Understanding the Impact of Graph Diffusion on Robust Graph Learning

Kutlu Emre Yilmaz
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
`kutluyilmaz@vt.edu`

Satvik Chekuri
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060
`satvikchekuri@vt.edu`

Abstract

In this paper, we propose a defense mechanism against adversarial attacks for graph learning tasks. We interpret the problem of malicious node and edge perturbations that leads to a decline in classification accuracy as a noise injection process. To improve the robustness of graph learning tasks on perturbed data, we propose a noise suppression methodology through which we obtain a smoother version of the original graph by applying graph diffusion convolution (GDC). To evaluate our proposed methodology, we use a global adversarial attack (meta-self attack), and present our results for different perturbation ratios. Our results show a significant increase in accuracy after applying GDC on graph data with less complex graph neural networks. The margin in performance improvement reduces with the increase in the complexity of the graph learning model. Our results are encouraging to further investigate the proposed approach under different adversarial attack types and on more complex graph neural networks (GNN) architectures using graph diffusion convolution as part of our proposed pre-processing pipeline. The code for this project is available at: https://github.com/satvikchekuri/DiffusionGraphRobustness.

1 Introduction

A general limitation of deep neural networks is their vulnerability to adversarial attacks (Szegedy, et al. 2013). In this scheme, an imperceptible perturbation introduced to input data in the form of a noise results in major deviations from the desired prediction results (Goodfellow, et al. 2014). This "intriguing property" of deep neural networks resulted in an "arms race" between attackers and defenders, which resulted in substantial amounts of research on robust learning from image, and text data (Xu, et al. 2020). With the recent success of deep learning models on graph-structured data, the robust training of deep graph learning models is an emerging field of research (Sun, et al. 2020).

Applications of graph neural networks is receiving increasing attention from both industry and academia. The exponential increase in the networked information and communication platforms (e.g., online social networks and computer networks) and the possible representation of relationships between entities in different domains as graphs points to the wide applicability of graph learning tasks on various domains. Therefore, developing robust and efficient training strategies for machine learning models on graphs has both social and technical merits that range from understanding and improving the society that we live-in to improving processing and storage of relational data (e.g., knowledge graphs, protein networks).

For robust training of graph neural networks, a substantial amount of work leverage adversarial training strategies (Wang, et al. 2019). In this scheme, the model is retrained on perturbed versions of the original data. Another defense strategy against adversarial perturbations is filtering of the training data with different data pre-processing and data selection techniques to detect and eliminate

Preprint. Under review.

the most attack-prone or attack-suspicious subsets of data. These include anomaly detection (Xu, et al. 2018), outlier detection (Ioannidis, et al. 2019), out-of-distribution detection (Zhang, et al. 2019) and importance sampling (Miller, et al. 2020).

Other methods modify the overall graph data without considering a specific attack model or detection mechanism and creates a final augmented graph data is resistant to perturbations. Some examples are training the model on low rank approximation of the original graph (Entezari, et al. 2020), leveraging a data augmentation scheme where the model is trained on nosier variants of the same graph (Fox, et al. 2019), training model on the subset of graph data with the most significant vertex degree and neighborhood features (Miller, et al. 2020).

Similarly, to improve the robustness of graph learning task, we propose a global graph modification/transformation scheme with an objective to reduce the effect of the most attack-prone nodes in the graph. Different from previous research, our approach considers graph diffusion (Klicpera, et al. 2019) as a potential data pre-processing tool that suppresses the effects of adversarial perturbations on graph data. Compared to other methods, our proposed scheme is independent of the underlying topology of the graph and does not require special feature selection or topological analysis of the original graph.

Our approach considers an adversarial perturbation as a form of specially crafted noise that is injected to dataset with an objective of increasing model loss and reducing the classification accuracy. In this sense, training the model on a noise resistant smoother variant of the graph, instead of the original graph, is expected to be less susceptible to adversarial perturbations. Analogous to smoothing in signal processing, graph diffusion contributes to reducing the effect of noise and improves the generalization performance of graph learning algorithms by providing a coarsened smoother version of the original graph (Klicpera, et al. 2019). Thus, this study investigates the impact of graph diffusion as a defense mechanism against adversarial attacks on graph learning tasks. Specifically, we explore whether graph diffusion, as a form of data pre-processing/transformation tool, applied to graph data, provides a robust training strategy that reduces susceptibility to adversarial perturbations.

2 Related Work

Adversarial Attacks: Szegedy, et al. (2013) discovered that local discontinuities on the decision surface of deep neural networks is susceptible large margin of prediction errors even for minor changes on input data distribution. Goodfellow, et al. (2014) introduced a proof-of-concept adversarial attack scheme using “fast gradient sign method". Zügner, et al. (2019) studied adversarial attacks on graph neural networks, specifically focusing on node attributes. Bojchevski, et al. (2019) and Sun, et al. (2020) devised attacks for unsupervised node embeddings. Zügner, et al. (2019) used meta-learning to design adversarial attacks on graph data. In our experiments, we test the robustness of our proposed defense approach against meta-learning based adversarial attacks of Zügner, et al. (2019).

Defense Strategies: Adversarial training is the most common defense approach (Goodfellow, et al. 2014). To improve model robustness, the model is retrained on the same dataset, but with varying strengths of perturbation applied. Data pre-processing filters out the most vulnerable parts of the dataset or attack-suspicious nodes and edges. For this purpose, several techniques, such as graph anomaly detection, out-of-distribution detection, outlier detection that transforms the graph are used (Sun, et al. 2020). Similarly, data modification techniques the transform graph, but in a generic way, without considering a specific attack or target type (Miller, et al. 2020). The core assumption of those techniques is to reduce the effects of less informative noisy components in graphs and to amplify the signal of the most informative vertices and edges.

Graph polynomial filters and graph neural networks: Defferrard, et al. (2016) introduced Chebnet, a convolutional graph neural network where convolution operations are approximated by a polynomial filter applied on on graph Laplacian. Graph attentions networks apply convolution operations on spatial domain and use attention to weight contributions of different nodes and edges to learning model (Veličković, et al. 2017). Simplifying graph convolutional networks is a purist approach: a simple low pass filter that operates on graph spectrum is followed by a logistic regression. Similarly, graph diffusion convolution (GDC) is a low pass filter that is designed as a polynomial filter of a random walk transition matrix of a graph; however, this method is non-local and allows for neighborhood aggregation of distant nodes (Klicpera, et al. 2019).

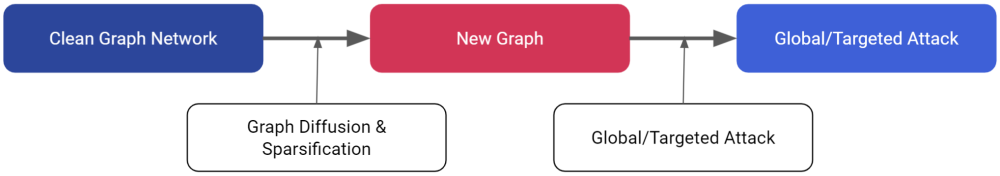


Figure 1: Graph pre-processing pipeline for robust training.

3 Methodology

The proposed pipeline is basically a noise resistant graph transformation. The original graph is transformed into a smoother noise reduced version of it using graph diffusion convolution. In principle, this acts as a low pass filter on the graph and reduces the effect of less informative nodes and edges which are natural targets of adversarial perturbations. Figure 1. gives an overview of our proposed graph pre-processing pipeline and training procedure. Compared to original graph, adversarial perturbations on coarsened graph, a result of graph diffusion, is expected to be less vulnerable to external injection of major structural irregularities due to smoothness constraints in action. Thus, compared to adversarial perturbations on original graph, adversarial perturbations on the coarsened graph is expected to have less training loss and increased classification accuracy.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the original graph with node set $\mathcal{V}$, edge set $\mathcal{E}$. Considering $\boldsymbol{A} \in \mathbb{R}^{NXN}$ is the adjacency matrix of $\mathcal{G}$, the generalized graph diffusion matrix by Klicpera, et al. (2019) is given as:

$$\mathbf{S} = \sum_{k=0}^{\infty} \theta^k \mathbf{T}^k$$

where $\boldsymbol{T}_{rw} = \boldsymbol{A}\boldsymbol{D}^{-1}$ is a row stochastic random walk transition matrix. Graph diffusion convolution transforms $\mathcal{G}$ to a new graph $\widetilde{\boldsymbol{S}}$ that is the sparsified version of $\boldsymbol{S}$ as shown in Figure 2. We propose the use of $\widetilde{\boldsymbol{S}}$ instead of $\mathcal{G}$ in graph learning tasks to improve the robustness of the learning algorithms in the presence of an adversary.

The overall goal of meta attack can be formulated as follows (Zügner, et al. (2019)):

$$\min_{\hat{G} \in \Phi(G)} \quad \mathcal{L}_{\text{atk}} \left(f_{\theta^*}(\hat{G}) \right) \quad \text{s.t.} \quad \theta^* = \arg\min_{\theta} \quad \mathcal{L}_{\text{train}} \left(f_{\theta}(\hat{G}) \right). \tag{1}$$

where $\mathcal{L}_{\text{atk}}$ is the loss function the attacker aims to optimize, and in the case of global and unspecific attacks, the attacker tries to decrease the generalization performance of the model on the unlabeled nodes.

4 Experiments and Results

As part of our experiments, we have performed node classification task on three different GNNs, and evaluated the classification accuracy before and after diffusion is applied to a perturbed graph network.

4.1 Dataset

Table 1 gives details about the five datasets we have used as part of our experiments along with the number of edges and nodes in each dataset. Cora, Cora_ML, Citeseer, and PubMed datasets are collections of publications related to CS/ML field, scientific, and medical respectively. PolBlogs dataset is a collection of political blogs. The data is split randomly into 10%/10%/80% for training/validation/test.

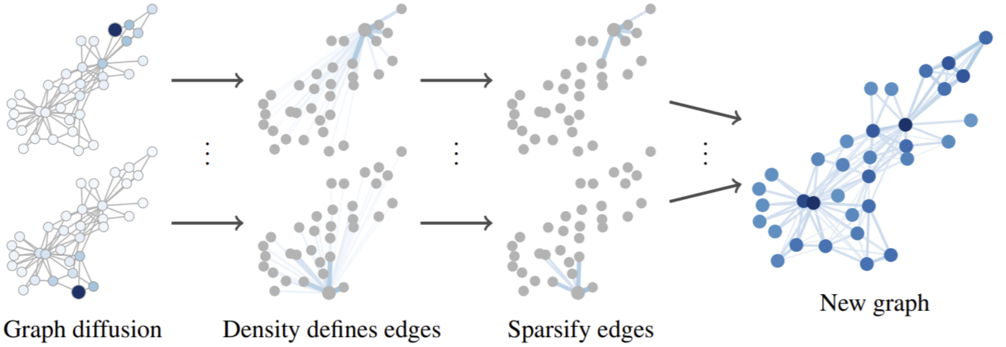


Figure 2: Illustration of graph diffusion convolution (GDC) (Klicpera, et al. (2019)).

Table 1: Datasets used for the experiments.

Dataset	**# of Nodes**	**# of Edges**
Cora	2,708	5,429
Cora_ML	2,810	7,981
Citeseer	3,312	4,732
PolBlogs	~1,500	~19,000
PubMed	19,717	44,338

4.2 Graph Neural Networks

As mentioned earlier, three different graph neural networks have been perturbed and later diffusion is applied to test the robustness. Following are the three networks.

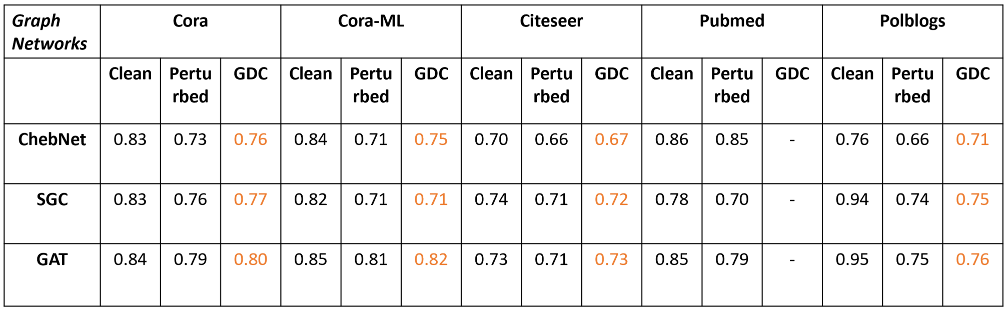
- ChebNet (Defferrard, et al. (2016)): It is an efficient generalization of CNNs to graphs using tools from graph signal processing.
- SGC (Wu, et al. (2019)): Simplifying Graph Convolution Networks is aimed at reducing the excessive complexity introduced by Graph Convolution Networks (GCN) through successively removing non-linearities and collapsing weight matrices between consecutive layers.
- GAT (Veličković, et al. (2017)): Graph Attention Network is a neural network architecture based on graph-structured data, leveraging masked self-attention layers.

These graph networks are trained individually and node classification is first performed on a clean graph. Next, adversarial attack of type Metattack (Zügner, et al. (2019)) is done on these three graphs. Finally, diffusion is performed on these three clean graphs and again attacks are introduced, and the final classification accuracy is recorded. These experiments are performed on NVIDIA GeForce RTX 2080 Ti 11GB GDDR6.

4.3 Observations

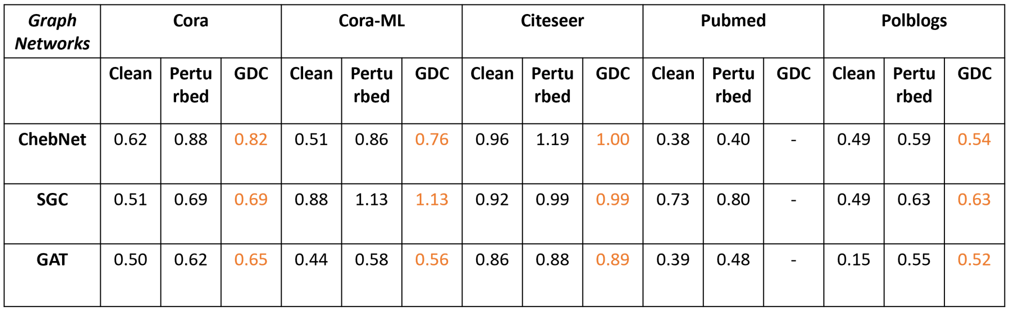
We evaluate the node classification accuracy as shown in Figure 3 and training loss as shown in Figure 4 of three different graph neural network models (ChebNet, SGC, GAT) trained on the perturbed versions of original graph and the transformed graph. In all datasets, the transformed graph achieves better accuracy and loss scores in the presence of adversarial perturbations. The best performance is achieved on ChebNet model while the least amount of improvement is observed on graph attention networks. As attention is a form of feature selection strategy, graph attention networks are inherently robust adversarial attacks. This might be the reason behind the the least improvement observed on this model.

We have also performed additional experiments on the three GNNs at different perturbation ratios (0.05, 0.1, 0.15, 0.2, 0.25) to analyze how would the graph diffusion affect a perturbed graph with increased attack percentage. Figures 5, 6, 7 represent the results of these experiments on only Cora and Citeseer datasets. We could observe from these results that, the accuracy of a diffused graph



Graph Networks	Cora			Cora-ML			Citeseer			Pubmed			Polblogs		
	Clean	Perturbed	GDC	Clean	Perturbed	GDC	Clean	Perturbed	GDC	Clean	Perturbed	GDC	Clean	Perturbed	GDC
ChebNet	0.83	0.73	0.76	0.84	0.71	0.75	0.70	0.66	0.67	0.86	0.85	-	0.76	0.66	0.71
SGC	0.83	0.76	0.77	0.82	0.71	0.71	0.74	0.71	0.72	0.78	0.70	-	0.94	0.74	0.75
GAT	0.84	0.79	0.80	0.85	0.81	0.82	0.73	0.71	0.73	0.85	0.79	-	0.95	0.75	0.76

Figure 3: F1-accuracy for Node Classification task at perturbation ratio = 0.05. Attack model is Metattack (Zügner, et al. (2019)). Pubmed-GDC failed due to GPU memory shortage.

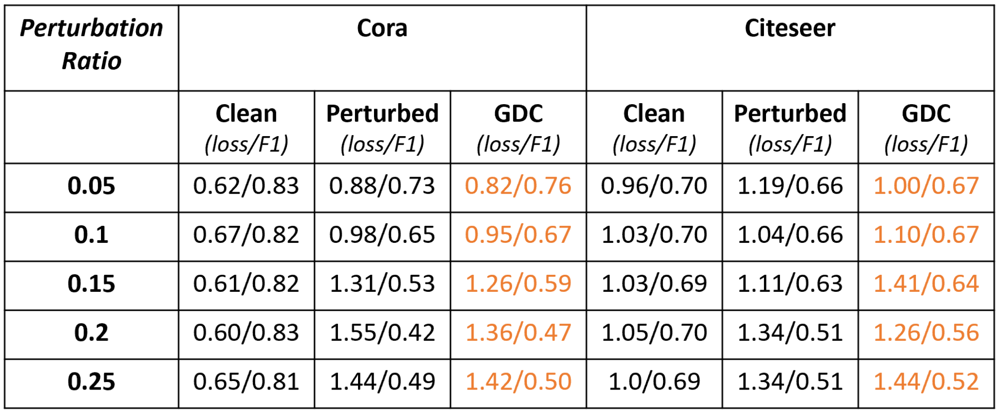


Graph Networks	Cora			Cora-ML			Citeseer			Pubmed			Polblogs		
	Clean	Perturbed	GDC	Clean	Perturbed	GDC	Clean	Perturbed	GDC	Clean	Perturbed	GDC	Clean	Perturbed	GDC
ChebNet	0.62	0.88	0.82	0.51	0.86	0.76	0.96	1.19	1.00	0.38	0.40	-	0.49	0.59	0.54
SGC	0.51	0.69	0.69	0.88	1.13	1.13	0.92	0.99	0.99	0.73	0.80	-	0.49	0.63	0.63
GAT	0.50	0.62	0.65	0.44	0.58	0.56	0.86	0.88	0.89	0.39	0.48	-	0.15	0.55	0.52

Figure 4: Test-set Loss for Node Classification task at perturbation ratio = 0.05. Attack model is Metattack (Zügner, et al. (2019)). Pubmed-GDC failed due to GPU memory shortage.

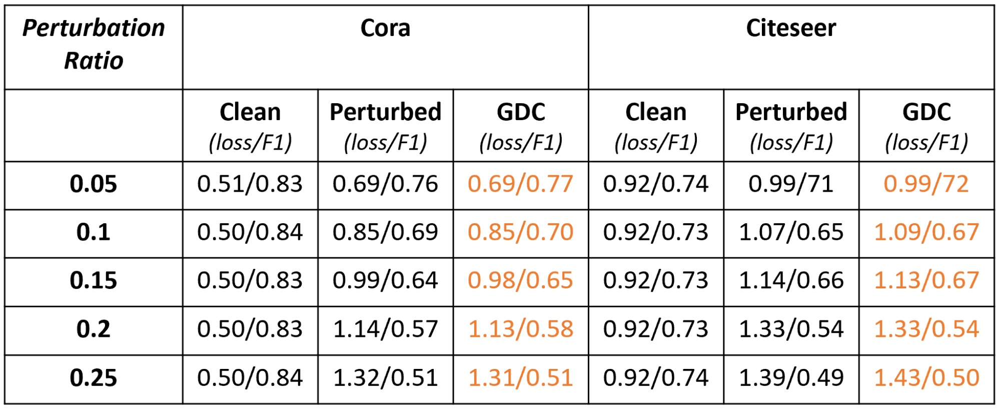
drops along with a non-diffused perturbed graph as the perturbation ratio increases. Even at high perturbation ratios, the diffused graph helps in improving the accuracy of the network.

A natural question is why diffusion in general improves resilience against adversarial perturbations? This might be related to stability of polynomial spectral graph filters under structural changes. As studied by Kenlay, et al. (2020), although adversarial perturbations in the form of node or edge removal leads to a change in the distribution of normalized graph Laplacian, spectral graph filters are reported to be robust to those structural perturbations.



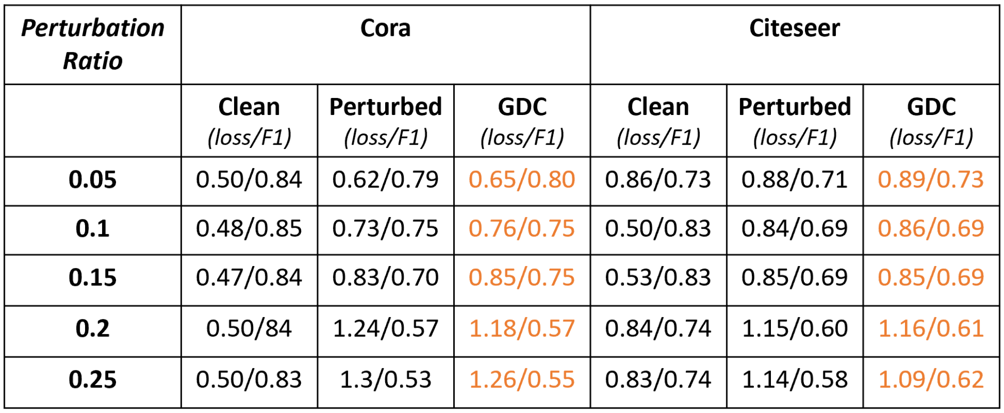
Perturbation Ratio	Cora			Citeseer		
	Clean *(loss/F1)*	**Perturbed** *(loss/F1)*	**GDC** *(loss/F1)*	**Clean** *(loss/F1)*	**Perturbed** *(loss/F1)*	**GDC** *(loss/F1)*
0.05	0.62/0.83	0.88/0.73	0.82/0.76	0.96/0.70	1.19/0.66	1.00/0.67
0.1	0.67/0.82	0.98/0.65	0.95/0.67	1.03/0.70	1.04/0.66	1.10/0.67
0.15	0.61/0.82	1.31/0.53	1.26/0.59	1.03/0.69	1.11/0.63	1.41/0.64
0.2	0.60/0.83	1.55/0.42	1.36/0.47	1.05/0.70	1.34/0.51	1.26/0.56
0.25	0.65/0.81	1.44/0.49	1.42/0.50	1.0/0.69	1.34/0.51	1.44/0.52

Figure 5: Test-set loss and accuracy for Node Classification on ChebNet graph network at different perturbation ratios. Attack model is Metattck (Zügner, et al. (2019)).



Perturbation Ratio	**Cora**			**Citeseer**		
	Clean *(loss/F1)*	**Perturbed** *(loss/F1)*	**GDC** *(loss/F1)*	**Clean** *(loss/F1)*	**Perturbed** *(loss/F1)*	**GDC** *(loss/F1)*
0.05	0.51/0.83	0.69/0.76	0.69/0.77	0.92/0.74	0.99/71	0.99/72
0.1	0.50/0.84	0.85/0.69	0.85/0.70	0.92/0.73	1.07/0.65	1.09/0.67
0.15	0.50/0.83	0.99/0.64	0.98/0.65	0.92/0.73	1.14/0.66	1.13/0.67
0.2	0.50/0.83	1.14/0.57	1.13/0.58	0.92/0.73	1.33/0.54	1.33/0.54
0.25	0.50/0.84	1.32/0.51	1.31/0.51	0.92/0.74	1.39/0.49	1.43/0.50

Figure 6: Test-set loss and accuracy for Node Classification on SGC graph network at different perturbation ratios. Attack model is Metattck (Zügner, et al. (2019)).



Perturbation Ratio	**Cora**			**Citeseer**		
	Clean *(loss/F1)*	**Perturbed** *(loss/F1)*	**GDC** *(loss/F1)*	**Clean** *(loss/F1)*	**Perturbed** *(loss/F1)*	**GDC** *(loss/F1)*
0.05	0.50/0.84	0.62/0.79	0.65/0.80	0.86/0.73	0.88/0.71	0.89/0.73
0.1	0.48/0.85	0.73/0.75	0.76/0.75	0.50/0.83	0.84/0.69	0.86/0.69
0.15	0.47/0.84	0.83/0.70	0.85/0.75	0.53/0.83	0.85/0.69	0.85/0.69
0.2	0.50/84	1.24/0.57	1.18/0.57	0.84/0.74	1.15/0.60	1.16/0.61
0.25	0.50/0.83	1.3/0.53	1.26/0.55	0.83/0.74	1.14/0.58	1.09/0.62

Figure 7: Test-set loss and accuracy for Node Classification on GAT graph network at different perturbation ratios. Attack model is Metattck (Zügner, et al. (2019)).

5 Conclusion and Future Work

Our study proposed graph diffusion convolution as a defense mechanism against adversarial perturbations in graph learning tasks. Under this framework, we interpret malicious node and edge perturbations as the noise injected to data. To prevent the the noisy and less informative signals introduced by imperceptible perturbations, we utilize noise suppression by transforming the graph to its smoother version using graph diffusion convolution.

We evaluate the effectiveness of our proposed methodology on self-meta attacks with experiments on five different datasets. Our results are promising where we achieve up to 8% improvement in accuracy with no decline in accuracy in any of our experiments. A point of concern in our studies was the isotropic nature the diffusion convolution kernel. The next logical step would be testing how different variants of graph diffusion, especially unisotropic ones, behave under the noise suppression framework that we proposed. This can be coupled with different types of adversarial attacks such as targeted attacks along with more complex architectures of GNNs.

6 Contributions

Contributions of individual teammates is listed below.

- Kutlu conceived of the presented idea and research hypotheses.

- Kutlu and Satvik planned experiments to be performed to capture the effect of graph diffusion.
- Satvik implemented the GNNs and GDC to carry out experiments on five datasets. Also, hosted project on GitHub and included a README file with instructions.
- Kutlu and Satvik worked on the final project presentation slides.
- Kutlu and Satvik interpreted results and wrote the project report.

Acknowledgments and Disclosure of Funding

We would like to thank Dr. Ismini Lourentzou, Assistant Professor, Department of Computer Science, Virginia Tech for her continued support throughout this project.

References

[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

[2] Goodfellow, I. J., Shlens, J., Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[3] Xu, H., Ma, Y., Liu, H. C., Deb, D., Liu, H., Tang, J. L., Jain, A. K. (2020). Adversarial attacks and defenses in images, graphs and text: A review. International Journal of Automation and Computing, 17(2), 151-178.

[4] Sun, L., Dou, Y., Yang, C., Wang, J., Yu, P. S., He, L., Li, B. (2018). Adversarial attack and defense on graph data: A survey. arXiv preprint arXiv:1812.10528.

[5] Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., Gu, Q. (2019, January). On the Convergence and Robustness of Adversarial Training. In ICML (Vol. 1, p. 2).

[6] Xu, X., Yu, Y., Li, B., Song, L., Liu, C., Gunter, C. (2018). Characterizing malicious edges targeting on graph neural networks.

[7] Ioannidis, V. N., Berberidis, D., Giannakis, G. B. (2019). Graphsac: Detecting anomalies in large-scale graphs. arXiv preprint arXiv:1910.09589.

[8] Zhang, Y., Khan, S., Coates, M. (2019, May). Comparing and detecting adversarial attacks for graph deep learning. In Proc. Representation Learning on Graphs and Manifolds Workshop, Int. Conf. Learning Representations, New Orleans, LA, USA.

[9] Miller, B. A., Çamurcu, M., Gomez, A. J., Chan, K., Eliassi-Rad, T. (2020). Topological Effects on Attacks Against Vertex Classification. arXiv preprint arXiv:2003.05822.

[10] Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., Papalexakis, E. E. (2020, January). All you need is low (rank) defending against adversarial attacks on graphs. In Proceedings of the 13th International Conference on Web Search and Data Mining (pp. 169-177).

[11] Fox, J., Rajamanickam, S. (2019). How Robust Are Graph Neural Networks to Structural Noise?. arXiv preprint arXiv:1912.10206.

[12] Klicpera, J., Weißenberger, S., Günnemann, S. (2019). Diffusion improves graph learning. arXiv preprint arXiv:1911.05485.

[13] Bojchevski, A., Günnemann, S. (2019, May). Adversarial attacks on node embeddings via graph poisoning. In International Conference on Machine Learning (pp. 695-704). PMLR.

[14] Zügner, D., Günnemann, S. (2019). Adversarial attacks on graph neural networks via meta learning. arXiv preprint arXiv:1902.08412.

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

[16] Kenlay, H., Thanou, D., Dong, X. (2020, May). On the stability of polynomial spectral graph filters. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5350-5354). IEEE.

[17] Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." arXiv preprint arXiv:1606.09375 (2016).

[18] Wu, Felix, et al. "Simplifying graph convolutional networks." International conference on machine learning. PMLR, 2019.

[19] Veličković, Petar, et al. "Graph attention networks." arXiv preprint arXiv:1710.10903 (2017).