Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/69/20241483

Topological consistency with low imperceptibility for graph adversarial attacks

Wenjiang Hu

Faculty of Information Science and Engineering, Ocean University of China, Qingdao 266100, China

huwenjiang2024@163.com

Abstract. Recent research shows that graph neural networks (GNNs) are easy to receive disruptions due to the lack of robustness, the phenomenon that poses a serious security threat. Currently, most efforts to attack GNNs mainly use gradient information to guide the attacks. However, the unreliability of gradient information, and the perceptibility of adversarial examples pose challenges that impede further progress in researching on graph adversarial attacks. From the unreliability of gradient information, we propose a Graph Distance Topological Consistency (GDTC). The scheme introduces graph connectivity, geodesic distance, cosine similarity, and Minkowski distance to construct the similarity matrices of the input space and embedding space of the surrogate model. The difference between the two similarities matrices is constrained during the training process of the surrogate model so that the surrogate model fully learns the topology of the original graph. From adversarial examples perceptibility, we propose attack loss with homogeneity restriction. Experiments show that GDTC can study the topological information of the original graph, enhance the reliability of gradient information, and significantly boost attack performance.

Keywords: Graph neural networks, Homogeneous graph, Adversarial attack.

1. Introduction

A family of deep learning frameworks called Graph Neural Networks (GNNs) is extensively used due to their proficiency with structured data. GNNs are particularly good at capturing the complex structures and relational information of graph data in domains like recommendation systems [1], traffic networks [2], and social networks [3]. However, subtle and imperceptible perturbations on graph data may lead GNNs errors in predicting target nodes [4]. Hence, research on the security and robustness of GNNs is crucial.

GNNs can project nodes to the proximity of centroids corresponding to clusters in the embedding layer, yet struggle to capture the topological structure information of the graph. The backpropagation mechanism based on attack loss leads to perturbations that are influenced by the embedding layer of the surrogate model, making the gradient-based perturbations specific to that model. Specifically, the propagation of gradients between the input layer and the embedding layer is directly linked to the topological structure of the embedding layer. As a result, the perturbations generated by the attacker exhibit a strong specificity towards the embedding layer mapping of the surrogate model, causing the attack to lose its transferability to other models.

DOI: 10.54254/2755-2721/69/20241483

In gradient-based attacks, attackers typically opt to continuously add edges rather than remove them. This strategy is chosen because adding edges swiftly contaminates the transmitted information, thereby impacting the performance of the target model. However, this attack strategy also leads to a decrease in graph homogeneity [5]. Because the victim model is unknown, this study proposes that the surrogate model should preserve consistency between the topological structures of the input layer and the embedding layer in order to improve the attack's transferability and imperceptibility. Additionally, in order to reduce the attack's effect on graph homogeneity, a new attack loss is introduced. In summary, the key contributions of this paper can be summarized as follows:

• We introduce graph geodesic distance to represent the similarity between nodes, enabling the embedding layer of the model to learn topological structure information.

• We introduce a homogeneity-based attack loss, with the aim of sacrificing a small amount of attack performance in exchange for high homogeneity, rendering adversarial examples imperceptible

• We compare our approach with other baselines to verify its efficacy.

2. Related work

There are three main methods to generate adversarial examples: graph node injection [6], graph structure attacks [7], and graph feature attacks [8].

In order to alter the target node labels, an attacker can introduce fake nodes into the graph and link them to certain weak nodes in the original graph. In order to introduce fictitious nodes into the graph data, Sun et al [9]. recommended the use of reinforcement learning using the NIPA approach. To be more precise, NIPA starts by adding a single n nodes to the initial graph. The attacker first chooses an injected node to link to another node in the graph, after which the injected node is given a label. Sequential execution of this process results in a final graph that is statistically comparable to the original graph, but lowers the overall performance of the model.

The attacker performs edge addition as well as edge deletion operations for the original graph nodes under a certain budget. Ma et al. [10] proposed a new framework, ReWatt, for black-box attacks. Using a reinforcement learning framework, ReWatt uses a rewiring operation. To make the perturbation unnoticeable and maintain the important properties of the original graph, one rewiring operation involves three nodes v1, v2 and v3, where ReWatt removes the existing edges between v1 and v2 and connects v1 and v3. ReWatt also restricts v3 to be a two-hop neighbor of v1 to reduce the impact. The number of nodes and edges in the graph remains unchanged by this disturbance.

The attacker only changes the original graph node feature while still maintaining the important properties of the graph. Ma et al. [8] allow the attacker to add a small constant perturbation to a set of nodes S, combining the idea of maximizing influence propagation in social networks [11], thus reducing the overall performance of the GNNs model.

3. Methodology

3.1. Topological Structural Consistency

The connectedness and geodesic distance of nodes in the graph are introduced in order to derive an approximate representation of node similarity in a non-Euclidean space [12]. First, we take into account node connectivity. The distance between unconnected nodes in the graph is regarded as limitless. For connected nodes in the graph, the distance between them is the sum of the number of edges along the shortest path. The shortest path between nodes v_i and v_j with k edges is represented as: $\zeta(i,j) = \{E_{(\gamma_0,\gamma_1)}, E_{(\gamma_1,\gamma_2)}, \dots, E_{(\gamma_{k-1},\gamma_k)}\}$, where E denotes the edge between two nodes, γ_0 corresponds to node v_i, and γ_k corresponds to node v_j. The geodesic distance from v_i to v_j is then expressed as: (v_i, v_j), calculated by the following equation:

DOI: 10.54254/2755-2721/69/20241483

$$\left(v_i, v_j\right) = \begin{cases} \sum_{p=0}^{k-1} d\left(E_{(\gamma_p, \gamma_{p+1})}\right) & \text{if } A_{i,j} > 0 \\ \kappa \max\left(\left[\left(v_i, v_j\right)\right]\right) & \text{if } A_{i,j} = 0 \end{cases} \quad (1)$$

Where $d(E)$ is defined as the cosine similarity between two nodes along the path. $A_{i,j} > 0$ indicates the connection between two nodes in the graph. The formula states that if nodes v_i and v_j are connected in the graph, their distance is the cumulative distance of edges in $\zeta(i,j)$. If the two nodes are not connected, the distance between the two points is multiplied by a large constant κ to indicate the lack of connection. Therefore, a graph's geodesic similarity matrix can be obtained: $S(A, X^*) = \{i, j \mid i, j = 1, \ldots, n\}$.

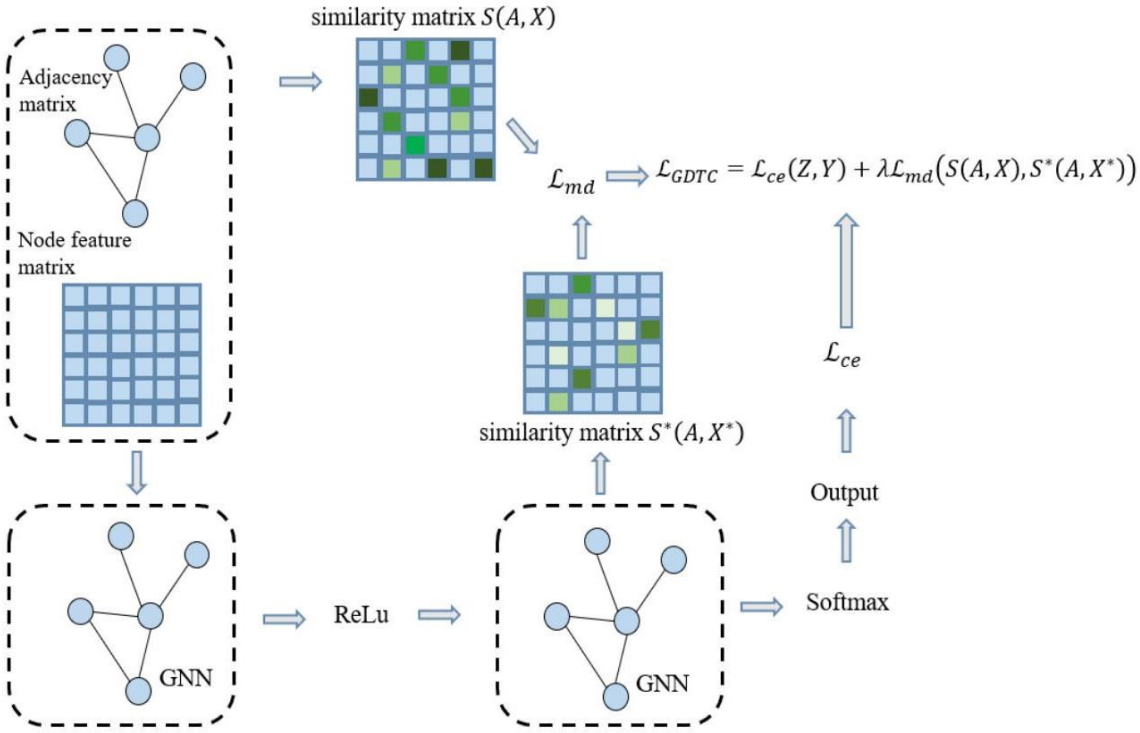

Figure 1. Illustration of the GDTC surrogate model.

By computing the geodesic graph similarity matrix $S(A, X^*)$, we can derive and constrain the topological structure of any layer in the surrogate model. In particular, we use $S^*(A, H^*)$ to represent the embedded layer's similarity matrix. $S^*(A, H^*)$ on the hidden layer is constrained by the geodesic graph similarity matrix $S(A, X)$ on the input layer. The general framework for training the surrogate model using GDTC is shown in Fig.1. The GDTC-based surrogate model's loss function is shown as follows:

$$\mathscr{L}_{GDTC} = \mathscr{L}_{ce}(Z, Y) + \lambda \mathscr{L}_{md}\left(S(A, X), S\left(A, H^{(L)}\right)\right) \quad (2)$$

In this case, Z stands for the probability distribution, Y for a set of labels, and $H^{(L)}$ for the node embeddings, the final hidden layer before the network output. The cross-entropy function that is typically utilized in the training phase of node classification tasks is represented by the first term, $\mathscr{L}_{ce}$. The Minkowski Distance of the node similarity matrices in the input and embedding spaces is represented by the second term, $\mathscr{L}_{md}$. It has the following definition:

DOI: 10.54254/2755-2721/69/20241483

$$\mathscr{L}_{md}\left(S(A,X),S\left(A,H^{(L)}\right)\right)=\left(\sum_{i}^{n}\sum_{j}^{n}\left|S_{i,j}-S_{i,j}^{*}\right|^{p}\right)^{\frac{1}{p}} \tag{3}$$

In our experiments, we utilize $p=2$. The graph similarity matrix $S(A,X)$ and the embedding layer similarity matrix $S^{*}\left(A,H^{(*)}\right)$ make up $\mathscr{L}md'$'s input. Reducing the differences in node topological structures between the two layers is the aim of the second term in GDTC. The surrogate model learns to fit the topological structure of the input layer network $S(A,X)$ to that of the embedding layer $S^{*}\left(A,H^{(*)}\right)$ while $S(A,X)$ stays fixed.

3.2. Incorporating Homogeneity Coefficient

In place of the original adversarial loss, we suggest a homogeneity-constrained adversarial loss to prevent the arbitrary choice of perturbation type to use in each iteration. The following represents the adversarial loss:

$$\begin{gathered}\mathscr{L}_{atk}=\frac{1}{N}\sum_{v_i}^{N}P\left(y_i\mid f_{\theta^{(t)}}(v_i)\right),\quad \mathscr{L}_{hr}=\left(\frac{\|AH\|_0}{\|A\|_0}-\frac{\left\|A^{(t)}H\right\|_0}{\left\|A^{(t)}\right\|_0}\right)^2,\\ \lambda_1=\left(1-\frac{h_g(A)-h_g\left(A^{(t)}\right)}{\varepsilon h_g(A)}\right)^2,\quad \lambda_2=\left(\frac{h_g(A)-h_g\left(A^{(t)}\right)}{\varepsilon h_g(A)}\right)^2,\\ \mathscr{L}_{atk-hc}=\lambda_1\cdot\mathscr{L}_{atk}+\lambda_2\cdot\mathscr{L}_{hc}\end{gathered} \tag{4}$$

The confidence of the substitution model in predicting the label class for node v_i is represented by $P\left(y_i\mid f_{\theta^{(t)}}(v_i)\right)$ in Eq.4. The cross-entropy loss is denoted by $\mathscr{L}_{ce}$, and the loss term that enforces graph homophily is $\mathscr{L}_{hc}$. $H_{ij}=1$ if nodes v_i and v_j have the same label; if not, $H_{ij}=0$. The pseudo-labels Y are used to compute the matrix H. In this case, the total number of edges in the graph is represented by $\|A\|_0$, whereas the number of intra-class edges is indicated by $\|AH\|_0$.

The aim of $\mathscr{L}_{hc}$ is to minimize the disparity in homophily coefficients between the original graph adjacency matrix A and the perturbed graph adjacency matrix $A^{(t)}$ at the t^{th} iteration. λ_1 and λ_2 serve as a pair of weighting parameters. With $h_g(A)\approx h_g\left(A^{(t)}\right)(t)$, as λ_1 approaches 1 and λ_2 approaches 0 , it signifies that the current iteration's attack is free from homophily constraints. Given that λ_1 tends towards 0 and λ_2 tends towards 1, $h_g(A)-h_g\left(A^{(t)}\right)\approx\varepsilon h_g(A)$. This suggests that the attack loss in this iteration is centered on ensuring graph homophily. The attack loss for $0<h_g(A)-h_g\left(A^{(t)}\right)<\varepsilon h_g(A)$ is a trade-off between the two loss terms. In the cooperative interaction between λ_1 and λ_2, the new loss function can prevent graph homophily from decreasing.

4. Experiments

4.1. Experiments Set

We utilized three citation network datasets: Citeseer, Cora, and Cora-ML. The comparative approaches, including Random Attack (RA), DICE, Meta-Self [13], EpoAtk [14], and AtkSE [15], are all aimed at perturbing Graph Neural Networks (GNNs) through edge manipulation. We attacked several common Graph Neural Networks: GCN, SGC, and GAT. For each target model, our approach was compared against other adversarial attack methods.

4.2. Adversarial Examples Performance

Table 1 displays the attack results when the surrogate model uses the victim model's network architecture. Although they have different initializations for the weight parameters, the victim and

DOI: 10.54254/2755-2721/69/20241483

surrogate models are both configured as GCN. In this case, the attacker learns more about the victim model, which enhances the attack's effectiveness. From Table 1, it can be observed that GDTC outperforms the best-performing baseline in all non-targeted poisoning attack experiments. In terms of overall attack performance, GDTC demonstrates the most significant effect, followed by AtkSE, Meta-Self, and EpoAtk. Compared to other methods, our approach shows certain improvements, especially on Citeseers, where at perturbation rates of 1%, 3%, and 5%, our method enhances the performance by 1.4%, 0.5%, and 0.8% respectively. On Cora, GDTC surpasses the runner-up solution by 0.6%, 1.9%, and 0.8%. On Cora-ML, GDTC's performance is elevated by 0.3% to 0.9% compared to the second-best approach.

Table 2 presents the experimental results where the victim model is unknown to the attacker. The surrogate model consists of SGC components, while the victim model is GCN. The results of the experiment where the attacker does not know the victim model are shown in Table 2. SGC components make up the surrogate model, whereas GCN components make up the victim model. The results illustrate how well GDTC, EpoAtk, DICE, Meta-Self, AtkSE, and RA perform on Citeseer, Cora, and Cora-ML at various perturbation rates.

Table 1. The surrogate model and the victim model are both GCN. The table shows the misclassification rates (%) at perturbation rates of 1%, 3%, and 5%

Datasets	Cora			Citeseer			CoraML		
Pert rate	1%	3%	5%	1%	3%	5%	1%	3%	5%
Original		17.8			30.2			15.9	
RA	17.9	18.2	18.7	30.3	30.6	31.1	16.5	16.9	17.2
DICE	18.1	18.5	19.2	30.9	31.3	31.6	16.8	17.5	18.3
Meta-Self	25.2	27.6	30.3	40.8	41.3	43.7	24.6	25.4	28.8
EpoAtk	23.7	24.3	24.8	34.5	34.8	34.7	19.2	21.2	22.9
AtkSE	27.8	28.8	31.7	41.5	43.2	44.4	26.8	28.7	29.2
GDTC	**28.4**	**30.7**	**32.5**	**42.9**	**43.8**	**45.2**	**27.1**	**29.1**	**30.1**

Table 2. The surrogate model is SGC, while the victim model is GCN. The table shows the misclassification rates (%) at perturbation rates of 1%, 3%, and 5%

Datasets	Cora			Citeseer			CoraML		
Pert rate	1%	3%	5%	1%	3%	5%	1%	3%	5%
Original		17.8			30.2			15.9	
RA	18.1	18.4	19.3	30.9	31.2	31.7	16.3	16.6	17.1
DICE	19.2	19.1	19.3	30.8	31.3	31.5	16.9	17.3	18.6
Meta-Self	23.3	25.1	25.8	38.3	39.7	42.3	22.1	23.8	26.3
EpoAtk	21.5	23.6	24.7	31.5	33.2	33.9	16.8	17.9	19.2
AtkSE	24.6	25.9	27.7	39.5	42.1	42.9	23.8	**26.9**	28.4
GDTC	**25.9**	**26.9**	**28.3**	**40.5**	**42.9**	**44.6**	**25.2**	**26.4**	**29.3**

DOI: 10.54254/2755-2721/69/20241483

A comparison between the results in Table 1 and Table 2 reveals that in both experiments, the victim model is GCN. However, when the architecture of the victim model is unknown, all attack performances significantly deteriorate. Particularly on Cora, Meta-Self, EpoAtk, AtkSE, and GDTC decrease by up to 2.7%, 3.0%, 1.5% , and 0.7% , respectively. Among them, GDTC exhibits the least decrease in misclassification rate and generally performs well. This underscores the substantial differences in learned content in the embedding layers across diverse network architectures, which is a key hindrance to attack transferability.

A sensitivity analysis was conducted at different levels of ε constants, as shown in Table 3, where the victim model is SGC, and the homogeneity coefficients of the original graph/perturbed graph are denoted as clean_h and $\mathscr{L}_{atk-hc}$ h, respectively. Note that as ε approaches $+\infty$, $\mathscr{L}_{atk-hc} = \mathscr{L}_{atk}$. From the results, it is observed that as ε increases, h decreases. A lower h corresponds to poorer classification performance. Under homogeneity constraints, $\mathscr{L}_{atk-hc}$ still demonstrates relatively good attack performance. For instance, with $\varepsilon = 1\%$ using the Cora dataset as an example, the homogeneity rate h decreases by 0.023 compared to the $\mathscr{L}_{\text{atk}}$ group, but at the cost of sacrificing 5.1% of attack performance, indicating a significant increase in imperceptibility.

Table 3. The victim model is SGC. The table shows perturbation rates (%) and graph homogeneity coefficients with constraints of ε at 1%, 3%, and 5%

Datasets	Cora			Citeseer			CoraML		
ε	1%	3%	5%	1%	3%	5%	1%	3%	5%
clean		19.2			31.8			17.5	
clean_h		0.746			0.583			0.757	
$\mathscr{L}_{\text{atk}}$		33.5			46.8			31.9	
$\mathscr{L}_{\text{atk}}$ h		0.703			0.542			0.716	
$\mathscr{L}_{atk-hc}$	28.4	28.9	30.2	42.5	43.6	44.9	26.3	27.8	29.2
$\mathscr{L}_{atk-hc}$ h	0.726	0.721	0.719	0.569	0.564	0.560	0.742	0.737	0.724

Table 4. The victim model is GCN. The table shows perturbation rates (%) and graph homogeneity coefficients with constraints of ε at 1%, 3%, and 5%.

Datasets	`Cora			Citeseer			CoraMI		
ε	1%	3%	5%	1%	3%	5%	1%	3%	5%
clean		17.8			30.2			15.9	
clean_h		0.746			0.583			0.757	
$\mathscr{L}_{atk}$		31.8			44.6			30.5	
$\mathscr{L}_{atk}$ h		0.705			0.547			0.719	
$\mathscr{L}_{atk-hc}$	26.7	27.2	28.5	38.9	41.2	43.5	24.2	26.1	27.4
$\mathscr{L}_{atk-hc\_h}$	0.725	0.714	0.711	0.571	0.568	0.559	0.742	0.735	0.728

In terms of attack performance, when $\varepsilon = 1\%$, compared to $\varepsilon = 5\%$, the attack performance of these three datasets decreased by 1.8%, 2.4% , and 2.9% respectively, but the homogeneity coefficients increased by 0.004,0.009, and 0.018 respectively. When the victim models are GCN, as shown in Tables 4, a negative correlation between attack performance and homogeneity is observed, indicating this characteristic applies to most homogeneous graph neural network models.

Therefore, by sacrificing a small portion of attack performance, we can achieve significant imperceptibility gains through $\mathscr{L}_{atk-hc}$. As ε gradually increases, attack performance improves, but the

DOI: 10.54254/2755-2721/69/20241483

imperceptibility of adversarial examples decreases. The experiment validates the effectiveness of our proposed attack loss algorithm in enhancing the imperceptibility of attacks.

5. Conclusion

In this research, we explore the impact of losing topological information on the reliability of gradients. To address this, we introduce a solution based on topological structure consistency, known as Graph Distance Topological Consistency (GDTC). This approach incorporates graph connectivity and cosine similarity to construct geodesic distance. By combining Minkowski distance with the commonly used cross-entropy function in the original classification task, a new surrogate model is formed.Furthermore, we introduce an attack loss with homogeneity constraints to enhance the imperceptibility of adversarial examples.

6. References

[1] Zhong T, Wang T, Wang J, Wu J and Zhou F 2020 IEEE Access 8 95223-95234
[2] Wang X, Ma Y, Wang Y, Jin W, Wang X, Tang J, Jia C and Yu J 2020 Traffic flow prediction via spatial temporal graph neural network Proceedings of the web conference 2020 pp 1082-1092
[3] Wu S, Tang Y, Zhu Y, Wang L, Xie X and Tan T 2019 Session-based recommendation with graph neural networks Proceedings of the AAAI conference on artificial intelligence vol 33 pp346 − 353
[4] Zügner D, Akbarnejad A and Günnemann S 2018 Adversarial attacks on neural networks for graph data Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery &^{2} data mining pp 2847 − 2856
[5] Zhu J, Jin J, Loveland D, Schaub M T and Koutra D 2022 How does heterophily impact the robustness of graph neural networks? theoretical connections and practical implications Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining pp 2637-2647
[6] Zou X, Zheng Q, Dong Y, Guan X, Kharlamov E, Lu J and Tang J 2021 Tdgia: Effective injection attacks on graph neural networks Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery 8 Data Mining pp 2461-2471
[7] Lin X, Zhou C, Wu J, Yang H, Wang H, Cao Y and Wang B 2023 Pattern Recognition 133109042
[8] Ma J, Deng J and Mei Q 2022 Adversarial attack on graph neural networks as an influence maximization problem Proceedings of the fifteenth ACM international conference on web search and data mining pp 675 − 685
[9] Wang X, Cheng M, Eaton J, Hsieh C J and Wu S F 2022 Journal of Computational and Cognitive Engineering 1 165-173
[10] Ma Y, Wang S, Derr T, Wu L and Tang J 2021 Graph adversarial attack via rewiring Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining pp 1161-1169
[11] Kempe D, Kleinberg J and Tardos É 2003 Maximizing the spread of influence through a social network Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining pp 137 − 146
[12] Shamai G and Kimmel R 2017 Geodesic distance descriptors Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp 6410-6418
[13] Wang Z, Hu G and Hu Q 2020 Training noise-robust deep neural networks via meta-learning Proceedings of the IEEE/CVF conference on computer vision and pattern recognition pp 4524-4533
[14] Lin X, Zhou C, Yang H, Wu J, Wang H, Cao Y and Wang B 2020 Exploratory adversarial attacks on graph neural networks 2020 IEEE International Conference on Data Mining (ICDM) (IEEE) pp 1136-1141

DOI: 10.54254/2755-2721/69/20241483

[15] Liu Z, Luo Y, Wu L, Li S, Liu Z and Li S Z 2022 Are gradients on graph structure reliable in gray-box attacks? Proceedings of the 31st ACM International Conference on Information & Knowledge Management pp 1360-1368