustness by introducing a stability constraint for the graph classification task. The universal existence of stability conditions makes our framework more concise and widely applicable to various domains and architectures.

3. METHOD

3.1. Problem Frame: Graph Classification with Stability Constraints

In this section, we first provide a definition of our problem framework. We give a formal description of the graph classification task, neural network robustness and neural network stability. Then, we introduce the definition of our optimization problem, namely, graph classification with stability constraints.Given a graph dataset $\mathcal{D} = \{(G_i, y_i)\}_{i=1}^N$ with N labeled graphs G_i and labels $y_i \in \mathcal{Y}$, our goal is learning a graph classifier $f_\theta : \mathcal{G} \to \mathcal{Y}$ parameterized by θ that is robust to graph perturbations.

Formally, the graph classifier f_θ aims to minimize the empirical risk:

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(G_i), y_i) \tag{2}$$

where $\ell(\cdot)$ is a loss function (e.g. cross-entropy). However, standard empirical risk minimization often leads to models that lack robustness and stability. To enforce this stability property, we propose adding an explicit stability regularization term:

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(G_i), y_i) + \lambda R(G_i) \tag{3}$$

where $R(\cdot)$ measures the stability between predictions on the original and perturbed graphs, and λ controls the regularization strength. In order to optimize Equation (5), we need to address the following issues:

1. How can we implement reliable stability constraints R within graph neural network models to enhance robustness?

2. How can we balance the trade-off between stability constraints and classification objectives, to maximize robustness gains while maintaining classification performance?

We will introduce our proposed model in the next section, and discuss how we address the above two challenges.

3.2. Model

3.2.1. Temporal Stability Regularization

Graph neural networks (GNNs) encode graph structured data $G = (V, E)$ into node representations Zv via message passing:

$$Z_v^{(t)} = f(Z_v^{(t-1)}, \text{aggregate}_{u\in N(v)} Z_u^{(t-1)}) \tag{4}$$

where $Zv^(t)$ is the representation of node v at layer t. After K layers, a readout function R aggregates the node representations into a graph-level embedding:

$$Z_G^{(t)} = R(\{Z_v^{(t)} | v \in G\}) \tag{5}$$

This graph representation $Z_G^{(t)}$ summarizes the structure of G at training epoch t. To improve robustness, we apply temporal

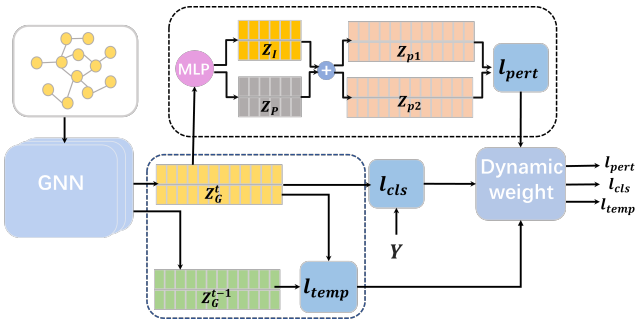


Fig. 2. **Model architecture.** The part inside the upper dotted line is the perturbation consistency stability module. The part within the lower dotted line is the temporal consistency stability module.

stability regularization between $Z_G^{(t)}$ at the current epoch and $Z_G^{(t-1)}$ from the prior epoch:

$$\mathcal{L}_{temp} = \begin{cases} |Z_G^{(t)} - Z_G^{(t-1)}| - 0.5 & \text{if } |Z_G^{(t)} - Z_G^{(t-1)}| > 1 \\ 0.5(Z_G^{(t)} - Z_G^{(t-1)})^2 & \text{if } |Z_G^{(t)} - Z_G^{(t-1)}| < 1 \end{cases} \tag{6}$$

$\mathcal{L}_{temp}$ allows small evolving differences between epochs but penalizes large inconsistent shifts in the graph representations. This allows small evolving changes in Z_G over epochs but heavily penalizes large inconsistent shifts. By constraining dramatic representation changes over time, temporal stability regularization focuses learning on robust invariant graph features and relies less on ephemeral fragile patterns.

3.2.2. *Perturbation Stability Regularization*

We decompose the graph representation Z_G into invariant Z_I and perturbed Z_P components using a learnable mask M, which is obtained by feeding Z_G through a multilayer perceptron (MLP) and sigmoid activation:

$$M = \text{MLP}(\sigma(Z_G)) \tag{7}$$

where $\sigma(x)$ is the sigmoid function .This allows M to learn to select robust invariant features and suppress fragile patterns based on the input Z_G. We then generate two perturbed samples by randomly adding Z_I with different Z_P:

$$Z_{pi} = Z_I + Z_{Pi} \tag{8}$$

We enforce consistency between two Z_{pi} stay consistent:

$$\mathcal{L}_{pert} = |Z_{p1} - Z_{p2}|^2 \tag{9}$$

By decomposing Z_G into stable Z_I and variable Z_P, and explicitly regularizing their perturbation consistency, the model becomes more robust. To dynamically adapt stability constraints for graph classification tasks across different domains and interference levels, so that temporal stability and perturbation stability can enhance model robustness without compromising model classification performance, we use multi-objective optimization for the graph classification objective and consistency constraints to achieve Pareto optimality of classification performance and stability [17]. Specifically, we compute gradients for three loss functions separately and dynamically calculate weights according to the gradients via the Frank-Wolfe algorithm [18] to find a common descent direction. We recalculate new weights according to this method for each propagation, to realize adaptive stability constraints.

4. EXPERIMENT

4.1. Datasets and Baselines

To demonstrate that our method is applicable to datasets with different biases and from different domains, we conduct experiments on synthetic datasets MNIST [19], KUZU [19], Fashion [19] with different bias levels (b) and real molecular graphs (MUTAG, PROTEIN, NCI-1) and social network graphs (IMDB-BINARY) from Tudateset[20].

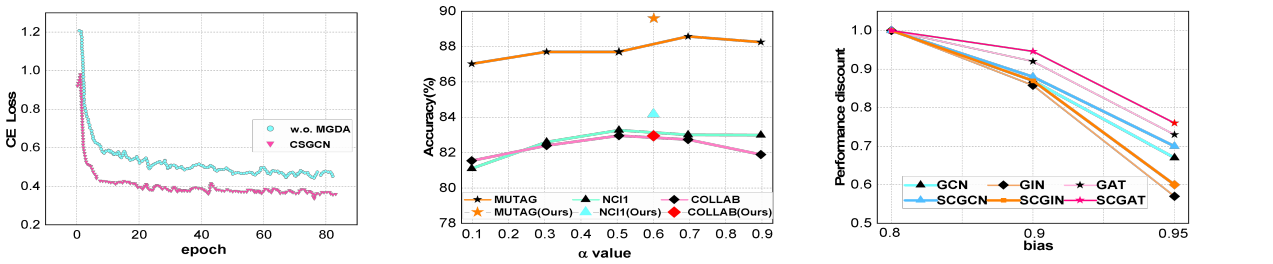
We compared advanced graph classification methods (DIR[7], Disc[16], SIB[6]) incorporating different insights into graph classification We conducted experiments on different architectures, including GCN[11], GIN[21] and GAT[12].

4.2. Performance Evaluation

The results of our method on real datasets are shown in Table 1. It can be seen that our model outperforms the baseline on molecules MUTAG, protein, and NCI-1), social network (IMDB-BINARY). This shows that our improvements are applicable to general graph data with different properties in various fields. The results in Table 2 illustrate that, compared with SOTA GNN graph classification methods (SIB[6] and DISC[16]), We obtain average performance gains of 8.2% and 2.6%. Our method has shown effective improvement on datasets with different degrees of deviation and is more robust to environmental interference than previous methods. Experiments on both sets of data sets demonstrate that our method is suitable for different GNN architectures.

4.3. Analysis

Subfigure (a) compares the training dynamics of the classification loss between SCGCN and its manually weighted variant on MUTAG. The blue scatter points are the best-performing settings under manual weight tuning. Compared to manually finding the trade-off, dynamically seeking the trade-off through multi-objective optimization achieves a faster decline of the classification loss. This demonstrates the facilitation of the dynamically balanced stability constraints on the graph classification task. Subfigure (b) compares the



(a) Training dynamics on MUTAG. (b) Parameter sensitivity analysis. (c) Performance discounts on different biased datasets.

Fig. 3. Futher analysis on some datasets.

Table 1. Experiments on datasets with varying degrees of bias .

Dataset	MNIST			KUZU			Fashion			Avg.
bias	**0.8**	**0.9**	**0.95**	**0.8**	**0.9**	**0.95**	**0.8**	**0.9**	**0.95**	-
DISC	77.40±1.23	77.28±2.45	54.23±3.67	52.43±0.98	47.60±3.51	34.75±1.23	60.59±1.34	61.92±1.76	59.33±1.23	58.03±1.98
DIR	45.13±2.15	37.29±1.87	23.84±2.56	41.08±1.23	37.27±2.98	32.06±1.45	60.48±2.34	58.25±1.76	52.42±2.13	42.39±2.04
SIB	70.27±1.67	67.29±2.11	44.24±1.87	47.26±1.56	42.60±2.34	33.75±1.23	60.54±1.87	60.09±2.56	55.88±1.45	55.01±1.78
GCN	74.08±1.78	64.68±2.13	49.87±2.34	52.17±1.67	41.25±1.98	31.34±1.56	67.59±2.11	61.39±1.89	54.54±2.04	55.87±1.91
GIN	73.88±2.33	63.41±1.67	42.40±1.89	49.65±2.02	38.95±1.78	28.83±1.34	65.81±1.56	59.84±2.13	53.28±1.45	52.93±1.76
GAT	84.48±1.34	78.32±1.12	62.38±0.87	62.11±0.96	47.98±0.67	37.38±1.01	71.59±0.78	64.76±1.23	56.24±2.06	63.38±1.02
SCGCN	76.58±0.67	67.18±0.89	53.37±0.45	53.67±0.56	46.75±0.78	35.84±0.34	70.09±0.91	63.89±0.67	57.04±0.82	59.53±0.71
SCGIN	75.88±0.56	65.41±0.79	44.40±0.67	51.65±0.81	40.95±0.62	30.83±0.48	67.81±0.73	61.84±0.91	55.28±0.83	54.93±0.69
SCGAT	86.04±0.62	81.45±0.74	69.42±0.88	63.64±0.71	49.83±0.93	35.33±0.56	71.76±0.85	65.98±0.73	58.56±0.68	65.38±0.77

Fig. 4. T-SNE analysis on the MINIST bias = 0.8 dataset, from left to right are DIR, GCN, and SCCCN.

Table 2. Experiments on real datasets.

Dataset	**MUTAG**	**NCI1**	**PROTEINS**	**COLLAB**	**Avg.**
GCN	88.20±7.33	82.97±2.34	75.65±3.24	81.72±1.64	82.14
GIN	89.42±7.40	82.71±1.52	76.21±3.83	82.08±1.51	82.61
GAT	88.58±7.54	82.11±1.43	75.96±3.26	81.42±1.41	81.77
SCGCN	89.39±6.54	84.06±1.81	75.12±3.36	83.06±1.85	82.91
SCGIN	90.12±7.72	83.87±1.63	75.76±3.93	82.75±1.29	83.13
SCGAT	91.55±6.64	84.06±1.81	76.01±2.49	82.28±1.29	83.48

performance between the two methods on three datasets from different domains. We fix the CE loss weight to 1, α is the weight for the temporal constraint, and 1-α is the weight for the invariance constraint. It can be observed that no matter how the relative ratio of the two constraints changes, our method always achieves optimal performance. This shows that the stability constraint strategies can automatically find the best trade-off for datasets from different domains, having better adaptability, while manual tuning of the trade-off parameters can hardly adapt to cross-domain situations. Subfigure (c) shows the performance discount of different methods as the imbalance ratio of the MINIST dataset gradually increases from 0.8 to 0.95. Stability constraints can reduce the performance discount brought by the gradually increased dataset bias for all GNN architectures. This demonstrates that the enhanced robustness from stability constraints alleviates the interference brought by dataset imbalance. Fig. 4 explains a potential reason why stability constraints enhance graph classification robustness. The three subfigures from left to right are respectively the t-SNE visualizations of the representations of DIR, GCN, and DSGCN on the MINIST dataset with bias=0.8. Stability constraints significantly overcome the influence brought by the confusing background, and clearly separate the different digits into categories.

5. CONCLUSION

We propose a new framework to improve the robustness of graph classification by incorporating stability constraints, and implement the stability constraints through temporal consistency and perturbation consistency. By adaptively finding the trade-off between stability and classification performance, our method can work with graph data from different domains and interference levels. Our research provides a new perspective to enhance the robustness of graph classification.

6. REFERENCES

[1] Han Xie, Jing Ma, Li Xiong, and Carl Yang, "Federated graph classification over non-iid graphs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18839–18852, 2021.

[2] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang, "Hierarchical graph pooling with structure learning," *arXiv preprint arXiv:1911.05954*, 2019.

[3] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[4] Liang Yao, Chengsheng Mao, and Yuan Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, pp. 7370–7377.

[5] Artur M Schweidtmann, Jan G Rittig, Andrea Konig, Martin Grohe, Alexander Mitsos, and Manuel Dahmen, "Graph neural networks for prediction of fuel ignition quality," *Energy & fuels*, vol. 34, no. 9, pp. 11395–11407, 2020.

[6] Junchi Yu, Jie Cao, and Ran He, "Improving subgraph recognition with variational graph information bottleneck," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19396–19405.

[7] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua, "Discovering invariant rationales for graph neural networks," *arXiv preprint arXiv:2201.12872*, 2022.

[8] Saurabh Verma and Zhi-Li Zhang, "Stability and generalization of graph convolutional neural networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1539–1548.

[9] Antonio Marino, Claudio Pacchierotti, and Paolo Robuffo Giordano, "On the stability of gated graph neural networks," *arXiv preprint arXiv:2305.19235*, 2023.

[10] Shufei Zhang, Kaizhu Huang, and Zenglin Xu, "Rethinking model robustness from stability: a new insight to defend adversarial examples," *Machine Learning*, vol. 111, no. 7, pp. 2489–2513, 2022.

[11] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

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

[13] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in neural information processing systems*, vol. 31, 2018.

[14] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar, "Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach," in *Proceedings of the Web Conference 2020*, 2020, pp. 673–683.

[15] Xiaojun Xu, Hanzhang Wang, Alok Lal, Carl A Gunter, and Bo Li, "Edog: Adversarial edge detection for graph neural networks," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2023, pp. 291–305.

[16] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang, "Debiasing graph neural networks via learning disentangled causal substructure," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24934–24946, 2022.

[17] Ozan Sener and Vladlen Koltun, "Multi-task learning as multi-objective optimization," *Advances in neural information processing systems*, vol. 31, 2018.

[18] Martin Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *International conference on machine learning*. PMLR, 2013, pp. 427–435.

[19] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha, "Deep learning for classical japanese literature," *arXiv preprint arXiv:1812.01718*, 2018.

[20] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann, "Tu-dataset: A collection of benchmark datasets for learning with graphs," *arXiv preprint arXiv:2007.08663*, 2020.

[21] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*.