Robust Heterophily Graph Learning via Uniformity Augmentation

Xusheng Yang
yangxs@stu.pku.edu.cn
Peking University
Shenzhen, China

Zhengyu Chen
chenzhengyu@zju.edu.cn
Meituan
Beijing, China

Yuexian Zou*
zouyx@pku.edu.cn
Peking University
Shenzhen, China

ABSTRACT

Graphs serve as fundamental representations for a diverse array of complex systems, capturing intricate relationships and interactions between entities. In many real-world scenarios, graphs exhibit non-homophilous, or heterophilous, characteristics, challenging traditional graph analysis methods rooted in homophily assumptions. Recent heterophilous methods frequently struggle with noise in node attributes, which can degrade the quality of graph representations and affect downstream task performance. Common graph augmentations, while useful, often introduce bias and irrelevant noise. This paper proposes a novel method, Robust Heterophily Graph Learning via Uniformity Augmentation (RHGL-UA), which incorporates uniformity in the augmentation process through controlled random perturbations. This approach ensures a more uniform distribution of representations across different layers of the model. By adapting to data variations and learning more diverse information, RHGL-UA significantly improves performance on downstream tasks and stands out as the first practical robust heterophily graph method using representation augmentation with a theoretical guarantee. Extensive experiments demonstrate the merit of our proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**; **Artificial intelligence**; **Neural networks**; • **Information systems** → **Data mining**.

KEYWORDS

Graph Neural Network, Heterophily Graphs, Data Augmentation, Uniformity Representation

ACM Reference Format:
Xusheng Yang, Zhengyu Chen, and Yuexian Zou. 2024. Robust Heterophily Graph Learning via Uniformity Augmentation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679991

*Corresponding author.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679991

1 INTRODUCTION

Graphs are robust mathematical structures with applications spanning various fields, including social networks, recommendation systems, biological networks, and information propagation analysis [4, 23, 29]. In numerous real-world situations, graphs inherently exhibit diversity, comprising nodes and edges that symbolize different entities and their interconnections [5, 30]. These relationships often display non-homophilous, or heterophilous, traits, where nodes with other different attributes are linked, posing challenges to traditional analysis techniques based on homophily assumptions.

Homophily, the propensity of nodes with similar attributes to connect, is widely studied in graph theory and social network analysis. However, many real-world graphs contradict this assumption due to the intricate interplay of diverse attributes. For instance, in a social network, individuals may connect not solely based on shared characteristics, but also a combination of diverse factors, including occupations, and geographic locations. Traditional analysis methods [3, 6, 11, 19, 22] designed for homophilous graphs fail to capture the underlying patterns within such heterophily graphs.

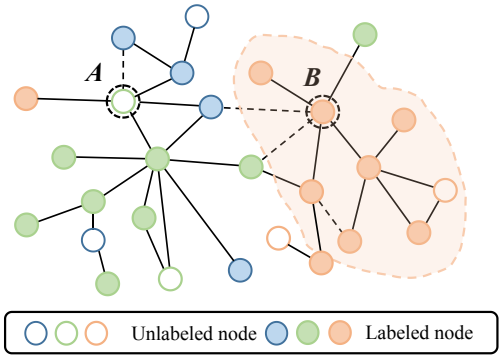


Figure 1: The node classification in heterophilic graphs. The dotted line connected with nodes A and B denotes some noisy edges while the solid line is the normal edge.

In recent years, there has been an increasing acknowledgment of the necessity to expand graph analysis techniques to accommodate heterophily graphs [21]. However, these methods often fail to address the potential noise in the attributes as shown in Fig.1, the dotted lines denote noisy connections, leading to subpar performance in subsequent tasks. Such bias can degrade the quality of graph representations and subsequently impair performance in downstream tasks. Recent empirical design of graph-specific augmentations, such as edge permutation and node dropout, has been

one approach to enhance performance [27]. Yet, these augmentations introduce irrelevant noise and can lead to biased embeddings that do not accurately reflect the true structure of the graph.

To alleviate the noise that may negatively impact representation quality or the high bias of the node embedding via graph augmentation, this paper introduces the Robust Heterophily Graph Learning via Uniformity Augmentation method (RHGL-UA). This method aims to learn robust representations for heterophilous graphs by enhancing the model's robustness through the introduction of random perturbations. Specifically, RHGL-UA enhances the model's learning capabilities by ensuring a more uniform distribution of representations, achieved through the strategic addition of random noise vectors at each layer of the model. The proposed uniformity augmentation perturbations enable the model to adapt to a broader range of data variations and learn more diverse information during the training process, thereby improving the model's performance on downstream tasks. Our contributions are summarized as follows:

- To alleviate the noisy edges and highly biased node embeddings that may negatively impact representation quality, we propose the RHGL-UA method to learn the robust representation for heterophilous graphs, enhancing model robustness by ensuring a more uniform distribution of representations.
- Our uniformity augmentation approach improves the learning capability of the model, achieving a more uniform representation distribution by incorporating controlled perturbations at each representation layer.
- We provide a complexity analysis, theoretical justification, and extensive experimental results to demonstrate the efficacy and advantages of our proposed method.

2 RELATED WORK

Heterophilic Graphs. There has been significant research interest in enhancing the performance of graph neural networks on heterophilic graphs [30]. Geom-GCN [17] is a geometric aggregation scheme for modeling structural information and long-range dependencies ability in low-homophily graphs. Mixhop [1] designs a graph convolutional layer that uses multiple powers of the adjacency matrix to learn general neighborhood mixing relationships. ProtoGNN [8] augments node features with structural information by learning multiple class prototypes for acquiring a global message. LINKX [13] separates the embedding in the adjacent A and node feature X, then operates their concatenated embedding by multi-layer perceptrons and skip connection. However, these works ignore that heterophilous graphs can contain noise that may negatively impact representation quality. Different from these works, we consider learning robust representations for heterophilous graphs.

Data Augmentation. Data augmentation usually refers to the augmentation performed in the input space. Attribute masking, edge permutation, and node dropout are common graph augmentation strategies [25]. Some works [9, 12] have suggested that Gaussian data augmentation could utilize its robustness to adversarial perturbations. Gaussian Data Augmentation (GDA) is proven to help explore any direction to smooth the model confidence and even improve accuracy [26]. However, these methods of graph augmentation are verified to have a negative effect with the high bias embeddings [27]. Inspired by the work on understanding representation learning through alignment and uniformity on the hypersphere [20], we aim to design a novel method to ensure a more uniform distribution of representations across the layers of the model to alleviate the noise. Different from these works, we learn the uniformity property prefers a feature distribution that preserves as much information of data as possible and directly optimizing uniformity metrics often leads to better representations.

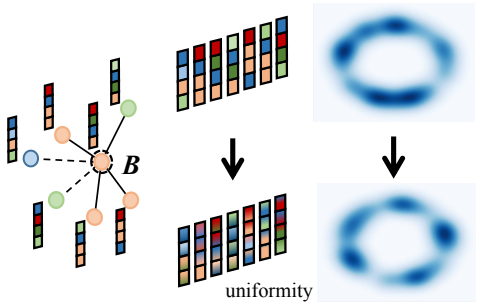


Figure 2: Our method can make embeddings of nodes more uniform in the orange region of Fig.1. The representations of node B are augmented from isolation towards uniformity.

3 METHOD

Notations of Heterophily. Let $G = (V, E)$ be a graph with n nodes. Further let each node $u \in V$ have a class label $k_u \in \{0, 1, ..., C-1\}$, and C_k denotes the set of nodes in k class. Then we give the measure by referring to former work LINKX [13], which is defined as: $h_E = \frac{1}{C-1}\sum_{k=0}^{C-1}[h_k - \frac{|C_k|}{n}]_+$, where $[x]+ = max(x, 0)$, and h_k is the class-wise homophily metric $h_k = \frac{\sum_{u\in C_k} d_u^{k_u}}{\sum_{u\in C_k} d_u}$.

Proposed Method. Heterophilic Graph Learning methods handle heterophilic graph data that take into account the diversity of node and edge types in the graph. However, these methods have certain limitations. Robust representations help uncover these hidden patterns, shedding light on unconventional relationships that contribute to a deeper understanding of the system. Learning robust representations aids in modeling how information spreads across nodes with varying attributes, leading to more accurate simulations of processes. These representations provide a deeper understanding of the complex dynamics within such graphs.

Inspired by previous work [20], we believe that it is reasonable to understand the introduction of random noise from the perspective of uniformity representations. From this perspective, the reason why adding random noise to the representation can obtain a more even representation distribution is that the values of some dimensions in random noise are close to zero, which plays a role in feature selection. As shown in Fig.2, our method makes the representations of node B augmented from isolation towards uniformity.

Firstly, we define the representation of node i at the h-th layer as ${v_i}^{(h)}$. Then, we introduce a uniform random perturbation $\Delta_i^{(h)'}$. This perturbation is random, which can help our model better adapt to changes in the data, thereby improving the robustness

of the model. We adjust the perturbation using the sign function $sign(v_i^{(h)})$, making the direction of the perturbation consistent with the direction of the node representation. Finally, we obtain the new representation of node i at the h-th layer after introducing the random perturbation, denoted as $(v_i^{(h)})'$, which is defined in:

$$(v_i^{(h)})' = v_i^{(h)} + sign(v_i^{(h)})\dot{\Delta}_i^{(h)'}, \tag{1}$$

Across all layers, we average the representations of node i to obtain its representation at the final layer H, denoted as $v_i^{(H)}$. The purpose is to make the node representation integrate information to enhance its globality. In addition, we introduce a global random perturbation $\Delta^{(H-1)}$ and adjust it using the adjacency matrix A. The adjacency matrix A can reflect the connection relationships between nodes. By multiplying with the adjacency matrix, we can incorporate the connection relationships of nodes into their representations. Finally, we obtain the representation of node i at the final layer H, denoted as $v_i^{(H)}$, which is calculated as follows:

$$\begin{aligned} v_i^{(H)} &= \frac{1}{H}\sum_{h=1}^{H} v_i^{(h)}, \\ v_i^{(h)} &= v_i^{(h-1)} + \Delta^{(h-1)}, \end{aligned} \tag{2}$$

Introducing random perturbations can make the model meet a wider range of data variations. This allows the model to learn more diverse information in the training process. Moreover, these random perturbations disrupted specific patterns in the data, preventing the model from relying solely on those patterns. As a result, the model learns a more uniform representation of the data distribution.

Complexity Analysis. The main computational cost of our method comes from two parts: the calculation of the node representations and the addition of random noise vectors. For the calculation of node representations, the complexity is $O(Nd)$, where N is the number of nodes in the graph and d is the dimension of the node representation. For the addition of random noise vectors, the complexity is $O(Nd)$ as well. This is because we need to add a noise vector to the representation of each node, and the noise vector has the same dimension as the node representation.

Therefore, the overall complexity of our method is $O(Nd) + O(Nd) = O(Nd)$, which is linear with respect to the number of nodes and the dimension of the node representation. For space complexity, our method requires $O(Nd)$ space to store the node representations and the noise vectors. Therefore, the space complexity is also $O(Nd)$, which is manageable for large-scale graphs.

Theoretical Justification. We aim to explain the theoretical basis for the robustness enhancement of heterophily graph learning through our proposed method. Some work [14, 28] suggest that a flatter loss landscape contributes to model robustness. Building upon this insight, we argue that our method achieves this flattening effect, thereby enhancing robustness. Inspired by previous work [16] that combines the sharpness of loss landscape and PAC-Bayes theory [15], we derive bounds on the expected error under certain assumptions. Assuming that the prior distribution P over the weights is a zero mean, σ^2 variance Gaussian distribution, with probability at least $1-\delta$ over the draw of M graphs, the expected error of the network can be bounded as:

$$\begin{aligned} \mathbb{E}_{\{G_i\}_{i=1}^{M},\Delta}[\mathcal{L}(\theta+\Delta)] <= \mathbb{E}_{\Delta}[\mathcal{L}(\theta+\Delta)] \\ +4\sqrt{\frac{KL(\theta+\Delta\|P)+\ln\frac{2M}{\delta}}{M}} \end{aligned} \tag{3}$$

We choose a uniform perturbation within embedding layers, given that the input is fixed and it is equivalent to setting the perturbation Δ to the weight θ concerning its magnitude $\sigma = \alpha\|\theta\|$. Besides, we substitute $\mathbb{E}_{\Delta}[\mathcal{L}(\theta+\Delta)]$ with $\mathcal{L}(\theta) + \mathbb{E}_{\Delta}[\mathcal{L}(\theta+\Delta)] - \mathcal{L}(\theta)$. Then, we could rewrite Eq.3 as:

$$\begin{aligned} \mathbb{E}_{\{G_i\}_{i=1}^{M},\Delta}[\mathcal{L}(\theta+\Delta)] <= \mathcal{L}(\theta) + \underbrace{\{\mathbb{E}_{\Delta}[\mathcal{L}(\theta+\Delta)] - \mathcal{L}(\theta)\}}_{\text{Expected sharpness}} \\ +4\sqrt{\frac{1}{M}(\frac{1}{2\alpha}+\ln\frac{2M}{\delta})} \end{aligned} \tag{4}$$

It is obvious that $\mathbb{E}_{\Delta}[\mathcal{L}(\theta+\Delta)] <= \max_{\Delta}[\mathcal{L}(\theta+\Delta)]$ and the third term $4\sqrt{\frac{1}{M}(\frac{1}{2\alpha}+\ln\frac{2M}{\delta})}$ is a constant. Thus, our method improves the worst-case sharpness of the loss landscape to the bound of the expected error, demonstrating an enhancement in robustness.

4 EXPERIMENTS

Datasets. Our experimental evaluation is conducted on seven diverse datasets, which exhibit heterophilous characteristics. These include a large-scale dataset, Penn94 which was introduced by Lim et al. [13]. We adhere to the original train/validation/test splits as proposed in their study. Additionally, we assess the performance of our model on six other homophilous datasets: Film, Squirrel, Chameleon, Cornell, Texas, and Wisconsin [17]. These datasets are derived from various networks, including a subgraph of the film-director-actor-writer network, a webpage dataset from computer science domains, and specific topics from Wikipedia.

Experimental Setup. In the experimental setup, we benchmark our method, RHGL-UA, against 11 baseline methods. These include methods that rely solely on node features, such as MLP and LINK, traditional GNN methods like GCN [11], GAT [19], GCNJK [24], APPNP [10], and non-homophilous methods such as MixHop [1], GPR-GNN [7], GCNII [2], LINKX [13]. Our RHGL-UA backbone is LINKX [13]. All methods are trained using full batch gradient descent for 500 epochs, and the test performance is reported based on the highest validation performance. The performance metric used in our experiments is classification accuracy.

Performance Comparison. To substantiate the efficacy of our proposed method, we have conducted a comparative analysis with the aforementioned approaches across a range of heterophilic datasets. The comparative outcomes are systematically documented in Table 1. As depicted in the final row of Table 1, our method, RHGL-UA, surpasses the performance of state-of-the-art techniques on the large-scale Penn94 dataset and achieves competitive (underline is suboptimal) if not the best performance on other benchmark datasets, except Texas. Specifically, our approach has demonstrated a 0.8% enhancement in performance on the Penn94 dataset and has outperformed LINKX, a leading method for non-homophilous graphs, by 0.9%,1.3%, 2.7%,2.2%, and 6.5%, respectively. These findings suggest that RHGL-UA can augment performance on both

Table 1: The test accuracy is presented in all datasets. We give the heterophily metric h_E for all datasets. They show that our datasets are indeed heterophily. (M) denotes some (or all) hyperparameter settings run out of memory.

	Penn94 (.046)	Film (.011)	Squirrel (.025)	Chameleon (.062)	Cornell (.047)	Texas (.001)	Wisconsin (.094)
#Nodes	41,554	7,600	5,201	2,277	251	183	183
MLP	73.61 ± 0.40	34.50 ± 1.77	31.10 ± 0.62	41.67 ± 5.92	67.03 ± 6.16	70.81 ± 4.44	71.77 ± 5.30
LINK	80.79 ± 0.49	23.82 ± 0.30	59.75 ± 0.74	64.21 ± 3.19	44.33 ± 3.63	51.89 ± 2.96	54.90 ± 1.39
GCN	82.47 ± 0.27	26.86	23.96	28.18	52.70	52.16	45.88
GAT	81.53 ± 0.55	28.45	30.03	42.93	54.32	58.38	49.41
GCNJK	81.63 ± 0.54	27.41	35.29	57.68	57.30	56.49	48.82
APPNP	74.33 ± 0.38	32.41	34.91	54.3	73.51	65.41	69.02
MixHop	83.47 ± 0.71	32.22 ± 2.34	43.80 ± 1.48	60.50 ± 2.53	73.51 ± 6.34	**77.84 ± 7.73**	75.88 ± 4.90
GPR-GNN	81.38 ± 0.16	33.12 ± 0.57	54.35 ± 0.87	62.85 ± 2.90	68.65 ± 9.86	76.22 ± 10.19	75.69 ± 6.59
GCNII	82.92 ± 0.59	34.36 ± 0.77	56.63 ± 1.17	62.48	76.49	77.84	81.57
Geom-GCN-P	(M)	31.63	38.14	60.90	60.81	67.57	64.12
LINKX	84.71 ± 0.52	36.10 ± 1.55	61.81 ± 1.80	68.42 ± 1.38	77.84 ± 5.81	74.60 ± 8.37	75.49 ± 5.72
RHGL-UA (Ours)	**85.51 ± 0.43**	**37.00 ± 0.91**	**63.15 ± 0.76**	**71.18 ± 2.58**	**80.00 ± 10.74**	76.22 ± 6.45	**81.96 ± 6.11**

standard and large-scale datasets, thereby underscoring its robustness and capacity for generalization across diverse graph structures.

Evaluation of Robustness. To evaluate the robustness, we have subjected it to a series of controlled adversarial perturbations. Specifically, we have introduced noise into the embedding generated by the LINKX model on the Penn94 dataset. The LINKX model was trained under its standard configuration, without incorporating noise perturbations, and was subsequently tested with the perturbed embedding. In contrast, our method adhered to the original training and testing conditions without modification. The results in Fig. 3 reveal a consistent performance advantage of our method over the LINKX model across a spectrum of noise intensities. This comparative superiority underscores the robustness of our approach in the face of adversarial perturbations, suggesting a heightened resilience to noise and potential adversarial attacks.

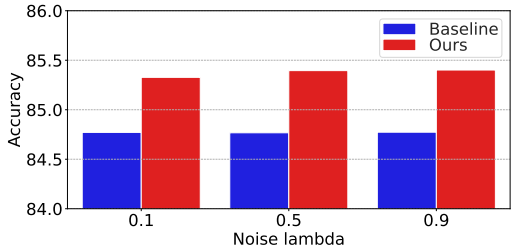
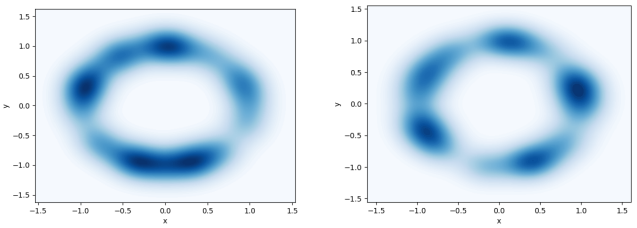


Figure 3: Evaluation of robustness. Baseline is LINKX model.

Uniformity Analysis. To ascertain the uniformity of embeddings, we have conducted a visual analysis of the Chameleon dataset following the concatenation operation within the network. Specifically, we have employed t-distributed Stochastic Neighbor Embedding (t-SNE) [18] to reduce the dimensionality of embeddings to two dimensions, facilitating visualization. To elucidate the distribution of uniformity, as depicted in Fig. 4, the x and y axes represent the scales of the two-dimensional features after unit sphere normalization. Subsequently, we have utilized Kernel Density Estimation (KDE) to estimate the probability density of the representations. Upon observation, it is evident that embeddings derived from the LINKX method, as shown in Fig. 4 (a), are more concentrated at the lower portion of the contour, indicating a less uniform distribution. In contrast, embeddings obtained using our method, as illustrated in Fig. 4 (b), display a more uniform distribution across the unit circle, suggesting a more evenly spread representation.



(a) embedding in baseline (b) embedding with noise

Figure 4: Visualization of the embedding.

5 CONCLUSION

This paper studies the problem of learning robust representations from heterophilous graphs, which are characterized by the presence of diverse and non-homophilous connections. This paper introduces a novel method, RHGL-UA, designed to learn robust representations for heterophilous graphs. Our approach enhances model robustness through the introduction of controlled random perturbations, achieving a more uniform distribution of representations. Extensive experiments validate the efficacy of RHGL-UA, demonstrating its capability to adapt to diverse graph structures and improve performance on downstream tasks. However, our method's computational complexity might limit its applicability to extremely large-scale graphs, which is a potential area for future improvement.

ACKNOWLEDGMENTS

This paper was partially supported by NSFC (No:62176008).

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*. PMLR, 21–29.

[2] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International conference on machine learning*. PMLR, 1725–1735.

[3] Zhengyu Chen, Jixie Ge, Heshen Zhan, Siteng Huang, and Donglin Wang. 2021. Pareto Self-Supervised Training for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13663–13672.

[4] Zhengyu Chen, Teng Xiao, and Kun Kuang. 2022. BA-GNN: On Learning Bias-Aware Graph Neural Network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 3012–3024.

[5] Zhengyu Chen, Teng Xiao, Kun Kuang, Zheqi Lv, Min Zhang, Jinluan Yang, Chengqiang Lu, Hongxia Yang, and Fei Wu. 2024. Learning to Reweight for Generalizable Graph Neural Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8320–8328.

[6] Zhengyu Chen, Teng Xiao, Donglin Wang, and Min Zhang. 2024. Pareto Graph Self-Supervised Learning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6630–6634.

[7] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2020. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988* (2020).

[8] Yanfei Dong, Mohammed Haroon Dupty, Lambert Deng, Yong Liang Goh, and Wee Sun Lee. 2022. ProtoGNN: Prototype-Assisted Message Passing Framework for Non-Homophilous Graphs. (2022).

[9] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. 2019. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513* (2019).

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

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

[12] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. 2019. Certified adversarial robustness with additive noise. *Advances in neural information processing systems* 32 (2019).

[13] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems* 34 (2021), 20887–20902.

[14] Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. 2020. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems* 33 (2020), 21476–21487.

[15] David A McAllester. 1999. PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*. 164–170.

[16] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. *Advances in neural information processing systems* 30 (2017).

[17] Haochen Pei, Jing Wei, Kejun Chang, Yatao Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. *arXiv preprint arXiv:2002.05287* (2020).

[18] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

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

[20] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[21] Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. 2022. Decoupled self-supervised learning for graphs. *Advances in Neural Information Processing Systems* 35 (2022), 620–634.

[22] Teng Xiao, Zhengyu Chen, Donglin Wang, and Suhang Wang. 2021. Learning how to propagate messages in graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1894–1903.

[23] Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. 2024. Simple and asymmetric graph contrastive learning without augmentations. *Advances in Neural Information Processing Systems* 36 (2024).

[24] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*. PMLR, 5453–5462.

[25] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. *Neural Information Processing Systems,Neural Information Processing Systems* (Jan 2020).

[26] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. 2017. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 39–49.

[27] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. 2022. COSTA: covariance-preserving feature augmentation for graph contrastive learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2524–2534.

[28] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. 2020. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060* (2020).

[29] Jiong Zhu, Junchen Jin, Donald Loveland, Michael T Schaub, and Danai Koutra. 2022. How does heterophily impact the robustness of graph neural networks? theoretical connections and practical implications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2637–2647.

[30] Jiong Zhu, Yujun Yan, Mark Heimann, Lingxiao Zhao, Leman Akoglu, and Danai Koutra. 2023. Heterophily and Graph Neural Networks: Past, Present and Future. *IEEE Data Engineering Bulletin* (2023).