Stronger and Transferable Node Injection Attacks

Samyak Jain and Tanima Dutta

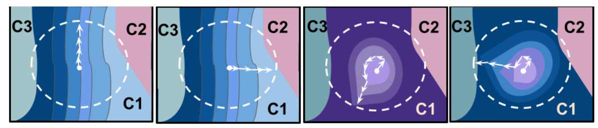
Indian Institute of Technology (BHU) Varanasi
samyakjain.cse18@itbhu.ac.in, tanima.cse@itbhu.ac.in

Abstract

Despite the increasing popularity of graph neural networks (GNNs), the security risks associated with their deployment have not been well explored. Existing works follow the standard adversarial attacks to maximize cross-entropy loss within an L-infinity norm bound. We analyze the robustness of GNNs against node injection attacks (NIAs) in black-box settings by allowing new nodes to be injected and attacked. In this work, we propose to design stronger and transferable NIAs. First, we propose margin aware attack (MAA) that uses a maximum margin loss to generate NIAs. We then propose a novel margin and direction aware attack (MDA) that diversifies the initial directions of MAA attack by minimizing the cosine similarity of the injected nodes with respect to their respective random initialization in addition to the maximization of max-margin loss. This makes the NIAs stronger. We further observe that using L-infinity norm of gradients in the attack step leads to an enhanced diversity amongst the node features, thereby further enhancing the strength of the attack. We incorporate transferability in NIAs by perturbing the surrogate model before generating the attack. An analysis of Eigen Spectrum Density of the hessian of the loss emphasizes that perturbing the weights of the surrogate model improves the transferability. Our experimental results demonstrate that the proposed resilient node injection attack (R-NIA) is significantly stronger and transferable than existing NIAs on graph robustness benchmarks.

Introduction

Graph neural networks (GNNs) have gained success in performing learning and inference on graph-structured data (Kipf and Welling 2017b; Hamilton, Ying, and Leskovec 2017; Zhou et al. 2020; Ye et al. 2022) An increasing amount of attention is being paid on *node injection attacks (NIAs)* (Chen et al. 2022; Zou et al. 2021; Wang and Gong 2019; Sun et al. 2020) that aim to insert new nodes into the graph without making changes in the existing nodes or structure. An attacker can make new accounts in the social networking graph and manipulate the attributes of those accounts in a way such that original nodes get fooled and suggest wrong recommendations to existing users. Similarly, in the case of graphs on financial data where the transactions between the customers and merchants are stored in a graph, fooling the fraud detection system by making fake accounts can lead to catastrophic results. Gradient-based optimization can be done to perturb the node features and identify the appropriate locations to inject new nodes. One of the popular attacks, projected gradient descent (PGD) attack (Madry et al. 2018), proposes to maximize cross-entropy loss to generate the attack. Carlini and Wagner (2017); Gowal et al. (2019) show that max-margin loss can generate stronger attacks on images as compared to PGD (Madry et al. 2018).



(a) PGD (b) MAA (c) CSM (d) MDA

Figure 1: Visualization of loss contours for different models. (a) Cross-Entropy loss maximization (like PGD) can move along loss contour lines of class C3, which is suboptimal. (b) MAA moves orthogonal to loss contours of class C2 due to max-margin loss maximization. Maximizing max-margin loss leads to traversal in the direction of the local smoothness of the loss surface near the point of initialization. (c) Minimizing the cosine similarity (CSM) between attacked and initial features of injected nodes ensures initial exploration in C1. (d) MDA moves orthogonal to loss contour lines of class C1 after exploring local space by using cosine similarity minimization. Thus, maximizing max-margin loss after minimizing cosine similarity leads to a strong attack.

As shown in Figure 1 the white dot ball represents the threat model in which the attack is constrained to remain. Within the defined threat model, the attacker can fool the model to classify class C3. Since PGD only minimizes the probability of the true class and does not take the probability of the true target class (class C3), as shown in Figure 1 (a), on perturbing the node features using PGD, it can end up traversing along the loss contour lines of class C3. Thus, the loss with respect to class C3 remains almost constant. However, maximizing the max-margin loss leads to traversal in the direction orthogonal to the loss contour lines (Figure 1 (b)). Motivated by this, we propose to maximize max-margin loss instead of cross-entropy. We term this attack as margin aware attack (MAA). But as shown in Figure 1 (b), just maximizing max-margin loss leads to traversal in the direction according to the local smoothness of the loss surface near the point of initialization, it might happen that loss in the local direction of class C2 increases more than class C3 near the initialization point. This can lead to the propagation of the attack in the wrong direction. Thus, it is important to consider the local smoothness of initialization to generate stronger attacks. Inspired by this, we propose to minimize the cosine similarity between the attacked and initial features of injected nodes near the initialization (Figure 1 (c)) followed by maximizing max-margin loss for generating stronger attacks. Figure 1 (d) depicts that using cosine-similarity minimization between the attacked and initial features of injected nodes helps in exploring the local space, and thereafter max-margin loss helps generate a stronger attack. We also find that ℓ_2 norm gradient ascent while generating the node injection attack can enhance diversity amongst the attacked features. We also find that perturbing the surrogate model within an ℓ_2 norm ball in the weight space before initiating the attack leads to improved transferability of the attack generated on the surrogate model. This may happened because the attack is no longer specific to the optimal solution of the surrogate model and therefore generalizes better to other models. We summarize our contributions as follows:

- PGD only minimizes the probability of the true class and does not take the probability of the true target classon perturbing the node features and can end up traversing along the loss contour lines Maximizing the max-margin loss overcomes this issue because it leads to traversal in the direction orthogonal to the loss contour lines. We therefore propose margin aware attack (MAA) that uses max-margin loss to modify the features of the injected nodes in a graph.
- We incorporate directionality in MAA by minimizing the cosine similarity with respect to the random initial direction in initial attack iterations to better explore the attack constraint space. We named the model as margin and direction aware attack (MDA) that explores the attribute space and result in perturbations that are diverse and exploratory.
- We demonstrate the use of ℓ_2 norm gradient ascent, instead of the commonly used ℓ_∞ norm, that leads to an improved attack strength, since there is no constraint of variation in the features of the injection nodes.
- We propose to improve the attack transferability by perturbing the weights of the surrogate model within an ℓ_2 norm constraint in the weight space, before generating the attack. We show that the new loss landscape generalizes better in a black box attack setting. We term this attack as margin, direction and transferability aware attack (MDTA).
- We experimentally show that the proposed methods consistently outperform PGD by margins over 14%. We demonstrate the effectiveness of the proposed method on small graph datasets like Cora, Flickr, and Citesee as well as large graphs like Aminer, Amazon2M, and Twitter. We also show that our attack generalizes well to GNN different classes and works well for adversarially trained GNNs.

The rest of the paper is organized as follows. We describe related works and preliminaries in next two sections. Then, we elaborated the proposed attacks. We finally show the experimental results and conclude in last two sections.

Related Works

Graph neural networks (GNNs) have recently gained much attention because of their wide applications. Graph convolutional networks (GCN) (Kipf and Welling 2017a; Wang, Jia, and Gong 2021), graph attention networks (GAT) (Velickovic et al. 2018), and Cluster-GCN (Chiang et al. 2019) are most popular GNN models. FGSM (Szegedy et al. 2013) is the most popular single-step attack, which maximizes the cross-entropy loss. PGD (Madry et al. 2018) is a multistep version of FGSM but it initializes the attack using uniform noise. TDGIA (Zou et al. 2021) is an edge selection strategy to choose the locations for injecting new nodes and generate features on the injected nodes in order to fool the classification of the existing nodes in the graph. G-NIA (Tao et al. 2021) aims to craft a single node which can fool some target nodes. Thus, it attacks only some target nodes of the graph. For a fair comparison, we adapted G-NIA (Tao et al. 2021).

While MetaAttack (Zügner et al. 2020) and NAttack (Zügner, Akbarnejad, and Günnemann 2018) were originally proposed as poisoning attacks, they have been adopted for node injections in Wang et al. (2020). As shown in Wang et al. (2020), the adaptation of MetaAttack and NAttack outperforms FGSM (Szegedy et al. 2013), but it is computationally very expensive. Motivated by this Wang et al. (2020) propose Approximate Fast Gradient Sign Method (AFSM), which performs similarly but it is computationally much cheaper than MetaAttack and NAttack. In this work, we compare the proposed method with MetaAttack, NAtatck and AFGSM as well. Recently, Chen et al. (2022) introduced a regularizer based on cosine similarity, which ensures that the homophility between the features of the added nodes and the original nodes in the graph is maintained. This helps in the generation of imperceptible attacks while being strong. Since it is important to testify that the proposed attack remains imperceptible in nature, we perform a study on the imperceptibility of the proposed method and utilize the closest attribute distance (CAD) (Zou et al. 2021) as the metric. While imperceptibility is important but it is also important to ensure that the attack remains strong. We demonstrate that the proposed attack, while being imperceptible, also outperforms the existing methods on the Graph robustness benchmark (Zheng et al. 2021).

Preliminaries

Threat Model: We consider the same threat model, as considered in Chen et al. (2022). The aim of the attack is to fool a GNN $\mathbf{f}_\Theta$ for graph $G = (A, X)$ by generating another graph $G' = (A', X')$, where $A \in \mathbb{R}^{d\times d}$ denotes the graph adjacency matrix and $X \in \mathbb{R}^{d\times b}$ denotes the node feature matrix. Here, d is the number of nodes in the original graph and each node has a feature vector of dimension b. The number of new nodes injected is given by n. The objective function is:

$$\max_{||G'-G||\leq \mathbf{\Delta}} \boldsymbol{\ell}_{atk}(\mathbf{f}_\Theta(G')), \tag{1}$$

$$\mathbf{\Delta} \rightarrow \{|X_{atk}| \leq \mathcal{P} \in \mathbb{Z};\ |A'-A| \leq \mathcal{R} \in \mathbb{Z};\ ||X'||_\infty \leq \boldsymbol{\epsilon} \in \mathbb{R}\},$$

where $\boldsymbol{\Delta}$ is the constraint on the perturbed graph (aforesaid three constraints) and $\boldsymbol{\ell}_{atk}$ is the attack loss which is maximized in order to fool the GNN. $A' = [A;A_{atk}]$, $X' = [X;X_{atk}]$ and ';' denotes concatenation operation. $\{\mathcal{P},\mathcal{R}\}$ are the upper bound on the number of injected nodes and the number of new edges inserted, respectively. The range of new nodes injected is also bounded. The loss is perturbed on a given input by limiting the upper value of features by $\boldsymbol{\epsilon}$. **Black Box Setting:** In black box setting, any information about the architecture of the system under attack is kept secret. Similar to (Lord, Mueller, and Bertinetto 2022; Chen et al. 2022), we train a surrogate GNN to check the transferability of the attack. We test the attack transferability for same and different architectures of surrogate and black box models.

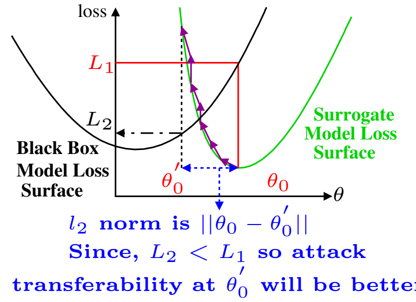
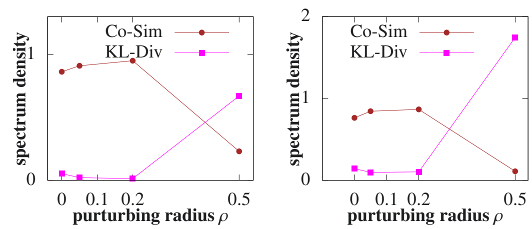


Figure 2: Weight perturbation on a model (surrogate) converged to a local minima escapes local minima and might generalize better to other model's (black box) loss surface.



(a) on Cora (b) on Citeseer

Figure 3: Similarity between the distribution of spectrum density for the perturbed surrogate and original black box models. The proposed MDA attack is used for evaluation.

Rethinking Strength of Node Injection Attacks

Margin-aware Node Injection Attack (MAA)

The maximization of the cross-entropy loss in PGD attack (Madry et al. 2018) leads to the minimization of the originally predicted class confidence only since the cross-entropy loss in PGD does not consider the confidence of other classes. Figure 1 (a) shows that ***decreasing the confidence of one class need not decrease the confidence of the potential attack class*** (class whose boundary is closest to input) in a multi-classification setting, and this results in no change in prediction. On the contrary, it is observed in Figure 1 (b) and Figure 1 (Supplementary) that by maximizing the max-margin loss, the above scenario can be avoided. The max-margin loss $\boldsymbol{\ell}_{mm}$ is given by:

$$\boldsymbol{\ell}_{mm}(\mathbf{f}_\Theta) = -\mathbf{f}_\Theta^y(x) + \max_{i \neq y} \mathbf{f}_\Theta^i(x). \quad (2)$$

Thus maximizing the max-margin loss leads to the maximization of the margin between the true class and the highest confident false class via decreasing the confidence of the true and maximizing the confidence of the false class. Motivated by this, we propose margin aware attack (MAA), which maximizes max-margin loss to generate the attack. We empirically show in Table 1 that MAA attack is stronger than PGD (Madry et al. 2018) by over 1.3% on Citeseer (Giles, Bollacker, and Lawrence 1998) dataset.

Leveraging Directional Similarity in MAA (MDA)

Max-margin loss helps in achieving a larger norm between original and attacked node features. However, Figure 1 (b) depicts that it is biased towards the smoothness of the local space near the initialization point. ***Since local smoothness need not lead to the best trajectory globally, thus simply maximizing max-margin loss can lead to suboptimal attack generation.*** Motivated by this, we propose:

> Inclusion of directional similarity by minimization of cosine similarity between attacked and original features of injected nodes leads to an increase in attack diversity, whereas maximizing max-margin loss ensures a stronger node injection attack.

Cosine similarity can only help in exploring the constraint space rather than generating stronger attacks. Based on the aforesaid analysis, we use cosine similarity along with max-margin loss for initial k iterations of the proposed attack in order to get a good initialization and later shift to the only max-margin loss (Figure 1 (d)). MDA loss $\boldsymbol{\ell}_{mda}$ is given by:

$$\boldsymbol{\ell}_{mda}(\mathbf{f}_\Theta(X \oplus X_{atk})) = \boldsymbol{\ell}_{mm}(\mathbf{f}_\Theta(X \oplus X_{atk})) \\ - \gamma \times \boldsymbol{\ell}_{co-sim}(\mathbf{f}_\Theta(X \oplus X_{init}), \mathbf{f}_\Theta(X \oplus X_{atk})), \quad (3)$$

where X_{init} denotes the randomly initialized feature matrix of the injected nodes. X_{atk} denotes the feature matrix of the injected nodes at some time stamp of the attack generation. $\oplus$ denotes the concatenation operation. The max-margin loss ($\boldsymbol{\ell}_{mm}$) is defined in Eq. (2) and the cosine similarity loss $\boldsymbol{\ell}_{co-sim}$ is defined as follows:

$$\boldsymbol{\ell}_{co-sim}(\mathbf{f}_\Theta(X \oplus X_{init}), \mathbf{f}_\Theta(X \oplus X_{atk})) = \\ \frac{\mathbf{f}_\Theta(X \oplus X_{init})^\top \mathbf{f}_\Theta(X \oplus X_{atk})}{||\mathbf{f}_\Theta(X \oplus X_{init})||\;||\mathbf{f}_\Theta(X \oplus X_{atk})||}. \quad (4)$$

We further present a detailed explanation of the proposed MDA attack using a toy example in Supplementary.

Rethinking ℓ_∞ Norm for Gradient Ascent

Past works Madry et al. (2018); Chen et al. (2022) naively use the sign of the gradients for crafting attacks. The sign of the gradient (in case of ℓ_∞ attack) in the attack generation

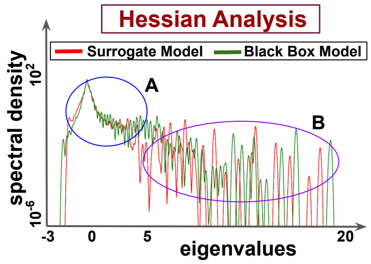


Figure 4: Eigenvalues spectrum of the loss calculated on training set. A: smaller eigenvalues have similar density. B: dominant larger eigenvalues have different densities. This indicates two loss landscapes are very different.

step may lead to a similar increase or decrease (by ϵ) in all the features in a node feature vector. This leads to less diverse features in the injected nodes resulting in a weaker attack. On the other hand, if instead of taking the sign of the gradients, the ℓ_2 norm of the gradients is taken for gradient ascent in attack iteration, there is no constraint of a definite change in all the features in the injection node's feature vector. This leads to enhanced diversity in each feature dimension of the node feature vector and thus leads to a stronger attack. Therefore, we propose to take the $\boldsymbol{\ell}_2$ norm of gradients for crafting attacks in the gradient ascent step instead of taking the sign of the gradients. Here the threat model is the same, and we propose to use a different way to update the gradients during attacks. As shown in Table 1 and Table 2 in Supplementary, this observation leads to improvements upto 7% on $\boldsymbol{\ell}_\infty$ norm.

Enhancing Transferability in MDA (MDTA)

As demonstrated by Lord, Mueller, and Bertinetto (2022), merely generating strong attacks on surrogate models does not guarantee strong attack transfer, thus leading to poor generalization. On the other hand, Foret et al. (2021) showed that flatter minima lead to improved generalization. Motivated by this, we analyze the effect of generating attack on a perturbed surrogate model so that the attack generated is on a loss landscape that is more generalizable across different architectures. Let, the i^{th} node feature vector x_i and corresponding ground truth label y_i, on maximizing the cross-entropy loss to perturb the weights of GNN $\mathbf{f}$, parameterized by Θ, we get, perturbed weights as follows:

$$\widetilde{\Theta} = \underset{\Theta \in \mathcal{M}(\Theta)}{\mathbf{argmax}} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\ell}_{\mathrm{ce}}(\mathbf{f}_\Theta(x_i), y_i). \tag{5}$$

To develop a better understanding, we present a motivational example in Figure 2, where the black curve represents the loss curve for the black box model and the green curve represents the loss curve for the surrogate model. Let θ_0 represent the trained weights of the surrogate model, which have over-fitted to its minima. Now, if a perturbation within the perturbation bound of $\boldsymbol{\ell}_2$ norm radius $||\theta_0 - \theta_0^{'}||$ is generated, then the weights of the models (θ_0) can change to $\theta_0^{'}$ after addition of the perturbation. If the loss of the black box model is computed at points θ_0 (L_1) and $\theta_0^{'}$ (L_2), then it is clear that $L_2 < L_1$. Therefore, perturbing the weights of the surrogate model can help in finding the weights which have better local properties in the black box model. Therefore, generating an attack from the perturbed surrogate model can lead to improved transferability. Therefore, we propose that:

> Perturbing a model Θ within a constraint $\widetilde{\boldsymbol{\epsilon}}$ in $\boldsymbol{\ell}_2$ norm bound leads Θ to come out of a sharp minima and the attack generated on the perturbed model $\widetilde{\Theta}$ can have better transferability.

We show the Eigen Spectrum Density of Hessian of the loss calculated using the black box and surrogate models with the same GCN architecture and trained independently in Figure 4. Here, the spectral density of black box and surrogate models is similar for small eigenvalues and quite different for larger eigenvalues which are responsible for determining the sharpness of the loss landscape (Foret et al. 2021). Therefore, perturbing the model in $\boldsymbol{\ell}_2$ norm ball can help in matching the spectral density of larger eigenvalues, thus improving the transferability. We quantify the similarity between the spectral distributions for the black box and perturbed surrogate models in Figure 3. We use cosine similarity and KL Divergence as the similarity metrics to compare the two distributions. It is observed that perturbing the surrogate model using low to moderate perturbation bound radius leads to improved similarity. This indicates that perturbing the surrogate model can indeed lead to better alignment between the loss landscapes of the surrogate and black box models. To investigate whether this alignment should lead to increased attack transferability, we analyze the robust accuracy (%) using different ℓ_2 norm and perturbation radius (ρ) in Table-3 in Supplementary for the Cora dataset. We observe improved transferability by over 3% as the value of ρ is increased to 0.2. This shows that perturbing the surrogate model can help in improving transferability. Further, since the accuracy of the surrogate model itself drops significantly at larger values of ρ, we observe poor transferability.

Resilient Node Injection Attack (R-NIA)

We propose resilient node injection attack (R-NIA), which is described in Algorithm 1. Given a black box model $\mathbf{g}_\Theta$, we first train a surrogate model $\mathbf{f}_\Theta$. Then, an attack is performed using this model and the black box g_Θ is used for evaluating the attack. After randomly initializing the attacked adjacency matrix and the attacked feature matrix (Line 3) with gaussian distribution, we first maximize the standard cross-entropy loss (Zhang et al. 2019; Wu, Wang, and Xia 2020), in order to come out of the sharp minima of the surrogate model where it might have converged to. For this, we use a $\boldsymbol{\ell}_2$ norm with a bound of $\rho = 0.2$. The details are presented in Line 4. The detailed study on the effect of the value of ρ is discussed in Table 3 in Supplementary. After perturbing the model, we get a new model $\mathbf{f}_{\widetilde{\Theta}}$. In order to identify the right locations where new nodes should be injected (similar to Chen et al. (Chen et al. 2022)), we adopt a gradient-based attack on the adjacency matrix. We calculate the gradients on the adjacency matrix and take the $\boldsymbol{\ell}_\infty$ norm of the gradients

Algorithm 1: Resilient Node Injection Attack (R-NIA)

1: **Input:** $\mathbf{f}_\Theta$, $G = \{(A, X)\}$;
2: **Output:** A_{atk}, perturbed weights $\widetilde{\Theta}$;
3: Randomly initialize the matrix (A_{atk}, X_{atk}); $A_{atk} = 0.1 \cdot \mathcal{N}(0,1)_{n\times n}$, $X_{atk} = X_{init}$ and $G' = (A \oplus A_{atk}, X \oplus X_{atk})$; $X_{init} = 0.1 \cdot \mathcal{N}(0,1)_{nxb}$; *% iid variables sampled from a standard normal. %*
4: $\widetilde{\Theta} = \underset{\Theta \in \mathcal{M}(\Theta)}{\mathbf{argmax}} \frac{1}{d} \sum_{i=1}^{d} \ell_{\text{ce}}(\mathbf{f}_\Theta(x_i), y_i)$; *% ℓ_{ce} denotes the standard cross-entropy loss. %*
5: $A_{atk} = A_{atk} + \mathbf{sign}\left(\nabla_{A_{atk}} \times \boldsymbol{\ell}_{\text{ce}}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk}))\right)$;
6: $A'_{atk} = \mathbf{top}_e(A_{atk})$;
7: $A_{atk} = \mathbf{round}(A'_{atk}, 0, 1)$;
8: $\boldsymbol{\ell}_{mda}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk})) = \ell_{mm}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk})) - \gamma \times \ell_{co-sim}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{init}), \mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk}))$;
$iter = 1$ **to** I
9: **if** $iter >= \frac{2}{3} I$ **then**
$\gamma = 0$;
10: **end if**
11: $X_{atk} = X_{atk} + \boldsymbol{\alpha} \times \frac{(\nabla_{X_{atk}} \boldsymbol{\ell}_{mda}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk})))}{\|(\nabla_{X_{atk}} \boldsymbol{\ell}_{mda}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk})))\|}$;
12: $X_{atk} = \mathbf{clamp}(X_{atk}, MAX, MIN)$; *% MIN and MAX depend on range of original nodes. %*
$G' = (A \oplus A_{atk}, X \oplus X_{atk})$;

to update the adjacency matrix (A_{atk}), as shown in Line 5. In order to follow the constraints on the number of inserted edges, we take the $\mathbf{top_e}$ values from the adjacency matrix and make others as zero, where e denotes the upper bound on the number of new injected edges. This gives a new adjacency matrix A'_{atk} (Line 6). Finally, rounding off is taken on the new adjacency matrix A'_{atk} (Line 7). We use a single-step gradient ascent to generate A'_{atk} as the adjacency matrix is unable to take continuous values. We further show an ablation on varying the number of attack steps to attack the adjacency matrix in Figure 5(a) in Supplementary.

We modify the injected node features so that the nodes belonging to the original graph G get fooled. For this, we utilize a combination of max-margin and cosine similarity loss (Line 8). We perform an iterative gradient ascent using the proposed MDA loss with I number of iterations and step size of $\boldsymbol{\varepsilon}$, where we maximize the max-margin loss while minimizing the cosine similarity loss between the random initialization and the attacked features of the injected nodes. We minimize the cosine similarity only for 2/3 of the total number of iterations I (where $I = 1000$), as shown in Line 10. As discussed previously, using $\boldsymbol{\ell}_2$ bound on gradients helps in enhancing the diversity and therefore leads to stronger attacks. Thus, we propose to take $\boldsymbol{\ell}_2$ norm of the gradients in Line 11. Finally, clamping is performed (Line 12) to ensure that constraints are maintained. At inference time, instead of the surrogate model, the black box model $\mathbf{g}_\Theta$ is used for evaluation on the perturbed graph G'.

Experiments Results

Training Details

We empirically test our models on six datasets, *i.e.*, Cora (Yang, Cohen, and Salakhutdinov 2016), Citeseer (Giles, Bollacker, and Lawrence 1998), Flickr (Young et al. 2014), Aminer (Tang et al. 2008), Amazon2M (McAuley et al. 2015), and Twitter (Boldi and Vigna 2004; Kwak et al. 2010). Table 1 in Supplementary shows the details of the number of nodes and edges in each of the datasets. Cora and Citeseer are citation networks and Flickr is a graph built by extracting features from images. Aminer, Amazon2M, and Twitter are larger-scale graph datasets. CN→ AT (Velickovic et al. 2018) means that the attack is generated on graph convolutional network GCN (Kipf and Welling 2017a) and transferred to graph attention network GAT, where GCN is the surrogate model and GAT is the black box model. Unless specified in all our experiments, the attack is crafted using 1000 iterations and upto 200 new nodes along with upto 500 edges are allowed to be injected in a GNN. For attacking, we use $\rho = 0.2$ and $\gamma = 0.1$ unless explicitly mentioned. The same constraint bound is used across all the datasets. We perform standard empirical risk minimization to train the GNNs. For training the model, we use stochastic gradient descent (SGD) with momentum as the optimizer. Training is done for 5K iterations and the best epoch is chosen based on a hold-out validation set. A learning rate of 0.1 is used for training with a step schedule decay at 2.5K and 3.75K iterations. All experiments are conducted on RTX 2080. A->S means that an attack is generated on adversarially trained GCN(A) and transferred to standard trained GCN(S). {K,M} indicates $\{10^3, 10^6\}$.

Evaluation Results

We present the experimental results of PGD (Madry et al. 2018), MAA (Margin Aware Attack), MTA (Margin and Transferability Aware attack), MDA (Margin and Direction Aware attack), MDTA (Margin, Direction And Transferability Aware attack), and R-NIA (Resilient Node Injection Attack) with $\boldsymbol{\ell}_2$ (R-NIA) and $\boldsymbol{\ell}_\infty$ norm (R-NIA-ℓ_∞) of gradients in Table 1 for Cora and Citeseer datasets and Table 2 in Supplementary for Flickr dataset. Further results on large datasets, like Aminer, Amazon2M, and Twitter, are present in Table 2. We utilize PGD (Madry et al. 2018), PGD with Harmonious Adversarial Objective (HAO) (Chen et al. 2022), TDGIA (Zou et al. 2021), MetaAttack (Wang et al. 2020), G-NIA (Tao et al. 2021) and Approximate FGSM (AFGSM) (Wang et al. 2020) as baselines. We consider two variants of MetaAttack. Either adding all the nodes at once (MetaAttack (one time)) or adding them in a sequential manner (MetaAttack (sequential)). G-NIA (Tao et al. 2021) aims to craft a single node which can fool certain target nodes of the graph. For a fair comparison, we modified the optimization problem of (Tao et al. 2021), to maximize the loss for all the injected nodes in the graph and attack all the original nodes.

★ **Is using $\boldsymbol{\ell}_\infty$ norm of gradients (Madry et al. 2018; Chen et al. 2022) the best choice?** As seen in the bottom half of Table 1 and Table 2 (Supplementary), we find that using $\boldsymbol{\ell}_2$ norm of the gradients instead of $\boldsymbol{\ell}_\infty$ leads to a significant drop in the classification accuracy. Due to enhanced diversity in the features in the case of $\boldsymbol{\ell}_2$ norm, we observe stronger attack with upto 7% higher attack strength.

★ **Is maximizing Cross-Entropy loss a good choice in PGD attack formulation?** We compare the results of PGD

	Cora				Citeseer			
Model	CN→CN	CN→AT	AT→CN	AT→AT	CN→CN	CN→AT	AT→CN	AT→AT
MetaAttack (Wang et al. 2020) (once)	30.74	32.21	33.46	31.01	22.31	24.78	24.14	22.75
MetaAttack (Wang et al. 2020) (seq.)	31.01	32.85	33.21	31.51	23.03	24.91	24.52	23.12
PGD (Madry et al. 2018)	33.21	36.42	37.46	35.68	21.42	23.14	24.02	22.68
G-NIA (Tao et al. 2021)	35.17	37.87	38.21	36.01	23.61	23.56	24.49	23.41
HAO (Chen et al. 2022)	25.21	27.21	26.74	26.31	19.78	19.79	20.65	19.64
MAA	32.71	35.42	36.67	34.21	20.17	21.45	22.03	20.87
MTA	31.87	32.48	34.21	33.98	19.64	20.01	20.99	20.65
PGD (co-sim↓)	52.10	53.89	53.01	52.63	31.44	32.03	32.87	31.26
MDA	21.02	22.89	23.01	21.63	19.44	21.03	21.87	20.87
MDTA	20.41	21.96	22.54	21.01	18.45	18.98	18.86	18.64
R-NIA-ℓ_∞	**19.23**	**20.45**	**21.22**	**19.85**	**17.21**	**17.46**	**17.58**	**17.32**
PGD-ℓ_2	26.12	29.36	30.06	28.54	18.01	20.98	22.03	20.01
MAT-ℓ_2 (Madry et al. 2018)	26.22	29.04	30.01	28.65	17.45	18.65	19.09	17.64
MTA-ℓ_2	26.06	26.36	29.07	28.45	17.32	17.94	18.21	17.74
PGD-ℓ_2(co-sim↓)	41.30	41.69	42.45	42.16	26.45	28.01	27.46	26.23
MDTA-ℓ_2	19.01	20.47	19.86	19.36	17.01	18.68	19.65	19.02
MDTA-ℓ_2	18.26	19.69	20.01	18.57	16.02	16.74	16.61	16.31
R-NIA	**17.03**	**18.43**	**18.69**	**17.13**	**14.86**	**15.02**	**15.36**	**15.03**

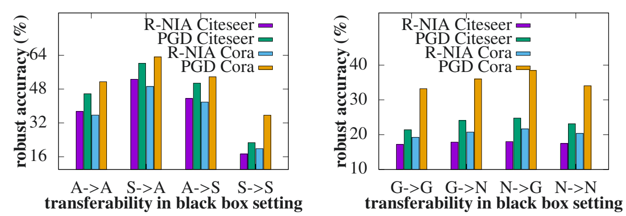
Table 1: Robust Accuracy (%) on Cora and Citeseer datasets for black box setting. CN denotes GCN and AT denotes GAT.

and MAA in Table 1 and Table 2 (Supplementary). We observe that the performance of MAA is better than PGD by over 1.5% on Cora and Flickr and 2% on the Citeseer.

★ **Can maximizing max-margin loss alone lead to sub-optimal search in output constraint space?** Based on Table 1 and Table 2 (Supplementary), it is clear that MAA leads to stronger attacks than PGD. However, minimizing cosine similarity in addition to MAA improves attack strength by upto 12% on Cora (MDA, MDA-ℓ_2 in comparison to PGD). This shows the importance of incorporating directional similarity in the attack objective. The attack optimization is biased on local smoothness of loss landscape. Minimizing cosine similarity helps in better optimization of the attack.

Based on the results, it is important to analyze the effect of minimizing cosine similarity without maximizing max-margin loss in MDA. It is clear from the comparison between PGD (Co-Sim) and PGD that simply minimizing the cosine similarity does not lead to strong attacks. This is because, unlike max-margin loss maximization, minimizing cosine similarity alone does not lead to a particular direction where a class can be changed. But it helps in finding better initialization that can generate strong attacks. Motivated by this, we use cosine similarity for the initial 750 attack iterations out of the total 1000 iterations to find a better starting point.

★ **Can we enhance the transferability of the attacks generated on a surrogate model?** Based on Table 1 and Table 2 (Supplementary), we observe that it is indeed possible to improve the transferability of the attacks. The comparision of MDA with MDTA clearly shows that the transferability of the attacks for different architectures (CN→AT and AT→CN) has improved by upto 3%. Finally, while we observe that HAO (Chen et al. 2022) is the stronger baseline in Table 1, the proposed R-NIA shows improvements upto 8.18% over it on the Cora and around 5% on the Citeseer datasets.



(a) (b)

Figure 5: Impact on transferability between (a) adversarial (A) and standard (S) trained models; (b) non-local GCN (N) and GCN (G) models.

Impact of Graph Size, #Inject Node & #Inject Edge

As shown in Table- 2, we observe that on adding a larger number of nodes and edges the proposed R-NIA attack improves the performance over PGD by over 8.67% on Flickr and by over 10% on an even larger dataset like Aminer. We also observe gains on up to 10 times further larger datasets (larger graph size - Table 1 in Supplementary) like Amazon2M and Twitter datasets. On Amazon2M, we outperform PGD by upto 6.09% and by 7% on Twitter dataset. We observe that as larger number of edges and nodes are allowed to be injected, the strength of PGD attack increases relatively slowly as compared to R-NIA attack.

Transferability on Robust Models

Figure 5 (a) checks the attack transferability where the surrogate model is either an adversarially trained model (A) or a standard trained model (S). We train the GCN and GAT models using ten steps PGD adversarial training to get the robust models (denoted by A). We observe transferability gains between robust models (A→A) over 15% on the Cora dataset. Further, we observe that R-NIA shows improved

	Flickr				Aminer			
Model	CN→CN	CN→AT	AT→CN	AT→AT	CN→CN	CN→AT	AT→CN	AT→AT
PGD (200,500) (Madry et al. 2018)	41.03	43.01	44.85	44.03	64.23	63.78	63.41	62.78
R-NIA (200,500)	**38.41**	**39.98**	**41.06**	**41.49**	**61.42**	**63.47**	**62.94**	**62.86**
PGD (1K,2.5K) (Madry et al. 2018)	37.12	38.97	39.21	38.45	51.27	52.79	52.45	50.78
R-NIA (1K,2.5K)	**29.56**	**30.01**	**30.54**	**29.78**	**40.03**	**41.87**	**41.65**	**40.47**
PGD (5K,12.5K) (Madry et al. 2018)	24.87	25.14	25.46	24.65	41.78	42.23	43.14	40.97
R-NIA(5K,12.5K)	**17.45**	**18.02**	**18.23**	**17.56**	**29.78**	**30.75**	**30.04**	**28.94**
	Amazon2M				Twitter			
PGD (500, 12500) (Madry et al. 2018)	52.12	54.78	54.32	53.01	58.78	60.01	59.12	58.65
R-NIA (500, 12500)	**46.03**	**47.08**	**47.84**	**46.23**	**51.79**	**52.07**	**52.47**	**51.06**

Table 2: Robust Accuracy (%) on injecting different number of new nodes $\mathbf{N}_{inj}$ and edges $\mathbf{E}_{inj}$ on Flickr and Aminer datasets, i.e., $\{\mathbf{N}_{inj},\mathbf{E}_{inj}\}$ pair. K and M indicate 10^3 and 10^6. We take ℓ_2 norm of gradients for the PGD (Madry et al. 2018) baseline as well. On larger graphs, more gains over the PGD baseline are observed on injecting larger number of nodes and edges.

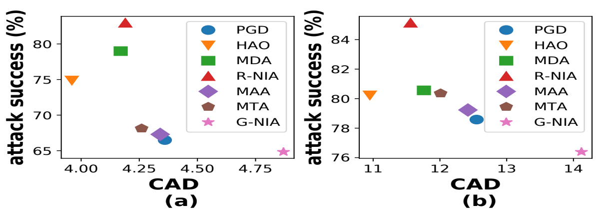


Figure 6: Results on attack success vs CAD, where attack success= $100-$ robust accuracy on (a) Cora and (b) Citeseer.

transferability as compared to PGD when the attack is generated using S/A and evaluated on A/S, respectively. Figure 5 (b) depicts the results of transferability experiments on nonlocal GCN (N). We observe that R-NIA performs better than PGD by up to 13.7% on the CORA dataset for N→N. These results demonstrate that the R-NIA attack is stronger than PGD not only on standard models but also on robust models and generalizes to different classes of GNNs as well.

Ablation Studies

We provide a detailed study of the impact of ℓ_2 norm constraint (ρ) for weight perturbation in MTA on the Cora dataset in Table 3 (Supplementary). It is evident that there is a certain range of values of ρ, which leads to the strongest transferability. $\rho < 0.2$ does not make any significant change where both surrogate and black box models have the same architecture but helps in improving the transferability rate. A larger value of ρ degrades the original model, thus degrading the transferability. We study the effect of changing the constraints on the number of nodes injected in the threat model, as shown in Figures 2 and 3 (Supplementary), where we observe that MDA and R-NIA consistently generate a stronger attack on using a different number of injected nodes. As shown in Figure 5 (a) in Supplementary, increasing the number of steps to generate an attack on A_{atk} does not lead to significant changes in robust accuracy, therefore we propose to use a single-step attack to generate A_{atk}. As shown in Figure 5 (b) in Supplementary using a γ value close to 0.1 leads to the strongest attack. The plot shows that γ is an important hyperparameter for MDAT. As shown in Figure 4 (a) in Supplementary, while PGD leads to a stronger attack for less number of attack steps, it saturates very early. MDA shows improved performance with the increase in the number of attack steps. Figure 4 (b) in Supplementary shows that MDA outperforms PGD on increasing the number of injected edges.

Imperceptibility Study

It is important to ensure that after node injection attack, the graph still remains imperceptible (Chen et al. 2022; Tao et al. 2022). For this, we use the closest attribute distance (CAD) as the metric (Tao et al. 2022; Zou et al. 2021). For each injected node, CAD calculates the nearest node feature with the smallest ℓ_2 norm feature distance with the injected node and averages it over all the injected nodes. The results are shown in Figure 6, where we observe that the proposed methods have a lower value of CAD when compared to PGD. Further, R-NIA leads to stronger attacks with a small increase in CAD when compared to HAO. This shows that the proposed methods ensure that the graph is imperceptible after the node injection attacks. As shown by Tao et al. (2022), there is a strong correlation between CAD and two other imperceptibility metrics: Smoothness (Chen et al. 2022) and Graph FD (Tao et al. 2022). Therefore, CAD is a reliable metric.

Conclusion

We highlight the need to rethink node injection attacks on GNNs.To enhance the attack transferability, we propose to perform a weight perturbation in ℓ_2 norm before crafting the attack. We show that the use of ℓ_∞ norm restricts the diversity of attack features and propose to use ℓ_2 norm instead. We demonstrate improved performance over graph robustness benchmark (Zheng et al. 2021) models. We show that our method is generalizable to larger graphs as well. It improves attack strength in the case of adversarially trained models.

Acknowledgements

We thank SERB MTR/2021/604 and SIR/2022/000536.

References

Boldi, P.; and Vigna, S. 2004. The Webgraph Framework I: Compression Techniques. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, 595–602. New York, NY, USA: Association for Computing Machinery. ISBN 158113844X.

Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *Proc. of IEEE Symposium on Security and Privacy*, 39–57.

Chen, Y.; Yang, H.; Zhang, Y.; KAILI, M.; Liu, T.; Han, B.; and Cheng, J. 2022. Understanding and Improving Graph Injection Attack by Promoting Unnoticeability. In *Proc. of Int. Conf. on Learning Representations*, 1–42.

Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proc. of the ACM*, 257–266.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *Proc. of Int. Conf. on Learning Representations*, 1–20.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: an automatic citation indexing system. In *Proc. of ACM Int. Conf. on Digital Libraries*, 89–98.

Gowal, S.; Uesato, J.; Qin, C.; Huang, P.-S.; Mann, T. A.; and Kohli, P. 2019. An Alternative Surrogate Loss for PGD-based Adversarial Testing. *ArXiv*, abs/1910.09338: 1–15.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Kipf, T.; and Welling, M. 2017a. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv*, abs/1609.02907: 1–14.

Kipf, T. N.; and Welling, M. 2017b. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 591–600. New York, NY, USA: Association for Computing Machinery. ISBN 9781605587998.

Lord, N. A.; Mueller, R.; and Bertinetto, L. 2022. Attacking deep networks with surrogate-based adversarial black-box methods is easy. In *Proc. of Int. Conf. on Learning Representations*, 1–17.

Madry, A.; Makelov, A.; Schmidt, L.; Dimitris, T.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *Proc. of Int. Conf. on Learning Representations*, 1–23.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52.

Sun, Y.; Wang, S.; Tang, X.; Hsieh, T.-Y.; and Honavar, V. 2020. Adversarial Attacks on Graph Neural Networks via Node Injections: A Hierarchical Reinforcement Learning Approach. In *Proc. of The Web Conference*, 673–683.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2013. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 990–998.

Tao, S.; Cao, Q.; Shen, H.; Huang, J.; Wu, Y.; and Cheng, X. 2021. Single Node Injection Attack against Graph Neural Networks. In *30th ACM International Conference on Information and Knowledge Management*, 1794–1803.

Tao, S.; Cao, Q.; Shen, H.; Wu, Y.; Hou, L.; and Cheng, X. 2022. Adversarial Camouflage for Node Injection Attack on Graphs.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio', P.; and Bengio, Y. 2018. Graph Attention Networks. *ArXiv*, abs/1710.10903: 1–12.

Wang, B.; and Gong, N. Z. 2019. Attacking Graph-based Classification via Manipulating the Graph Structure. In *Proc. of ACM SIGSAC Conf. on Comp. & Comm. Security*, 1–18.

Wang, B.; Jia, J.; and Gong, N. Z. 2021. Semi-supervised node classification on graphs: Markov random fields vs. graph neural networks. In *Proc. of AAAI*, volume 35, 10093–10101.

Wang, J.; Luo, M.; Suya, F.; Li, J.; Yang, Z.; and Zheng, Q. 2020. Scalable Attack on Graph Data by Injecting Vicious Nodes. In *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 1–20.

Wu, D.; Wang, Y.; and Xia, S. 2020. Revisiting Loss Landscape for Adversarial Robustness. *ArXiv*, abs/2004.05884: 1–20.

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. *ArXiv*, abs/1603.08861: 1–9.

Ye, Z.; Kumar, Y. J.; Sing, G. O.; Song, F.; and Wang, J. 2022. A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs. *IEEE Access*, 10: 75729–75741.

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. of the Association for Computational Linguistics*, 2: 67–78.

Zhang, H. R.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. *ArXiv*, abs/1901.08573: 1–11.

Zheng, Q.; Zou, X.; Dong, Y.; Cen, Y.; Yin, D.; Xu, J.; Yang, Y.; and Tang, J. 2021. Graph Robustness Benchmark: Benchmarking the Adversarial Robustness of Graph Machine Learning. In *Proc. of NIPS Datasets and Benchmarks Track*, 1–8.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1: 57–81.

Zou, X.; Zheng, Q.; Dong, Y.; Guan, X.; Kharlamov, E.; Lu, J.; and Tang, J. 2021. TDGIA: Effective Injection Attacks on Graph Neural Networks. *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, 1–11.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2847–2856.

Zügner, D.; Borchert, O.; Akbarnejad, A.; and Günnemann, S. 2020. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(5): 1–31.