Understanding and Improving Adversarial Robustness of Deep Neural Networks

Thesis submitted in fulfilment of the requirements
for

Integrated Dual Degree (B.Tech. + M.Tech.)
in
Computer Science and Engineering

Submitted by
Samyak Jain

Under the guidance of
Dr. Tanima Dutta

Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi
Varanasi, Uttar Pradesh, India – 221005

Semester X - May, 2023

Roll Number
18074015

Declaration

I certify that

1. The work contained in this thesis is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted for any project or the award of any other degree/diploma.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date: September 28, 2024

Samyak Jain (18074015 - IDD)
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi
Varanasi, INDIA 221005.

Certificate

This is to certify that the work contained in this thesis entitled **“Understanding and Improving Adversarial Robustness of Deep Neural Networks”** *being submitted by* **Samyak Jain (18074015)** *carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bonafide work of my supervision.* It has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree

It is further certified that the student has fulfilled all the requirements of the Comprehensive Examination, Candidacy and SOTA for the award of **Integrated Dual Degree (B.Tech.+ M.Tech.)**

Dr. Tanima Dutta
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Place: IIT (BHU) Varanasi
Date: September 28, 2024

Acknowledgments

It is a great pleasure for me to express respect and deep sense of gratitude to my supervisor Dr. Tanima Dutta, Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, for her wisdom, vision, expertise, guidance, enthusiastic involvement and persistent encouragement during the planning and development of this work.

I also gratefully acknowledge her painstaking efforts in thoroughly going through and improving the manuscripts without which this work could not have been completed. I am also highly obliged to Prof. Pramod Kumar Jain, Director, Indian Institute of Technology (BHU) Varanasi, and Prof. Sanjay Kumar Singh, Head of Department, CSE for providing all the facilities, help and encouragement for carrying out this masters thesis work. I also thank Dr. Ravi Shankar Singh, as a convener of this course and the other faculty members for their timely help and cooperation extended throughout the course of investigation. I am also obliged to my parents for their moral support, love, encouragement and blessings to complete this task.

Finally, I am indebted and grateful to the Almighty for helping me in this endeavor.

Place: IIT (BHU) Varanasi
Date: September 28, 2024

Samyak Jain

Abstract

The incredible success of Deep Neural Networks (DNNs) has made them very popular. However, Deep Networks are also known to be susceptible to imperceptible perturbations crafted on their inputs (adversarial attack), which can lead them to misclassify images into unrelated classes with exceptionally high confidence. The increasing popularity and adoption of Deep Neural Networks in a variety of domains make it necessary to understand and mitigate their vulnerability towards adversarial attacks. In order to achieve improved robustness, efforts need to be spent not only on developing an improved training methodology but also improving the attack strength to testify the robust models reliably. This thesis focuses on understanding and improving the attack strength on Graph Neural Networks and Vision Transformers. In the first part of the thesis, we demonstrate the flaws in existing node injection attacks on graph neural networks and propose stronger node injection attacks. In the second part of the thesis, we demonstrate the fundamental reasons for gradient masking in VITs and propose attacks which achieve improved attack strength by overcoming the masking effect. Further, we incorporate the proposed attack into existing adversarial training methods to achieve improved robustness.

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 $\boldsymbol{\ell}_\infty$ norm bound. We analyze the robustness of GNNs against node injection attacks (NIAs) in a black-box setting, where new nodes are allowed to be injected and attacked. In this work, we first propose *Margin Aware Attack* (MAA), which uses a maximum margin-based loss formulation to generate an attack. We further propose a novel *Margin and Direction Aware attack* (MDA) that diversifies the initial directions of the 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. To further improve the transferability of MDA, we propose to perturb the surrogate model before generating the attack. Analysis of Eigen Spectrum Density of the loss hessian emphasizes that perturbing the weights of the surrogate model indeed improves the transferability. We further rethink the

gradient update step while generating the attack. We show that using the $\boldsymbol{\ell}_2$ norm of gradients in the attack step leads to enhanced diversity amongst the node features, thereby enhancing the strength of the attack. Our experimental results demonstrate that the proposed *Resilient Node Injection Attack* (R-NIA) is significantly stronger than existing attacks on GNNs and outperforms all existing attack methods on the Graph Robustness Benchmark.

Recent literature has demonstrated that vision transformers (VITs) exhibit superior performance compared to convolutional neural networks (CNNs). The majority of recent research on adversarial robustness, however, has predominantly focused on CNNs. In this work, we bridge this gap by analyzing the effectiveness of existing attacks on VITs. We demonstrate that due to the softmax computations in every attention block in VITs, they are inherently vulnerable to floating point underflow errors. This can lead to a gradient masking effect resulting in suboptimal attack strength of well-known attacks, like PGD, Carlini and Wagner (CW), and GAMA attack. Motivated by this, we propose *Adaptive Attention Scaling* (AAS) attack that can automatically find the optimal scaling factors of pre-softmax outputs using gradient-based optimization. We show that the proposed simple strategy can be incorporated with any existing adversarial attack as well as adversarial training method to get improved performance. On VIT-B16, we demonstrate an improved attack strength of upto 2.2% on CIFAR10 and upto 2.9% on CIFAR100 by incorporating the proposed AAS attack with state-of-the-art single attack methods like GAMA attack. Further, we utilise the proposed AAS attack every few epochs in existing adversarial training methods, which we term as *Adaptive Attention Scaling Adversarial Training* (AAS-AT). On incorporating AAS-AT with existing methods, we outperform them on VITs by over 1.3-3.5% on CIFAR10.

In this thesis, we make progress in the field of adversarial robustness and propose simple ways to overcome the issues with existing works claiming to either develop stronger node injection attacks on GNNs or adversarial attacks on VITs. We demonstrate that our methods are generic and can be incorporated with any existing approaches to achieve improved performance. We hope that this work will open new avenues on improving the robustness of VITs and GNNs even further.

Contents

List of Figures

Chapter 1

Introduction

In recent years, deep neural networks (DNNs) have gained tremendous popularity due to their remarkable success. Different variants of deep neural networks have been proposed. For instance in computer vision for image classification/ generation Convolutional Neural Networks (CNNs) and Vision Transformers are popular. For natural language understanding and generation Recurrent Neural Networks (RNNs) and Transformers are commonly used. Similarly for temporal data and data in form of a graph, Graph Neural Networks are generally used.

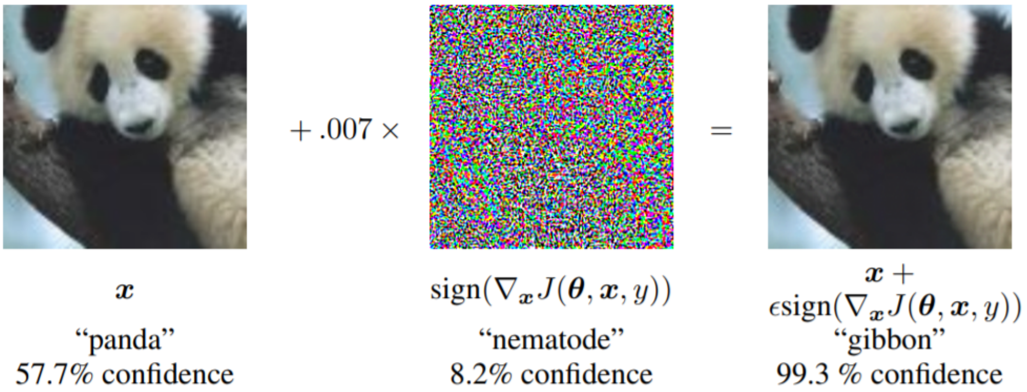

+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1.1: **An adversarial example** crafted for GoogLeNet trained on imagenet data (Source: Goodfellow et al. [29])

1.1 Adversarial Robustness

1.1.1 Adversarial Attacks

Despite the wide popularity and use of DNNs, it has however been revealed that they are susceptible to imperceptible perturbations known as Adversarial Attacks [30, 53]. These attacks can alter the predictions of DNN models, which can lead to catastrophic outcomes. As shown in Figure-1.1, while the original image x and the pertubed image are perceptually similar, the model's classification changes from the true class panda to a false class gibbon. Further, the confidence of the model in classifying the perturbed image as gibbon is very high, which makes it almost impossible to identify if the image is attacked. This unsettling discovery has sparked a surge in research endeavors aimed at enhancing the robustness of DNNs against powerful attacks.

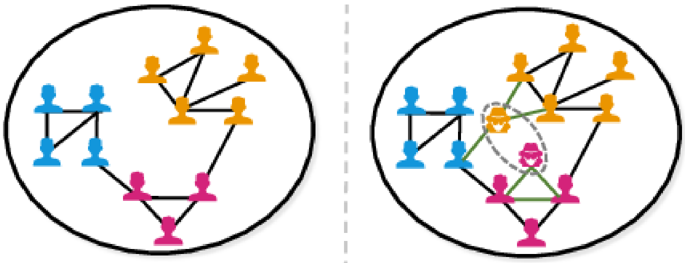


Figure 1.2: **Node Injection Attack (NIA)** is performed on the clean graph (left) by adding new nodes and edges by the attacker (right) which can fool the classification of the nodes in clean graph. (Source: Sun et al. [50])

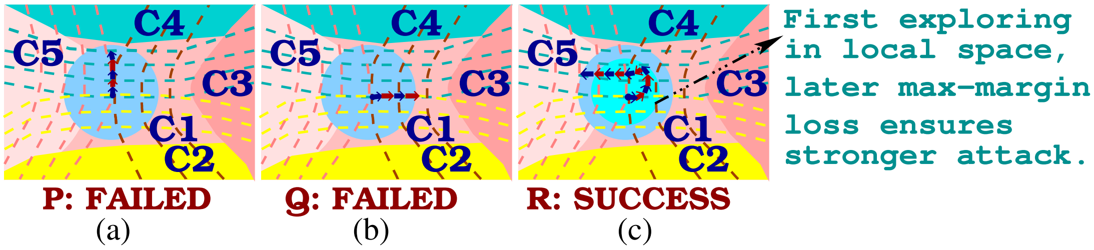
The common way to generate the adversarial attack is to maximize the loss by iteratively backpropogating and perturbing the image as shown in middle column of Figure-1.1. PGD [41] which maximizes the cross-entropy is one of the most popular attack. There have been many attacks, like GAMA [49], Carlini and Wagner (in short CW) [11], AutoAttack [18], which have been shown to surpass PGD.

1.1.2 Node Injection Attacks

It is not just the vision models that are found to be susceptible to the adversarial attacks, graph neural networks (GNNs) have also been shown to be vulnerable to them. GNNs have been shown to be vulnerable to *graph modification attacks (GMAs)* [14, 21, 40, 69, 80] and *node injection attacks (NIAs)* [15, 51, 61, 79] attacks. GMAs focus on deleting nodes in the existing graph or modifying the edges in the graph. Since in real-world scenarios, it might be expensive to modify the graph structure, an increasing amount of attention is being paid on node injection attacks (NIAs) [15, 51, 61, 79] that aim to insert new nodes into the graph without making changes in the existing nodes or structure. We demonstrate the node injection attack on a graph neural network in Figure-1.2. The attacker aims to inject new nodes (right) and perturb the clean graph (left). The aim is to fool the classification of the original nodes in the clean graph.

1.2 Applications

The vulnerability of the deep models towards specially crafted perturbations known as adversarial attacks limits their applicability on safety-critical applications like healthcare and autonomous driving. In these places, it is important to ensure that if the decision is being made using a deep neural network, then it should be correct. For instance, an autonomous self-driving system can be fooled by perturbing the


C4
C5
C3
C1
C2
P: FAILED
Q: FAILED
R: SUCCESS
First exploring in local space, later max-margin loss ensures stronger attack.

(a) (b) (c)

Figure 1.3: **Visualization of loss contours of different classes.** (a) **P**: Cross-Entropy loss maximization (PGD) can move along loss contour lines of class C5, which is suboptimal. (b) **Q**: MAA moves orthogonal to loss contour lines of class C3 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) **R**: MDA moves orthogonal to loss contour lines of class C4 after exploring local space. Minimizing the cosine similarity between attacked and initial features of injected nodes ensures the initial exploration while maximizing max-margin loss leads to a strong attack.

traffic sign on the roadside. This can lead to catastrophic consequences. Therefore ensuring that the models being deployed in these safety-critical scenarios are indeed robust and reliable is very important.

As shown in Figure-1.2, the threat from NIAs can be easily realized in real-world scenarios. Social networking platforms have millions of users, and their data, along with the relationships between them, is typically maintained as a graph. Here edges represent connections between the users, and nodes represent the attributes or features of each user. An attacker can make new accounts in the 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. By enhancing the robustness of the deep networks, we can make them more reliable to use in safety-critical applications as well.

1.3 Motivation

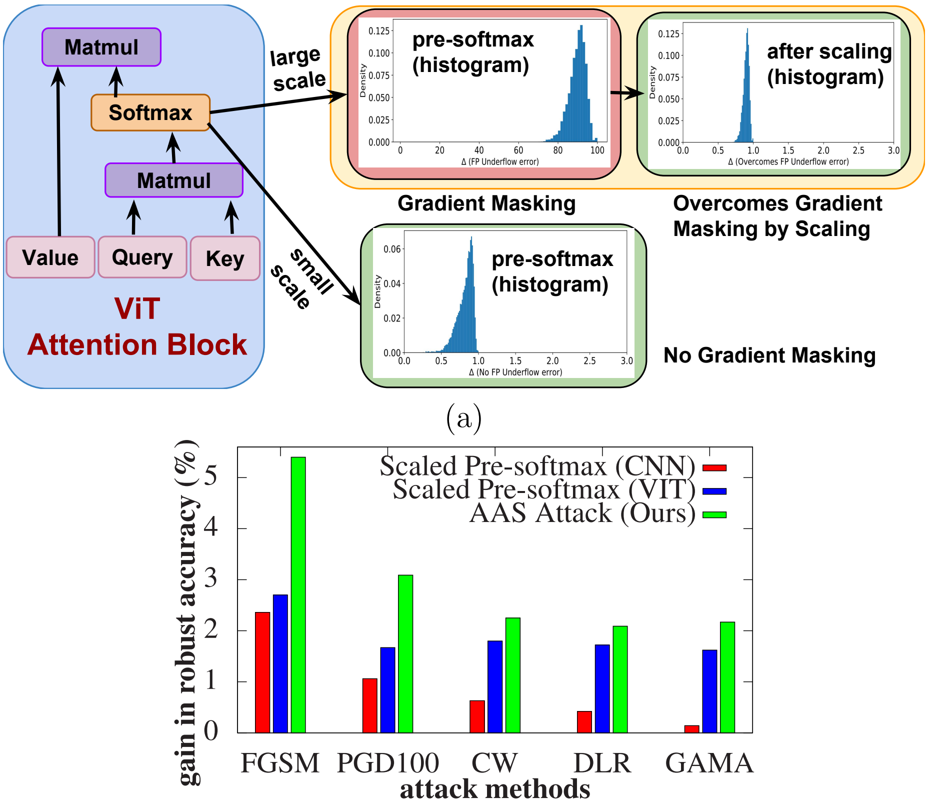
1.3.1 Rethiking Node Injection Attacks

NIAs involve two design strategies: identification of appropriate node injection locations and modification of the injected node features to cause misclassification of other nodes. 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 [41], proposes to maximize cross-entropy loss to generate the attack. Carlini and Wagner [12], Gowal et al. [32] show that max-margin loss can generate stronger attacks on images as compared to PGD [41]. As shown in Figure 1.3 the blue 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 C5. Since PGD only minimizes the probability of the true class and does not take the probability of the true target class (class C5 in this case), as shown in Figure 1.3 (a), on perturbing the node features using PGD, it can end up traversing along the loss contour lines of class C5. Thus, the loss with respect to class C5 remains almost constant. Maximizing the max-margin loss overcomes this issue. As shown in Figure 1.3 (b), maximizing the max-margin loss leads to traversal in the direction orthogonal to the loss contour lines. Motivated by this, we propose to maximize max-margin loss instead of cross-entropy. We term this attack as margin aware attack (MAA). However, merely maximizing max-margin loss, as shown in Figure 1.3 (b), 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-C3 increases more as compared to class-C5 near the initialization point. This can lead to the propagation of the attack in the wrong direction. Thus, there is a need for proper exploration of the local constraint space near the 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, followed by maximizing max-margin loss for generating stronger attacks. As shown in Figure 1.3 (c), the use of cosine-similarity minimization between the attacked and initial features of injected nodes helps in exploring the local space, and thereafter max-margin loss helps in generating a stronger attack. Further, 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. We hypothesize that this happens because the attack is no longer specific to the optimal solution of the surrogate model and therefore generalizes better to other models. We also find that ℓ_2 norm gradient ascent while generating the node injection attack can lead to enhanced diversity amongst the attacked features.

1.3.2 Improving adversarial robustness of VITs

Despite the proposal of various defense mechanisms, many of them fall victim to a phenomenon known as gradient masking, creating a false sense of security [9, 34, 48, 68, 75]. Some defenses in this category incorporate randomized or non-differentiable components into the model to obstruct the calculation of accurate gradients, thereby evading the generation of strong attacks. However, studies by Athalye et al. [3], Carlini et al. [13], Tramer et al. [58] have demonstrated that these defenses can be bypassed by adaptive attacks specifically tailored to the targeted model.

Yu and Xu [72] surprisingly demonstrated on CNNs that even PGD [41] can perform similarly if logits in the output space are scaled properly. The authors



(a)

(b)

Figure 1.4: **Floating point underflow errors in attention blocks of a VIT lead to gradient masking. (a)** Δ is the difference between the largest and the second-largest pre-softmax output. As highlighted in red, if the scale of the pre-softmax outputs is high, floating point underflow error occurs. It will not occur if the scale is low (as highlighted in green). Motivated by this, we propose to downscale the pre-softmax outputs. **(b)** Gains in PGD-100 robust accuracy on using AAS-attack (green) on adversarially trained (PGD-AT [41] with AWP [67]) VIT-B16 model [25] are higher as compared to manually finding the scaling factors (blue). In case of CNNs (red), the logits of adversarially trained (PGD-AT [41]) WideResNet-28-10 are downscaled.

characterized that taking *softmax* leads to **floating point underflow errors**. Here, the difference between the first and second largest logit values is so high that on taking exponential in softmax, the output can't be saved in CPU's memory, thereby leading to floating point underflow error. Since attacks like GAMA [49] and CW [11] don't utilize softmax in output space; they are automatically resilient to floating point underflow errors. This seems to be a probable reason behind their success in the case of CNNs. Further, Hitaj et al. [35] has also shown that training with adversarial training methods, like GAIRAT [75] leads to an enhanced magnitude of the logits in the output space, resulting in the failure of the attack. But, scaling the logits down [10, 19, 35] leads to strong attacks, which results in significantly drop of robustness of GAIRAT [75].

Transformers have been shown to establish new benchmarks in many tasks

[8, 24, 46, 59]. While VITs [25, 57] also have shown improved performance over CNNs, still there has been a debate on whether VITs are more robust than CNNs. Past literature [6, 42, 47] have shown that VITs demonstrate lower attack transferability in the black box setting. Based on this observation, they concluded that VITs are inherently more robust than CNNs. This hypothesis has however been challenged by the recent work [5], where the authors demonstrated for the first time that on using strong attacks, like AutoAttack [18] and activation functions, like GELU in CNNs, transformers are no more robust than CNNs. Further, Naseer et al. [45] showed that adversarial transferability can be significantly improved for VITs if the attack is generated by training a classifier on top of the VIT blocks followed by backpropagating through each of these classifiers. Similarly, Wei et al. [65] has shown that randomly dropping the input patches helps in improving the attack transferability. Though the authors in [45, 65] have shown that it is possible to generate stronger attacks on VITs, however, these methods are indeed very complex. There is a lack of fundamental cause of why it is difficult to generate strong attacks on VITs.

In the second work (chapter), we rethink the understanding of the adversarial robustness of VITs and discuss more fundamental causes of gradient masking in VITs that leads to poor attack strength on using standard attacks, like PGD [41], GAMA [49], and CW [11] attacks, which have demonstrated good attack strength on CNNs. Inspired by Yu and Xu [72], we hypothesize that the reason for gradient masking is VITs is the floating point underflow error which occurs due to softmax calculation in every attention block in VITs. As shown in Figure 1.4 (a), as highlighted in red, we observe that in the case of VITs, a larger scale of pre-softmax outputs can result in floating point underflow. This leads to a false estimate of the gradient, resulting in a weaker attack. But, we can overcome the floating point errors by scaling down the pre-softmax outputs by the right scaling factor. As shown in Figure 1.4 (b), on scaling down the pre-softmax outputs in every attention block the proposed Adaptive Attention Scaling (AAS) attack leads to a boost upto 3% over standard PGD [41] attack on CIFAR10 dataset. In the case of CNNs (shown in red) as demonstrated by [72], scaling down the logits help in improving the attack strength by overcoming floating point errors. We hypothesize that the effect of gradient masking is much more intense in the case of VITs because of softmax functions present in attention blocks. This is evident from the improved gains on simply scaling the pre-softmax outputs manually (shown in blue) over the same method applied to logits in the output space of CNNs (shown in red). We show strong empirical evidence for our hypothesis and point out a more fundamental reason for the poor performance of VITs on adversarial attacks.

To achieve robustness against strong adversarial attacks, the most popular approach is adversarial training (AT). While it is easy to train CNNs [41, 67, 73] using adversarial training, VITs seem to pose multiple challenges [22, 44]. Recently, Mo et al. [44] demonstrated that training a VIT from scratch doesn't converge to a good solution. Therefore, a pre-trained initialization is necessary for training VITs using AT. The authors also showed that to stabilize the adversarial training of VITs, gradients need to be clipped. These clipping or pretraining is not required in the case of CNNs. Further, using complex augmentations like Cutmix and Mixup gives improved results. Similarly, Debenedetti et al. [22] proposes a training recipe to improve VITs robustness. The authors propose to use a ten epoch linear ϵ warmup along with high weight decay to get improved performance. While these tricks, like gradient clipping, warmup, and high weight decay, improve the robustness of VITs, it is not well understood why these tricks are needed. Though it is very easy to train CNNs using adversarial training, it seems difficult to train VITs.

We hypothesize that the floating point underflow error not only leads to weaker attack generation during inference but also during training. This results in suboptimal adversarial robustness on performing adversarial training on VITs. Motivated by this, we propose a new adversarial training method, Adaptive Attention Scaling Adversarial Training (AAS-AT), where we ensure that the scale of the logits doesn't exceed too much. We demonstrate that this simple check by using the proposed AAS attack at regular intervals of training helps in stabilizing the training of VITs and results in improved robustness. We demonstrate that the proposed training method AAS-AT can be combined with different existing adversarial training methods leading to improved performance.

1.4 Contributions

The contributions of the thesis are discussed below:

- We propose margin aware attack (MAA), which uses max-margin loss to modify the features of the injected nodes in a graph.
- We propose margin and direction aware attack (MDA) where cosine similarity with respect to the random initial direction is minimized in initial attack iterations to explore the attack constraint space and thereafter, the max-margin loss is maximized to generate stronger attack

  • We propose to perturb the weights of the surrogate model within an ℓ_2 norm constraint in the weight space. Perturbing the surrogate model overcomes the local minima, and thus, the new loss landscape generalizes better in a black box attack setting. This improves the transferability of the attack. We term this attack as margin, direction and transferability aware attack (MDTA).

- Motivated by the observation that using ℓ_2 norm gradient ascent in the attack step enhances the diversity within the attacked features, we reconsider the past attack methods, which use ℓ_∞ norm of gradients to perform gradient ascent.

- We experimentally show that the proposed methods consistently outperform PGD [41] by margins over 14%. We demonstrate the effectiveness of the proposed method on small graph datasets like Cora [70], Flickr [71], and Citeseer [28] as well as large graphs like Aminer [54], Amazon2M [43], and Twitter [7, 37]. We also show that the proposed attack generalizes well to different classes of GNNs and works well for adversarially trained GNNs as well.

- We demonstrate that the floating point underflow error is caused due to softmax operations in attention blocks. It leads to weak attack generation in the case of VITs. We show this to be the fundamental cause behind weaker white box attacks on VITs.

- We propose a novel attack, named Adaptive Attention Scaling (AAS), that automatically finds the optimal scaling values for pre-softmax outputs in attention blocks, thus mitigating floating point underflow error. We maximize the LPIPS distance in feature space to get perceptually aligned gradients and find optimal scaling factors.

- We propose a robust training model, known as Adaptive Attention Scaling Adversarial Training (AAS-AT), that combines the proposed AAS attack to make the VITs more robust.

- We show the proposed AAS attack and AAS-AT model can be combined with any existing adversarial attack and adversarial training method respectively. We demonstrate improved results as compared to existing methods on CIFAR10, CIFAR100, and Imagenet-100 datasets.

1.5 Organization of the Thesis

The thesis is divided into five chapters: Introduction, Related Works, Rethinking Node Injection Attacks on GNNs, Understanding and Improving Adversarial Robustness of Vision Transformers and Conclusion and Future Directions. We first discuss the existing works related to adversarial attacks on CNNs and VITs. We explain a few popular adversarial attack algorithms in detail. Further, we discuss existing works on node injection attacks and also discuss adversarial training, which is the most popular method to achieve robustness to adversarial attacks. Then we present our work on improving the strength of node injection attacks on GNNs, where we demonstrate that our proposed method Resilient Node Injection Attack (R-NIA), is better than all existing node injection attacks. Next, we present our work on improving the robustness of Vision Transformers, where we first present the proposed Adaptive Attention Scaling (AAS) attack and later propose Adaptive Attention Scaling Adversarial Training (AAS-AT) by incorporating the AAS attack into existing adversarial training methods. Finally, we discuss the conclusion and future directions of this thesis.

Chapter 2

Related Works

2.1 Adversarial Attacks.

Andriushchenko et al. [2], Carlini and Wagner [11], Croce and Hein [17, 18], Madry et al. [41], Sriramanan et al. [49] have focused on building stronger attacks. FGSM [52] is the most popular single-step attack, which maximizes the cross-entropy loss. PGD [41] maximizes the same cross-entropy loss to generate an attack, but as opposed to FGSM, PGD is a multistep attack which uses uniform random noise for attack initialization. Carlini and Wagner [11] showed that maximizing max-margin loss instead of the standard cross-entropy leads to stronger attacks. Croce and Hein [18] proposed AutoAttack, which is an ensemble of four attacks including three white box (i.e., Adaptive PGD with cross-entropy loss, Adaptive PGD with difference of logits ratio loss, Fast adaptive boundary attack [17]) and one black box (square attack [2]). AutoAttack is stronger than existing attacks. However, AutoAttack is computationally expensive because it is an ensemble of four attacks. Therefore, stronger single attacks like GAMA attack [49] have also been proposed, which are weaker than AutoAttack, but give a close estimate of the robustness. It uses a ℓ_2 norm regularizer between the outputs of clean and perturbed images along with a max-margin objective in the first few iterations of the attack. Later, only max-margin loss is maximized and the regularizer is shown to help in improved optimization of the attack. While in the case of VITs it has been observed that attacks don't transfer between them [6, 42, 47]. But [45] showed that training a classifier on each of the attention blocks and further generating an attack by backpropagating through each of them leads to enhanced attack transfer. Further, [44] showed that randomly dropping out some attention blocks and randomly selecting input patches while back-propagating to generate the attack leads to enhanced attack transferability. Though [5, 44, 45] demonstrated that VITs are no more robust than CNNs, they don't point out to the fundamental cause of poor transferability on generating attacks from the original model itself. In the second work (chapter), we show that because of the scale of the pre-softmax outputs, floating point underflow errors can occur which can lead to the gradient masking effect, thus resulting in weaker attack generation. The details of individual attacks is given by:

- Fast Gradient Sign Method (FGSM) [30]: FGSM is a single-step attack which maximizes cross-entropy loss (defined as L_{CE}). The attack objective of FGSM is given below:

$$argmax_{x'} L_{CE}(f_\theta(x'), y) \qquad s.t. ||x' - x||_\infty < \epsilon \tag{2.1}$$

- Projected Gradient Descent (PGD) [41]: PGD is a multistep version of FGSM where the perturbation is initialized using random noise sampled from a uniform distribution.

- Carlini and Wagner (in short CW) [11]: CW attack maximizes max-margin loss (defined as L_{MM} instead of the standard cross-entropy loss used in PGD attack. The attack formulation of the CW attack considered in this work is shown below:

$$argmax_{x'} L_{MM}(f_\theta(x'), y) \qquad s.t. ||x' - x||_\infty < \epsilon \tag{2.2}$$

- Guided Margin Aware Attack (GAMA) [49]: GAMA attack proposed to aid the initial optimization path by maximizing the ℓ_2 norm between the output logits of the adversarial and the clean images along with maximizing the standard max-margin loss. The objective function of GAMA attack is shown below:

$$argmax_{x'} L_{MM}(f_\theta(x'), y) + \lambda ||f_\theta(x^{'}) - f_\theta(x)|| \qquad s.t. ||x' - x||_\infty < \epsilon \tag{2.3}$$

  Over the training iterations, the value of λ is decayed.

- AutoAttack (AA) [18]: AutoAttack is an ensemble of four attacks including three white box (Adaptive PGD with cross-entropy loss, Adaptive PGD with difference of logits ratio loss, Fast adaptive boundary attack [17]) and one black box (square attack [2]). The details of these attacks are described below:

  - Adaptive PGD: APGD is the same as the standard PGD attack but as opposed to PGD it adjusts the step size of the attack automatically. An untargeted APGD is used in the AutoAttack framework.

  - Adaptive PGD with difference of logits ratio loss: This attack uses the DLR loss instead of the standard cross-entropy loss. Further, the target attack is used in the AutoAttack framework. Therefore this attack is expensive because its frequency depends on the number of classes in the dataset.

– Fast Adaptive Boundary Attack (FAB): FAB attack aims to find the minimum perturbation required to change the true class predicted by the model. In the AutoAttack framework, a targeted FAB attack is used and it is the most expensive attack amongst all others in the AutoAttack framework.

– Square Attack: Square attack is the only black box attack present in AUtoAttack. It is a search-based attack, where randomly coloured squared and rectangles are added to the input image and then they are retained if there is an increase in loss value. Since the square attack is a gradient-free attack, it helps to circumvent and identify the models suffering from gradient masking issues.

2.2 Node Injection Attacks

Graph neural networks (GNNs) have recently gained much attention because of their wide applications. Graph convolutional networks (GCN) [36, 62], graph attention networks (GAT) [60], and Cluster-GCN [16] are most popular GNN models. PGD [41] has shown to be very strong for node injection attacks (NIA). TDGIA [79] 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 [55] 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 [55] to attack all the original nodes in the graph.

While MetaAttack [82] and NAttack [81] were originally proposed as poisoning attacks, they have been adopted for node injections in Wang et al. [63]. As shown in Wang et al. [63], the adaptation of MetaAttack and NAttack outperforms FGSM [52], but it is computationally very expensive. Motivated by this Wang et al. [63] propose Approximate Fast Gradient Sign Method (AFSM), which performs similarly but it is computationally much cheaper than MetaAttack and NAttack. In the first work (chapter), we compare the proposed method with MetaAttack, NAtatck and AFGSM as well. Recently, Chen et al. [15] 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) [79] 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 [77], which is a popular benchmark to track the progress of node injection attacks.

2.3 Robustness against Node Injection Attacks

Adversarial training [41] is known to be the most reliable defence strategy against adversarial attacks on images. Zhu et al. [78] focuses on adding stochasticity into the model as a way to evade the success of the NIAs. Further, an attention mechanism is used to identify the nodes which are vulnerable to attacks and can be pruned off, thus making the model robust. However, Athalye et al. [4] demonstrated that robust methods incorporating stochasticity could be broken using adaptive attacks. To achieve robustness, adversarial training seems to be the most reliable option explored in the literature [41, 74].

2.4 Gradient Masking.

Athalye et al. [3], Carlini et al. [13], Tramer et al. [58] demonstrated that many defenses claiming to achieve enhanced robustness can actually be broken down by using adaptive attacks. These defenses which can give a false sense of security on some gradient-based white box attacks are said to have gradient masking. Due to gradient masking, the attacker ends up calculating a false estimate of actual gradients, resulting in a false sense of security. Alike Logit Scaling Attack [35], Yu and Xu [72] demonstrated that larger scale of logits leads to floating point underflow error. In the second work (chapter), we hypothesize that the gradient masking effect due to floating point underflow errors is more intense in VITs.

2.5 Adversarial Training (AT).

PGD-AT [41] showed that maximizing the cross-entropy loss helps to generate a multi-step attack and minimising the same for training the model helps in achieving robustness. MART [64] uses a different minimization loss for the misclassified and correctly classified examples. Trades [73] maximizes the Kullback-Leibler (KL) loss between the outputs of clean and adversarial images while minimizing the same with the cross-entropy loss on clean samples. Trades [76] demonstrated the existence of the fundamental tradeoff between clean and adversarial accuracy. Adversarial Weight Perturbations (AWP) [67] showed that perturbing the weights within a fixed ℓ_2 norm perturbation bound leads to convergence to a flatter minima. This helps to enhance robustness. While CNNs don't suffer from instability issues on performing adversarial training with strong enough attacks. The case of VITs is quite different.

The details of some of the popular adversarial training (AT) methods are given below:

- PGD-AT [41]: PGD-AT performs the standard ten-step PGD attack to generate the adversarial images and later minimized the cross-entropy loss on the generated adversarial images to train the model. The objective function of PGD-AT is shown below:

$$argmax_{x'} L_{CE}(f_\theta(x'), y) \qquad s.t. ||x' - x||_\infty < \epsilon \tag{2.4}$$

$$min\ L_{CE}(f_\theta(x'), y) \tag{2.5}$$

- Trades [73]: Trades maximizes the KL divergence (L_{KL}) loss between the adversarial and the clean image to generate the perturbations and later minimizes the combination cross-entropy loss on clean image and KL divergence loss between the clean and the generated adversarial image. The objective function of PGD-AT is shown below:

$$argmax_{x'} L_{KL}(f_\theta(x'), f_\theta(x)) \qquad s.t. ||x' - x||_\infty < \epsilon \tag{2.6}$$

$$min\ L_{CE}(f_\theta(x), y) + \lambda L_{KL}(f_\theta(x'), f_\theta(x)) \tag{2.7}$$

- Trades-AWP[67]: Trades-AWP proposes to first generate the attack by maximizing the KL divergence between the clean and the perturbed image and then perturb the weights of the model within an ℓ_2 norm perturbation bound (ρ). Later the Trades adversarial training is performed on the perturbed model. The objective function of AWP is shown below:

$$\theta^{'} = \theta + \delta,\ \delta = argmax_{\theta'}(L_{CE}(f_{\theta'}(x), y) + \lambda L_{KL}(f_{\theta'}(x'), f_{\theta'}(x))) - \theta, \\ s.t. ||\theta' - \theta|| < \rho \tag{2.8}$$

$$argmax_{x'} L_{KL}(f_{\theta'}(x'), f_{\theta'}(x)) \qquad s.t. ||x' - x||_\infty < \epsilon \tag{2.9}$$

$$min\ L_{CE}(f_{\theta'}(x), y) + \lambda L_{KL}(f_{\theta'}(x'), f_{\theta'}(x)) \tag{2.10}$$

$$\theta = \theta' - \delta \tag{2.11}$$

In the case of VITs, Mo et al. [44] showed the importance of using pre-trained initializations for training them adversarially. The authors demonstrated the importance of using gradient clipping for stabilizing the adversarial training of VITs. They also claim to randomly remove the gradient flow through some multi-head attention modules and randomly mask the input perturbation during forward propagation.

Though these modifications are not properly justified, they help in enhancing the stability and adversarial robustness on using any existing adversarial training approach. Debenedetti et al. [22] showed that using a larger value of weight decay and a few initial epochs of epsilon warmup can help in improved adversarial robustness. Debenedetti et al. [22] demonstrated that these tricks helps in enhancing the robustness of VITs significantly. On CIFAR100, the authors achieve significant improvement leading to a second entry on the robustness leaderboard [20]. This is the first successful demonstration that VITs can indeed achieve good adversarial robustness. In the second work (chapter), we demonstrate that the proposed Adaptive Attention Scaling Adversarial Training (AAS-AT) can be incorporated with any existing AT methods to achieve improved robustness.

Chapter 3

Rethinking Node Injection Attacks on GNNs

3.1 Introduction

In this chapter, we will discuss some of our findings to improve the strength of the node injection attacks on graph neural networks. We rethink the strength of the existing node injection attack methods. One of the most popular attacks, projected gradient descent (PGD) attack [41], proposes to maximize cross-entropy loss to generate the attack. It is commonly used to craft the features for node injection attacks (NIA). We first present the motivation for our work. As shown in Figure 1.3 the blue 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 C5. Since PGD only minimizes the probability of the true class and does not take the probability of the true target class (class C5 in this case), as shown in Figure 1.3 (a), on perturbing the node features using PGD, it can end up traversing along the loss contour lines of class C5. Thus, the loss with respect to class C5 remains almost constant. Maximizing the max-margin loss overcomes this issue. As shown in Figure 1.3 (b), maximizing the max-margin loss leads to traversal in the direction orthogonal to the loss contour lines. Motivated by this, we propose to maximize max-margin loss instead of cross-entropy. We term this attack as margin aware attack (MAA). However, merely maximizing max-margin loss, as shown in Figure 1.3 (b), 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-C3 increases more as compared to class-C5 near the initialization point. This can lead to the propagation of the attack in the wrong direction. Thus, there is a need for proper exploration of the local constraint space near the 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, followed by maximizing max-margin loss for generating stronger attacks. As shown in Figure 1.3 (c), the use of cosine-similarity minimization between the attacked and initial features of injected nodes helps in exploring the local space, and thereafter max-margin loss helps in generating a stronger attack. Further, we also find that perturbing the surrogate model within an $\boldsymbol{\ell}_2$ norm ball in the weight space before initiating the attack leads to improved transferability of the attack generated on the surrogate model. We hypothesize that this happens because the attack is no longer specific to the optimal solution of the surrogate model and therefore generalizes better to other models. We also find that $\boldsymbol{\ell}_2$ norm gradient ascent while generating the node injection attack can lead to enhanced diversity amongst the attacked features.

In this chapter, firstly, we define the threat model and setup in Section-3.2, then in Section-3.3 we discuss four of our findings which can improve the strength as well as transferability of PGD [41] attack. Motivated by these findings, we propose Resilient Node Injection Attack (R-NIA) in Section-3.4. Finally, we share the experimental findings and ablation results in Section-3.5, where we demonstrate gains over 14% over the PGD [41] baseline on using the proposed R-NIA attack on Cora datasets. We also show that the proposed R-NIA is generalizable to even large-scale graph datasets and robust GNNs as well.

3.2 Preliminaries

Threat Model: We consider the same threat model, as considered in Chen et al. [15]. 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 defined as follows:

$$\max_{||G'-G||\leq \boldsymbol{\Delta}} \boldsymbol{\ell}_{atk}(\mathbf{f}_\Theta(G')),\ \boldsymbol{\Delta} \rightarrow \text{i. } |X_{atk}| \leq \mathcal{P} \in \mathbb{Z};\ \text{ii. } |A'-A| \leq \mathcal{R} \in \mathbb{Z};\ \text{iii. } ||X'||_\infty \leq \boldsymbol{\epsilon} \in \mathbb{R} \tag{3.1}$$

where Δ 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. $A' = [A\boldsymbol{;}A_{atk}]$, $X' = [X\boldsymbol{;}X_{atk}]$ 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 [15, 39], we train a surrogate GNN to check the transferability of the attack. We test the attack transferability for the same and different architectures of the surrogate and black box models.

3.3 Rethinking Strength of Node Injection Attacks

3.3.1 Exploring Node Injection Attacks

The maximization of the cross-entropy loss in PGD attack [41] 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 1.3 (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 1.3 (b) that by maximizing the max-margin loss, the above scenario can be avoided. The max-margin loss $\boldsymbol{\ell}_{mm}$ is given by

$$\boldsymbol{\ell}_{mm}(\mathbf{f}_\Theta) = -\mathbf{f}^y_\Theta(x) + \max_{i \neq y} \mathbf{f}^i_\Theta(x). \tag{3.2}$$

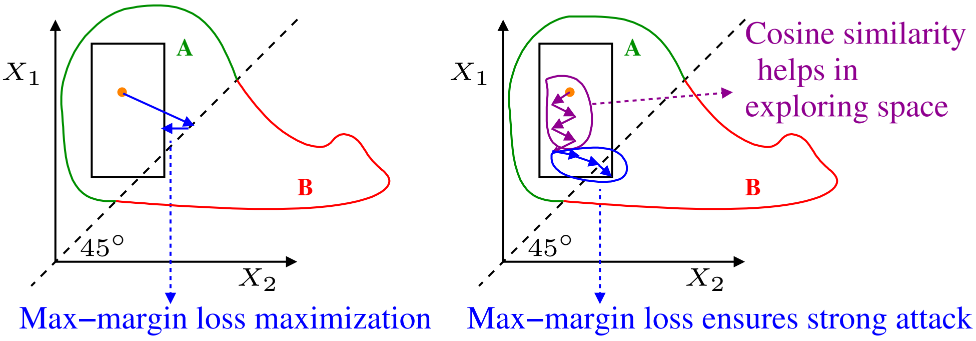
Thus maximizing the max-margin loss leads to the maximization of the margin between the true class and the highest confident false class via decreasing the confidence of the true and maximizing the confidence of the false class. Motivated by this, we propose margin aware attack (MAA), which maximizes max-margin loss to generate the attack. We empirically show in Tables-3.1, and 3.3 that MAA attack is stronger than PGD [41] by over 1.3% on Citeseer [28] dataset.

3.3.2 Leveraging Directional Similarity in MAA (MDA)

Max-margin loss helps in achieving a larger norm between original and attacked node features. However, Figure 1.3 (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 the attack diversity along with maximization of max-margin loss ensures a stronger node injection attack.

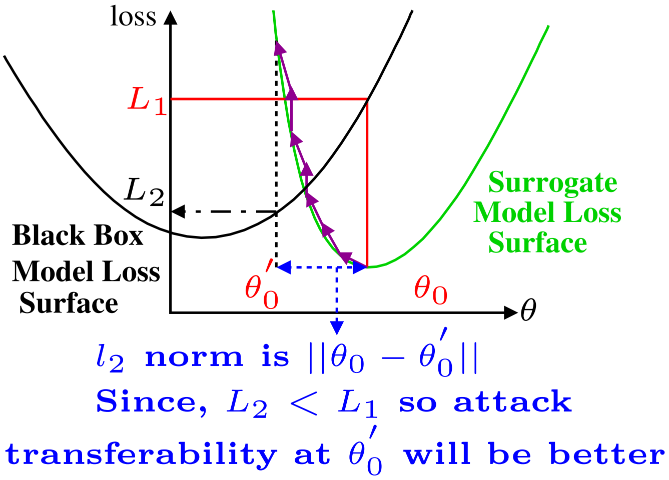
Cosine similarity can only help in exploring the constraint space rather than generating stronger attacks. Based on the aforesaid analysis, we use cosine similarity along with max-margin loss for initial k iterations of the proposed attack in order to get a good initialization and later shift to the only max-margin loss. The MDA loss



X_1
A
B
45°
X_2
Max−margin loss maximization
X_1
A
B
45°
X_2
Cosine similarity helps in exploring space
Max−margin loss ensures strong attack

(a) (b)

Figure 3.1: Pictorial representation of binary classifier's output space. **(a)** Maximizing max-margin loss leads to traversal in the orthogonal direction of the decision boundary. **(b)** Minimization of cosine similarity leads to exploring the space (purple) and the attack is able to fool the model on using max-margin loss later (blue).



loss
L_1
L_2
Black Box Model Loss Surface
Surrogate Model Loss Surface
θ_0'
θ_0
θ
l_2 norm is $||\theta_0 - \theta_0'||$
Since, $L_2 < L_1$ so attack transferability at θ_0' will be better

Figure 3.2: 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.

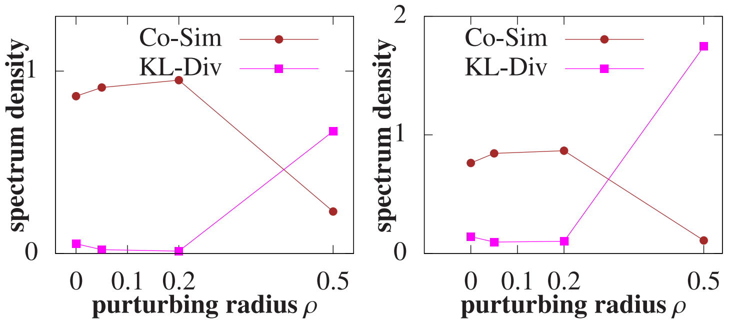
$\boldsymbol{\ell}_{mda}$ is given by:

$$\boldsymbol{\ell}_{mda}(\mathbf{f}_\Theta(X \oplus X_{atk})) = \ell_{mm}(\mathbf{f}_\Theta(X \oplus X_{atk})) - \gamma \times \boldsymbol{\ell}_{co-sim}(\mathbf{f}_\Theta(X \oplus X_{init}), \mathbf{f}_\Theta(X \oplus X_{atk})), \tag{3.3}$$

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. (3.2) and the cosine similarity loss $\boldsymbol{\ell}_{co-sim}$ is defined as follows:

$$\boldsymbol{\ell}_{co-sim}(\mathbf{f}_\Theta(X \oplus X_{init}), \mathbf{f}_\Theta(X \oplus X_{atk})) = \frac{\mathbf{f}_\Theta(X \oplus X_{init})^\top \mathbf{f}_\Theta(X \oplus X_{atk})}{||\mathbf{f}_\Theta(X \oplus X_{init})||\ ||\mathbf{f}_\Theta(X \oplus X_{atk})||}. \tag{3.4}$$

To develop a better understanding, we consider an example setting of binary classification using a trained classifier GNN f_Θ. Figure 3.1 represents the 2D output space, where X_1 and X_2 correspond to the output logits of the classifier. The black



(a) on Cora (b) on Citeseer

Figure 3.3: Similarity between the distribution of spectrum density for the perturbed surrogate and original black box models. The proposed MDA attack is used for evaluation.

dotted line is the zero-margin line, where $X_1 = X_2$. The black-coloured box is the constraint set in the output space, where outputs of all possible attacks in the input space, following the threat model will lie. Note that the constraint set is originally defined in the input space, but here we have propagated it through the network and represented it in the output space. It is guaranteed to be some closed shape as shown by Gowal et al. [31]. We assume it to be a rectangle for simplicity of explanation.

Let for some input feature x_i, the trained network f_Θ maps it to the red point in output space. For a given length of a gradient ascent step $\boldsymbol{\epsilon}$, the ideal solution of max-margin loss maximization (Figure 3.1 (a)) will lead to a direction **orthogonal** to the zero margin dotted black line. Hence, if MAA is used to generate an attack, max-margin loss maximization can't generate an attack to fool the model. However, as shown in Figure 3.1 (b), if initially cosine similarity between attacked and randomly initialized features of the injected nodes is minimized, then it would lead to exploration of the space. This is because we are minimizing the cosine similarity with respect to random initialization. Maximization of max-margin loss thereafter leads to miss-classification. Thus, on using a cosine similarity loss, the attack can fool the model to miss-classify x_i. On the other hand, by merely maximizing the max-margin loss, the attack is able to explore only a small fraction of the area of the attack constraint bound and is therefore suboptimal. This highlights the presence of directional constraint (maximization occurs in the direction orthogonal to the dotted black line) on using max-margin loss.

Motivated by the aforesaid hypothesis, we minimize the cosine similarity between the feature outputs of randomly initialized and attacked injected nodes. This helps in exploring the space while maximizing the max-margin loss helps in generating strong attacks.

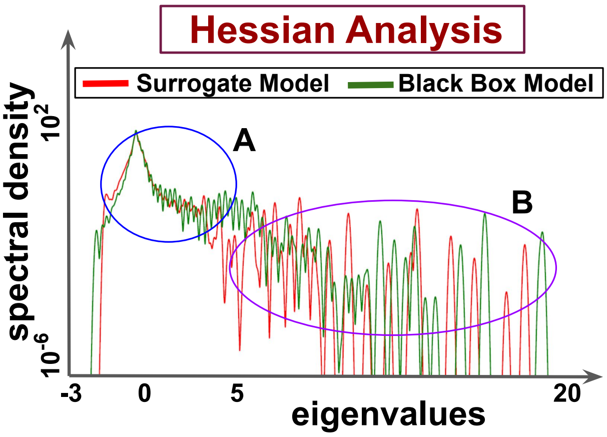


Figure 3.4: Eigenvalues spectrum of the loss calculated on the training set. **A**: smaller eigenvalues have similar density. **B**: dominant larger eigenvalues have different densities. This indicates that the two loss landscapes are very different.

3.3.3 Transferability awareness in MDA (MDTA)

As demonstrated by Lord et al. [39], 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. [26] 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 f, parameterized by Θ, we get, perturbed weights as follows:

$$\widetilde{\Theta} = \underset{\Theta \in \mathcal{M}(\Theta)}{\mathbf{argmax}} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\ell}_{\text{ce}}(\mathbf{f}_{\Theta}(x_i), y_i). \tag{3.5}$$

To develop a better understanding, we present a motivational example in Figure 3.2, 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 ℓ_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 3.4. 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 [26]. 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 3.3. 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 3.4 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 rates.

3.3.4 Rethinking $\boldsymbol{\ell}_\infty$ Norm for Gradient Ascent

Past works Chen et al. [15], Madry et al. [41] naively use the sign of the gradients for crafting attacks. We hypothesize that using the sign of the gradient (in case of $\boldsymbol{\ell}_\infty$ attack) in the attack generation step leads to a similar increase or decrease (by $\boldsymbol{\epsilon}$) 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 Tables 3.1 and 3.3, this simple observation leads to improvements upto 7% over $\boldsymbol{\ell}_\infty$ norm.

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: $\widetilde{\Theta} = \underset{\Theta \in \mathcal{M}(\Theta)}{\textbf{argmax}} \frac{1}{d} \sum_{i=1}^{d} \ell_{\text{ce}}(\mathbf{f}_\Theta(x_i), y_i)$;   % $\ell_{\text{ce}}$ denotes the standard cross-entropy loss. %
5: $A_{atk} = A_{atk} + \textbf{sign}\left(\nabla_{A_{atk}} \times \boldsymbol{\ell}_{\text{ce}}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk}))\right)$;
6: $A'_{atk} = \textbf{top}_e(A_{atk})$;
7: $A_{atk} = \textbf{round}(A'_{atk}, 0, 1)$;
8: $\boldsymbol{\ell}_{mda}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk})) = \ell_{mm}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk})) - \gamma \times \ell_{co-sim}(\mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{init}), \mathbf{f}_{\widetilde{\Theta}}(X \oplus X_{atk}))$;
9: for $iter = 1$ to $I$ do
10:     if $iter >= \frac{2}{3} I$ then $\gamma = 0$;
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:     $X_{atk} = \textbf{clamp}(X_{atk}, MAX, MIN)$;   % MIN and MAX depend on range of original nodes. %
13: $G' = (A \oplus A_{atk}, X \oplus X_{atk})$;
```

3.4 Resilient Node Injection Attack (R-NIA)

Motivated by the insights discussed above, 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 [66, 74], 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 3.4. 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. [15]), 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 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 3.10 (a).

Table 3.1: Robust Accuracy (%) on Cora and Citeseer datasets for black box setting. CN denotes graph convolutional network and AT denotes graph attention network.

	Cora				Citeseer			
Model	CN→CN	CN→AT	AT→CN	AT→AT	CN→CN	CN→AT	AT→CN	AT→AT
AFGSM [63]	35.78	38.65	39.02	37.32	24.78	26.87	29.24	27.47
TDGIA [79]	32.10	35.14	35.98	34.23	20.97	22.45	23.87	21.94
MetaAttack [63] (one time)	30.74	32.21	33.46	31.01	22.31	24.78	24.14	22.75
MetaAttack [63] (sequential)	31.01	32.85	33.21	31.51	23.03	24.91	24.52	23.12
PGD [41]	33.21	36.42	37.46	35.68	21.42	23.14	24.02	22.68
G-NIA [55]	35.17	37.87	38.21	36.01	23.61	23.56	24.49	23.41
HAO [15]	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 [41]	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**

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 $G^{'}$.

3.5 Experiments Results

3.5.1 Training Details

We empirically test our models on six datasets, *i.e.*, Cora [70], Citeseer [28], Flickr [71], Aminer [54], Amazon2M [43], and Twitter [7, 37]. Table 3.5 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 [60] means that the attack is generated on graph convolutional network GCN [36] 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 $\{10^3, 10^6\}$.

3.5.2 Evaluation Results

We present the experimental results of PGD [41], 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 ℓ_2 (R-NIA) and ℓ_∞ norm (R-NIA-ℓ_∞) of gradients in Table 3.1 for Cora and Citeseer datasets and Table 3.3 for Flickr dataset. Further results on large datasets, like Aminer, Amazon2M, and Twitter, are present in Table-3.2. We utilize PGD [41], PGD with Harmonious Adversarial Objective (HAO) [15], TDGIA [79], MetaAttack [63], G-NIA [55] and Approximate FGSM (AFGSM) [63] 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 [55] aims to craft a single node which can fool certain target nodes of the graph. For a fair comparison, we modified the optimization problem in eq-8 of [55], to maximize the loss for all the injected nodes in the graph instead of a single node. We further modified [55] to attack all the original nodes. We now discuss our experimental findings.

Is using ℓ_∞ norm of gradients [15, 41] the best choice? As seen in the bottom half of Tables 3.1 and 3.3, we find that using ℓ_2 norm of the gradients instead of ℓ_∞ leads to a significant drop in the classification accuracy. Due to enhanced diversity in the features in the case of ℓ_2 norm, we observe stronger attack with upto 7% higher attack strength.

Is maximizing Cross-Entropy loss a good choice in PGD attack formulation? To investigate this, we compare the results of PGD and MAA in Tables 3.1

Table 3.2: 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 [41] baseline as well. On larger graphs, more gains over the PGD baseline are observed on injecting larger number of nodes and edges.

	Flickr				Aminer			
Model	CN→CN	CN→AT	AT→CN	AT→AT	CN→CN	CN→AT	AT→CN	AT→AT
PGD (200,500) [41]	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) [41]	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) [41]	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) [41]	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**

and 3.3. We observe that the performance of MAA is better than PGD by over 1.5% on Cora and Flickr datasets and 2% on the Citeseer dataset.

Can maximizing max-margin loss alone lead to sub-optimal search in output constraint space? Based on Tables 3.1 and 3.3, 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. On minimizing cosine similarity helps in improved optimization of the attack leading to stronger attacks.

Based on these 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. It however helps in improving the search of 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 Tables 3.1 and 3.3, 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 [15] is the

Table 3.3: Robust Accuracy (%) on Flickr dataset. CN denotes graph convolutional network and AT denotes graph attention network

Model	CN→CN	CN→AT	AT→CN	AT→AT
PGD [41]	49.24	51.02	53.98	52.64
HAO [15]	47.65	48.95	48.68	47.98
MAA	48.74	49.65	51.98	51.06
MTA	48.32	48.79	51.03	50.97
MDA	49.42	50.01	51.86	51.15
MDTA	46.98	48.12	49.37	48.01
R-NIA-ℓ_∞	**45.71**	**47.01**	**48.23**	**46.82**
-ℓ_2				
PGD	41.03	43.01	44.85	44.03
MAA	40.64	41.63	42.98	42.64
MTA	40.45	40.98	42.61	42.34
MDAT	41.36	42.45	42.76	43.79
MDTA	39.66	41.02	42.31	42.54
R-NIA	**38.41**	**39.98**	**41.06**	**41.49**

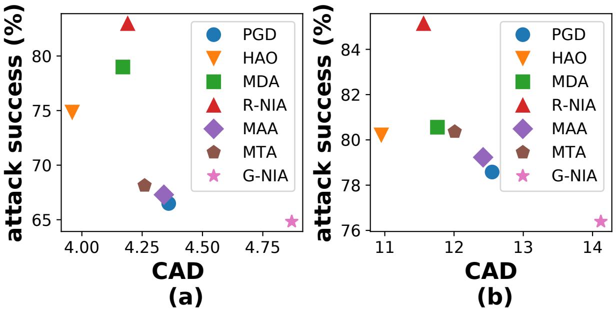
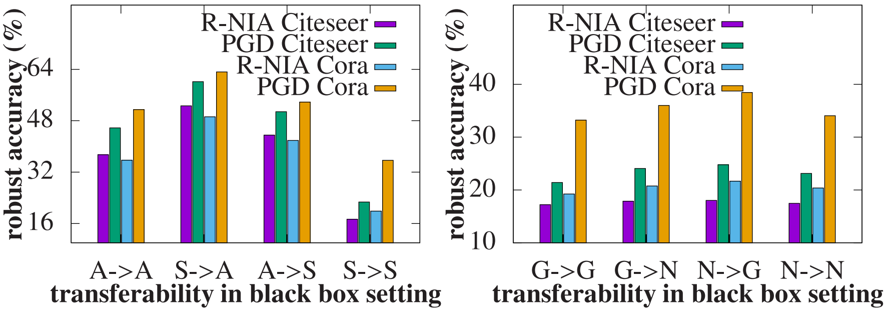


Figure 3.5: Results on attack success vs CAD, where attack success= 100− robust accuracy on Cora (a) and Citeseer (b).

stronger baseline in Table-3.1, the proposed R-NIA shows improvements upto 8.18% over it on the Cora dataset and around 5% on the Citeseer dataset.

3.5.3 Effect of Graph Size and number of injected nodes and edges

As shown in Table- 3.2, we observe that on adding a larger number of nodes and edge the proposed R-NIA attack improves the performance over PGD by upto 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 (for graph size ref-Table-3.5) like Amazon2M and Twitter datasets. On Amazon2M, we outperform PGD by upto 6.09% and by around 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. R-NIA is significantly stronger than PGD, even on larger graph datasets.

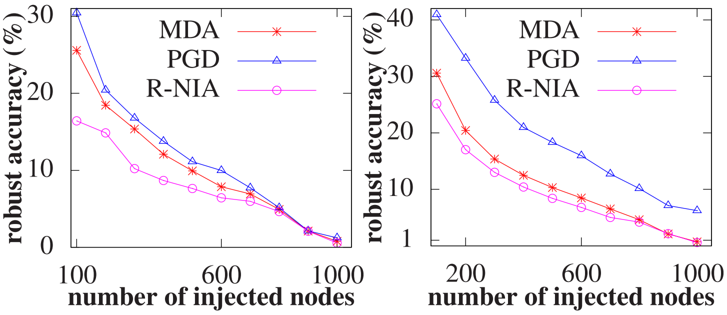


(a) (b)

Figure 3.6: Impact on transferability between (a) adversarial (A) and standard (S) trained models; (b) non-local GCN (N) and GCN (G) models.

3.5.4 Transferability on robust models

Figure 3.6 (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 transferability as compared to PGD when the attack is generated using S/A and evaluated on A/S, respectively. Figure 3.6 (b) depicts the results of transferability experiments on non-local 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.

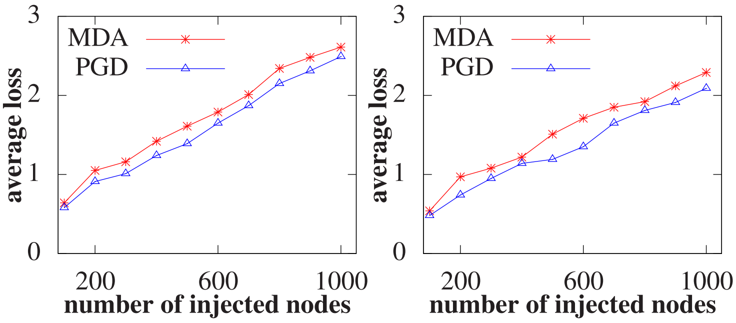


(a) on Cora (b) on Citeseer

Figure 3.7: Impact on robust accuracy on number of injected nodes using GCN architectures.

3.5.5 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 3.4. 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



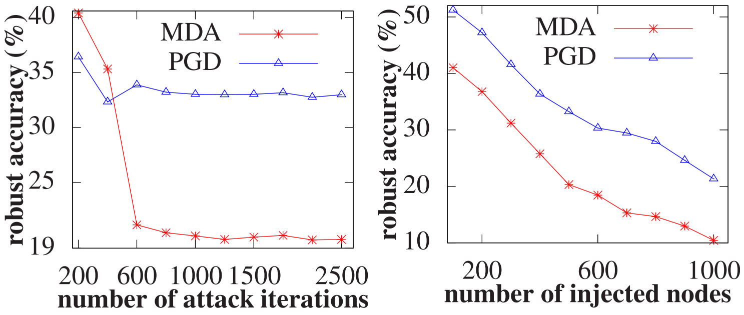
(a) on Cora (b) on Citeseer

Figure 3.8: Impact on average loss on number of injected nodes for MDA and PGD models.

Table 3.4: Effect of weight perturbation on Cora dataset. Robust Accuracy (%) on Cora dataset for different weight perturbation magnitude ρ is shown.

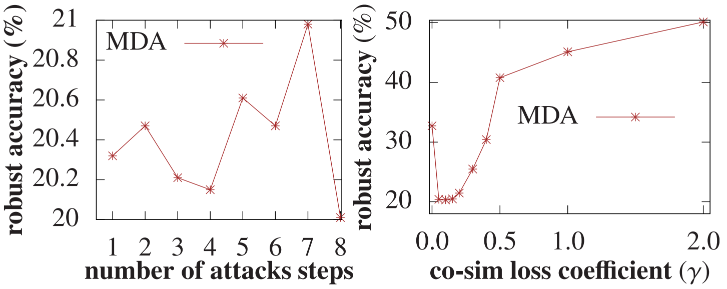
Model	CN→CN	CN→AT	AT→CN	AT→AT
MAA	32.71	35.42	36.67	34.21
MTA				
$\rho = .05$	32.04	34.12	35.28	34.28
$\rho = 0.1$	32.16	33.06	34.66	34.06
$\rho = 0.2$	**31.87**	**32.48**	**34.21**	**33.98**
$\rho = 0.3$	32.45	33.98	35.36	34.87
$\rho = 0.5$	65.31	67.25	55.06	65.84
MAA-ℓ_2	26.22	29.04	30.01	28.65
MTA-ℓ_2				
$\rho = .05$	26.33	28.01	29.64	**28.41**
$\rho = 0.1$	**26.03**	27.66	29.31	28.74
$\rho = 0.2$	26.06	**26.36**	**29.07**	28.45
$\rho = 0.3$	29.45	29.78	33.48	33.76
$\rho = 0.5$	58.64	60.75	59.01	60.32

the effect of changing the constraints on the number of nodes and edges injected in the threat model, as shown in Figures 3.7 and 3.8, 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 3.10 (a), 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 3.10 (b) 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 3.9 (a), 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 3.9 (b) shows that MDA outperforms PGD on increasing the number of injected edges.



(a) (b)

Figure 3.9: Effect of the number of (a) attack steps for generating X_{atk} and (b) injected edges on Cora.



(a) (b)

Figure 3.10: Effect of (a) number of attack steps to generate attacked adjacency matrix (A_{atk}) and (b) cosine similarity loss coefficients (γ) on Cora.

3.5.6 Imperceptibility Study

It is important to ensure that after node injection attack, the graph still remains imperceptible [15, 56]. For this, we use the closest attribute distance (CAD) as the metric [56, 79]. 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 3.5, 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. [56], there is a strong correlation between CAD and two other imperceptibility metrics: Smoothness [15] and Graph FD [56]. Therefore, CAD is a reliable metric.

3.6 Additional Details

3.6.1 Dataset and Attack Details

We consider six different graph datasets for evaluating the proposed R-NIA attack. The details on the number of nodes and edges of the datasets are given in Table-3.5. While the Twitter dataset has 4,16,52,230 nodes and 1,46,83,64,884 edges, due to limited resources, we could not train on 4,16,52,230 nodes. Therefore, we used randomly selected 41,65,223 nodes and corresponding edges for training. Unless specified, we perform a thousand iterations of attack for all the baselines and the proposed methods. We compare the effect of the number of attack steps and the attack strength in Figure 3.9 (b). As can be seen, the MDA as well as PGD attacks saturates till 1000 attack iterations, and thus 1000 iterations are sufficient enough to get an estimate of the attack strength. For all the experiments, we use the codebase of Zheng et al. [77] [1]. We use RTX-2080 for all the experiments, and on COra dataset, the attack takes around 45 seconds.

Table 3.5: Number of nodes $\boldsymbol{N}$ and edges $\boldsymbol{E}$.

	Cora	Citeseer	Flickr	Reddit	Aminer	Amazon2M	Twitter
$\boldsymbol{N}$	2.68K	3.19K	89.25K	2.32M	6.59M	2.44M	41.65M
$\boldsymbol{E}$	5.14K	4.17K	4.49M	11.6M	2.87M	61.85M	1468.36M

3.6.2 Error Analysis

On performing five reruns of R-NIA on Cora dataset, we observed 17.09% as the mean accuracy and a standard deviation of 0.12% in robust accuracy.

3.6.3 Limitations

Choosing the appropriate value of γ, which is the weight given to co-sim loss in the proposed MDA attack, might ‘require some tuning efforts. As shown in Figure 3.10 (b), the proposed R-NIA is sensitive to large variations in the value of γ.

3.6.4 Societal Impact

Since the popularity of graph neural networks is increasing, it is crucial to understand their robustness in order to build more reliable models in the future. We hope this work will open new avenues in building stronger node injection attacks in future and would also be incorporated into adversarial training frameworks to make GNNs more robust.

3.7 Conclusion

In this chapter, we highlight the need to rethink node injection attacks on GNNs. Firstly, we find that using cross-entropy loss is not the best choice and propose to use max-margin loss in the PGD attack formulation. Further, we observe that maximizing max-margin loss may lead to sub-optimal search within the constraint ball. Thus, in addition to max-margin loss maximization, we minimize cosine similarity between random initialization and attacked features of injected nodes to search the constraint space. Since transferability is important in the context of black box attacks, we also perform a weight perturbation in $\boldsymbol{\ell}_2$ norm before crafting the attack to get a model whose loss surface can generalize better. We show that the use of $\boldsymbol{\ell}_\infty$ norm restricts the diversity of attack features since same magnitude of update is enforced in all

[1] https://github.com/THUDM/grb

feature dimensions. We propose to use $\boldsymbol{\ell}_2$ norm perturbation instead for gradient ascent. We demonstrate improved performance over graph robustness benchmark [77] models. We show that our method is generalizable to larger graphs, like Twitter and Amazon2M, and different classes of GNNs. It also improves black box attack strength in case of adversarially trained models and remains imperceptible making it difficult to detect it.

Chapter 4

Understanding and Improving Adversarial Robustness of Vision Transformers

4.1 Introduction

In this chapter, we propose a conjecture that the presence of intermediate softmax in attention blocks of vision transformers makes them inherently vulnerable to the gradient masking effect. We rethink the understanding of the adversarial robustness of VITs and discuss more fundamental causes of gradient masking in VITs that leads to poor attack strength on using standard attacks, like PGD [41], GAMA [49], and CW [11] attacks, which have demonstrated good attack strength on CNNs. Inspired by Yu and Xu [72], we hypothesize that the reason for gradient masking is VITs is the floating point underflow error which occurs due to softmax calculation in every attention block in VITs. As shown in Figure 1.4 (a), as highlighted in red, we observe that in the case of VITs, a larger scale of pre-softmax outputs can result in floating point underflow. This leads to a false estimate of the gradient, resulting in a weaker attack. But, we can overcome the floating point errors by scaling down the pre-softmax outputs by the right scaling factor. As shown in Figure 1.4 (b), on scaling down the pre-softmax outputs in every attention block the proposed Adaptive Attention Scaling (AAS) attack leads to a boost upto 3% over standard PGD [41] attack on CIFAR10 dataset. In the case of CNNs (shown in red) as demonstrated by [72], scaling down the logits help in improving the attack strength by overcoming floating point errors. We hypothesize that the effect of gradient masking is much more intense in the case of VITs because of softmax functions present in attention blocks. This is evident from the improved gains on simply scaling the pre-softmax outputs manually (shown in blue) over the same method applied to logits in the output space of CNNs (shown in red). We show strong empirical evidence for our hypothesis and point out a more fundamental reason for the poor performance of VITs on adversarial attacks.

In this chapter, firstly, we define the threat model and setup in Section-4.2, then in Section-4.3 we present some of our findings to justify the proposed conjecture. Motivated by the proposed conjecture, we propose Adaptive Attention Scaling (AAS) attack in Section-4.4. Further, in Section-4.5 we propose to use the proposed ASA attack at regular intervals in adversarial training. Finally, we present our experimental results in Section-4.6. On VITs, we demonstrate improved performance over all the existing attacks and by incorporating the AAS attack with existing adversarial training methods, we outperform them.

4.2 Preliminaries

Threat model. Let f_θ denote a deep neural network parameterized by θ mapping input sample X to R^N where N is the number of classes. The goal of an adversary is to fool the model while restricting the perturbation within a threat model. The threat model is defined by:

$$||X^{'} - X||_p < \epsilon \;\; \text{and} \;\; f_\theta : X \rightarrow R^N, \tag{4.1}$$

where p represents the type of ℓ_p perturbation norm, $X^{'}$ represents the perturbed image and ϵ is the maximum allowed ℓ_p perturbation bound. In this work, we consider ℓ_∞ perturbation norm.

Background. Some of the past findings that motivate our proposed attack are discussed as follows:

• As shown in Figure 4.2 (second row, left), it is well known that the gradients from a robust model are perceptually aligned. Recently, Ganz et al. [27] demonstrated that if the gradients from a model are perceptually aligned, then it implies that the model is adversarially robust.

• Through extensive human evaluation, Zhang et al. [76] demonstrated that LPIPS distance is a good perceptual model. Motivated by this, it has been further used in Laidlaw et al. [38] to define a perceptual threat model. Addepalli et al. [1] proposed to minimize the LPIPS distance between the clean and adversarial images to achieve robustness to larger perturbation bounds by ensuring that the images don't change their original class perceptually.

4.3 Motivation: Gradient Masking in VITs

Yu and Xu [72] demonstrated that the attacks involving softmax calculation may suffer from floating point underflow error leading to a suboptimal attack. Further, VITs use softmax to calculate the attention weights in every attention block. Therefore, the effect of floating point error should be more severe in the case of VITs as compared to CNNs. Based on this, we propose the following conjecture:

Conjecture 1: The presence of intermediate softmax in attention blocks of vision transformers makes them inherently vulnerable to the gradient masking effect.

Justification: To verify this, we plot the histogram of the difference between the largest and the second largest values before taking softmax (Δ) for different attention blocks of the VIT-B16 [25] model in Figure 4.1 (first row). The histogram is plotted for the ImageNet-100 dataset, which is a 100-class random subset of ImageNet-1K [23] using a normally trained VIT-B16 model. We observe that for many images, the difference is significantly high and would lead to floating point underflow errors on taking exponential in the softmax calculations. As opposed to CNNs, since the floating point underflow error will occur in intermediate layers of VITs, the effect would simply magnify. Therefore, depending on the amount of floating point error, it can even lead to significant gradient masking. On using the proposed attack, as observed in Figure 4.1 (second row), the scale of the pre-softmax outputs gets significantly reduced, which helps in overcoming the gradient masking effect. We further analyze the effect of manually scaling down the features used for softmax computation on CIFAR10 and CIFAR100 datasets in Table 4.1. We observe that even using the same scaling factor for all the attention blocks can give a boost of up to 1.67% in the PGD-100 and 2.7% in FGSM attack strengths on the CIFAR10 dataset. On CIFAR100, improved attack strength of up to 2% is observed on the PGD-100 attack.

4.4 Adaptive Attention Scaling (AAS) Attack

The process of finding the optimal combination of scaling factor is difficult because different attention blocks can have different scaling factors. We analyze if these scaling values can be found automatically using gradient-based optimization before generating the attack. Zhang et al. [76] performed a human study and demonstrated that LPIPS distance is a good perceptual metric. The LPIPS distance is a feature-level distance defined for a set of inputs to a given model. For a given set of clean (x) and adversarial ($x^{'}$) images, LPIPS distance is the sum of the normalized ℓ_2 distances between the features of the two images taken after every attention block. A greater value of LPIPS distance indicates that the two images are perceptually dissimilar to each other [1, 38].

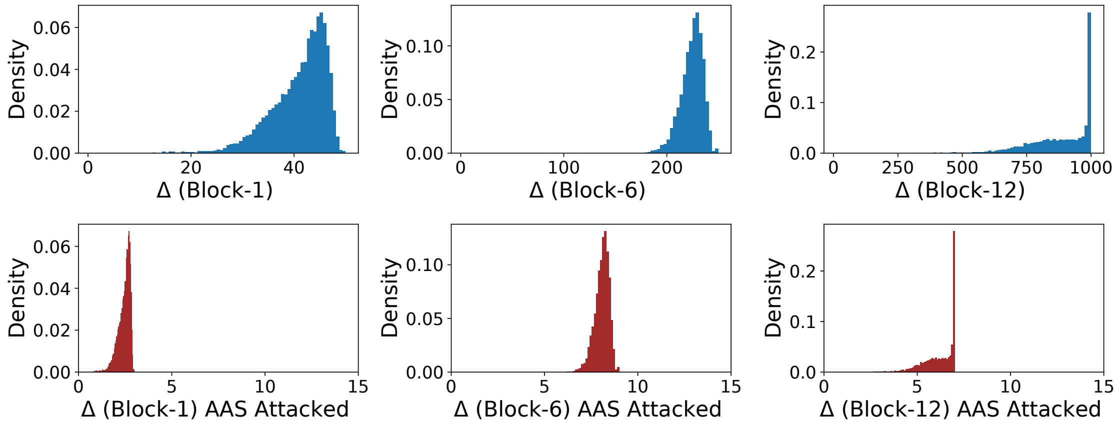


Figure 4.1: **Histogram of the difference between largest and second largest pre-softmax outputs** for different blocks of the normally trained VIT-B16 on ImageNet-100 dataset. As shown in the second row, the proposed AAS attack successfully downscales these values.

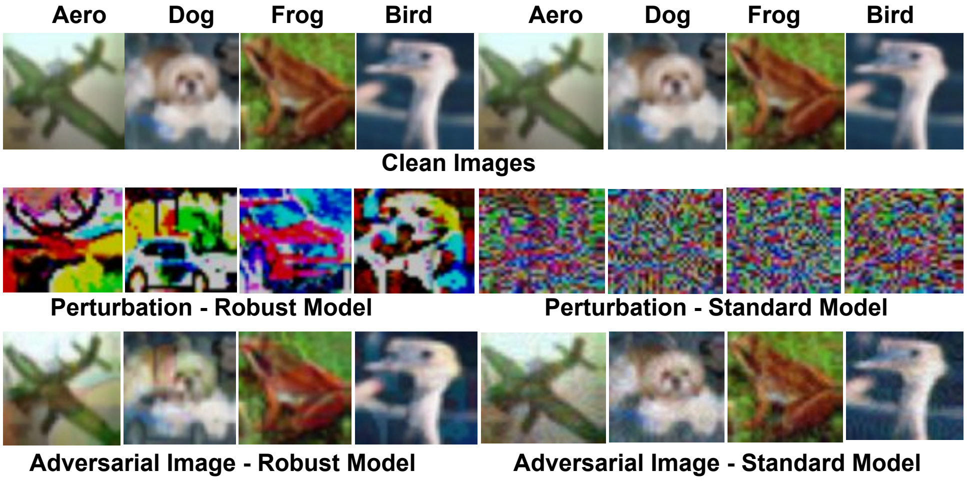

Aero
Dog
Frog
Bird
Aero
Dog
Frog
Bird
Clean Images
Perturbation - Robust Model
Perturbation - Standard Model
Adversarial Image - Robust Model
Adversarial Image - Standard Model

Figure 4.2: **Nature of Gradients of an Adversarially Robust Model** [1]**.** Attack obtained using a robust model at $\epsilon = 16/255$ on CIFAR10 leads to perceptually aligned gradients (second row, left). While generate the same using a standard model doesn't give perceptually aligned gradients (second row, right).

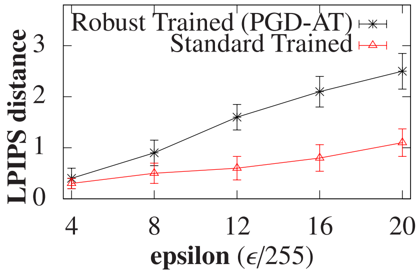


Figure 4.3: **LPIPS distance** [76] **between the clean and perturbed images** [1] is calculated using ResNet18 model and trained using PGD-AT[41]. The perturbations are generated using a robust (black) and a non-robust (red) model. Since the perturbations generated from a robust model are perceptually aligned, LPIPS distance is higher.

Definition: In this work, instead of using a set of clean and adversarial images for calculating LPIPS distance, we perturb the pre-softmax output scaling factors denoted by $S = \{s_1, s_2, ..., s_m\}$ and calculate the LPIPS distance for a given set of normal and perturbed models. Here, m is the number of attention blocks in the VIT. Thus, for an image x, the LPIPS distance, in our case, is defined by:

$$\textbf{LPIPS}(f_{\theta(S)}, f_{\theta(S')}) = \sum_{i=1}^{m} \frac{||f_{\theta(s'_i)}(x) - f_{\theta(s_i)}(x)||_2}{||f_{\theta(s'_i)}(x)||_2 ||f_{\theta(s_i)}(x)||_2}. \tag{4.2}$$

Table 4.1: **Manually scaling down the pre-softmax outputs** leads to enhanced attack strength of PGD attacks [41] on CIFAR10 and CIFAR100 datasets.

Scaling	CIFAR10		CIFAR100	
Factor	Clean	PGD-100	Clean	PGD-100
1	**87.43**	61.10	**62.47**	30.01
10^{-4}	86.23	60.23	61.13	28.45
10^{-3}	86.79	60.14	61.42	**27.76**
10^{-2}	87.01	**59.43**	61.79	28.01
10^{-1}	87.22	60.87	62.04	28.12
10	85.71	66.78	61.03	31.78

Table 4.2: **Effect of perturbing the scaling factors using different loss functions**. A feature level distance like LPIPS [76] leads to better attack strength by overcoming gradient masking effectively.

Loss Function	CIFAR10		CIFAR100	
(Perturb Scales)	Clean	PGD-100	Clean	PGD-100
No Attack	**87.43**	61.10	**62.47**	30.01
Scaling (0.01)	87.01	59.43	61.79	28.01
Cross-Entropy	86.84	60.13	61.48	29.11
Max-Margin	86.47	59.78	61.23	28.46
GAMA [49]	86.74	59.85	61.41	28.31
LPIPS (ASA)	87.31	**58.01**	62.03	**27.02**

As shown in Figure 4.2 (second row , left), the gradients calculated from an adversarially robust model are perceptual in nature [27]. Through a human study, Zhang et al. [76] demonstrated that LPIPS is a good perceptual metric. Thus, maximizing LPIPS distance while perturbing the pre-softmax scaling factors should lead to finding the scaling factors which can produce gradients that are more perceptually aligned. As demonstrated by Ganz et al. [27], perceptually aligned gradients imply adversarial robustness. Therefore, by making the gradients more perceptually aligned by maximizing the LPIPS distance between the original and perturbed models, we tend to overcome gradient masking and enhance the adversarial robustness of the model. In order to make this claim stronger, as shown in OAAT [1], we analyze if calculating LPIPS distance using a robust model can indeed differentiate between the perturbations generated from standard verses adversarially robust models. As shown in Figure 4.3, since the perturbations generated from a standard trained model (Figure 4.2 second row, right) are less perceptually aligned, LPIPS distance between the clean images (first row) and the corresponding adversarial images (third row, right) perturbed using the adversarial attack generated from a standard model is less as compared to LPIPS distance between clean images (first row) and corresponding adversarial images (third row, left) generated by attacking a robust model. Therefore, maximizing LPIPS distance between the features of a normal and perturbed model should lead to the generation of perceptually aligned gradients, thereby overcoming the gradient masking effect.

Why LPIPS is better? To build a better understanding of whether maximizing a feature level distance, like LPIPS, is better than other standard attacks that use cross-entropy or max-margin loss, we analyze the effect of using different loss functions to perturb the scaling factors of pre-softmax outputs in Table 4.2. Since gradient masking is present at *feature level* in VITs, using a feature-level attack, like LPIPS, can give improved performance by up to $1-2\%$ over standard attacks. We observe that manually scaling the pre-softmax outputs outperforms the standard attacks, like GAMA, PGD, and CW. This shows the importance of attacking at the feature level rather than the output space.

Motivated by the above discussion, we propose to maximize LPIPS distance for perturbing the pre-softmax output scaling factors. The proposed AAS attack is presented in Algorithm 2. As common in practice, we initialize the attack with a standard normal distribution (L3). We perturb the pre-softmax output scaling factors in the attention blocks by maximizing the LPIPS distance before generating the actual attack (L4-L7). Later, the adversarial attack is generated using the model whose scaling factors is perturbed (L8).

4.5 Adaptive Attention Scaling Adversarial Training

Motivated by Conjecture-1, since VITs are inherently subjected to the gradient masking effect, the adversarial training of VITs should be difficult. This is indeed observed in the prior works [22, 44], which demonstrate the need to use gradient clipping, larger weight decays and epsilon warmup to stabilize and achieve improved robustness on VITs on performing adversarial training. Though these tricks help

Algorithm 2 Adaptive Attention Scaling (AAS) Attack

```
1: Input: Network $f_{\theta(S)}$ where $S = \{s_1, s_2, ..., s_m\}$ is the pre-softmax scaling factor
   and $m-1$ is the number of attention blocks in the model. Training Dataset
   $\mathcal{D} = \{(x_i, y_i)\}$, and $M$ training mini-batches of size $n$;
2: for iter = 1 to $M$ do
3:     $\delta = \mathcal{N}(0, 1)$;
4:     for steps = 1 to 10 do
5:         $\delta = \delta + \nabla_S \textbf{LPIPS}(f_{\theta(S)}(x_i), f_{\theta(S')}(x_i))$;
6:         $S' = \textbf{Clamp}(S + \delta, 10^{-r}, 1)$; % to prevent zero scaling factors, we considered $r = 7$%
7:     $S = S'$;
8: Generate the attack on the perturbed model;
```

Algorithm 3 Adaptive Attention Scaling Adversarial Training (AAS-AT)

```
1: **Input:** Network $f_{\theta(S)}$ where $S = \{s_1, s_2, ..., s_m\}$ is the pre-softmax scaling factor and $m - 1$ is the number of attention blocks in the model. Training Dataset $\mathcal{D} = \{(x_i, y_i)\}$, Adversarial Threat model: $\ell_\infty$ bound of radius $\varepsilon$, coefficient of KL divergence term $\beta$, Cross-entropy loss $\ell_{CE}$, number of epochs E, $M$ training mini-batches of size $n$, Maximum Learning Rate LR$_{max}$, Frequency of AAS attack $\lambda$;
2: **for** epoch = 1 **to** $E$ **do**
3:     LR $= 0.5 \cdot \mathrm{LR}_{max} \cdot (1 + \mathbf{cosine}((\mathbf{epoch} - 1)/E \cdot \pi))$;
4:     **for** iter = 1 **to** $M$ **do**
5:         **if** epoch$\%\lambda == 0$ **then**
6:             $\delta = \mathcal{N}(0, 1)$;
7:         **for** steps = 1 **to** 10 **do**
8:             **if** epoch$\%\lambda == 0$ **then**
9:                 $\delta = \delta + \nabla_S \mathbf{LPIPS}(f_{\theta(S)}(x_i), f_{\theta(S')}(x_i))$;
10:                $S' = \mathbf{Clamp}(S + \delta, 10^{-r}, 1)$; % prevent zero scaling factors, $r = 7\%$
11:            **else**
12:                $\delta = 0.001 \cdot \mathcal{N}(0, 1)$;
13:                $\delta = \delta + \varepsilon_{\text{asc}} \cdot \text{sign}\,(\nabla_\delta \mathbf{KL}(f_\theta(x) || f_\theta(x + \delta)))$;
14:                $\delta = \mathbf{Clamp}\ (\delta, -\varepsilon_{\text{asc}}, \varepsilon_{\text{asc}})$;
15:                $\widetilde{x} = \mathbf{Clamp}\ (x + \delta, 0, 1)$;
16:        **if** epoch$\%\lambda == 0$ **then**
17:            $S = S'$;
18:        **else**
19:            $\mathcal{L}_{\text{TR}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{\text{CE}}(f_\theta(x_i), y_i) + \beta \cdot \mathbf{KL}(f_\theta(x_i) || f_\theta(\tilde{x}_i))$;
20:            $\theta = \theta - \mathrm{LR} \cdot \nabla_\theta(\mathcal{L}_{\text{TR}}(\theta))$;
```

in stabilizing the VITs, the reason for their effectiveness is not well known. We hypothesize that the reason for the sub-optimal adversarial robustness on training VITs is the gradient masking effect caused due to the large scale of pre-softmax outputs.

To mitigate the associated gradient masking effect, we propose to use the proposed AAS attack in every few epochs of training while training on standard attacks for the remaining epochs. More specifically, we propose to perturb the pre-softmax scaling weights every few epochs using LPIPS loss maximization to ensure that the scale of the pre-softmax values is within the suitable range. This will help in preventing floating point underflow errors resulting in a better estimate of the gradients and stronger attack generation during training.

We present the proposed AAS-AT in Algorithm 3. We train the VIT-B16 model for 110 epochs (L2) using a cosine learning rate schedule (L3). The proposed AAS attack is performed every λ epoch (L5). Firstly, the AAS attack is initialized using Gaussian noise (L6). Every λ epochs (L8) AAS attack is performed where the

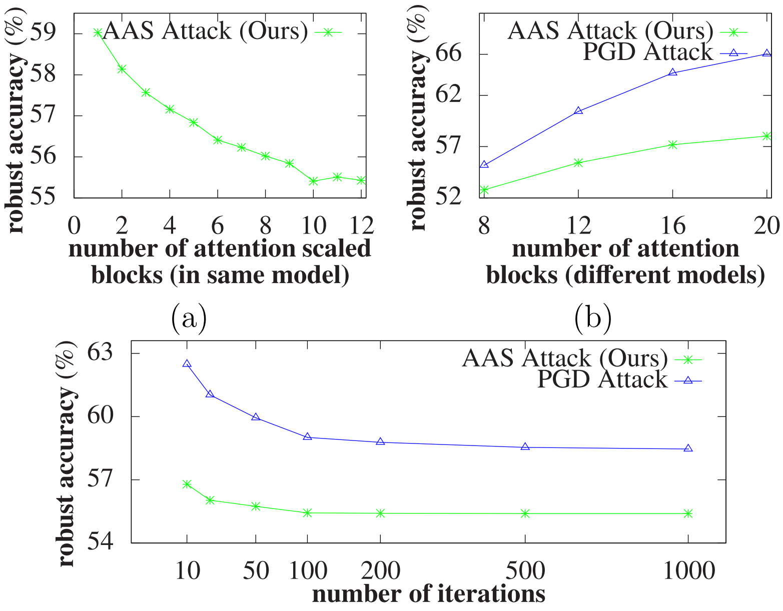
Table 4.3: **Comparison of AAS attack** with different attack methods. Bracket shows the scaling factor.

Data	Attack	Clean Acc. w/o AAS / Scale	Robust Acc. w/o AAS / Scale	Clean Acc. + Scale	Robust Acc. + Scale	Clean Acc. + AAS	Robust Acc. + AAS
CIFAR10	FGSM [30]	87.43	66.48	87.01 (0.01)	63.78	87.31	**61.04**
	PGD-20 [41]		63.14		60.64		**58.21**
	PGD-100[41]		61.10		59.43		**58.01**
	CW [11]		59.98		58.13		**57.73**
	DLR [18]		60.03		58.31		**57.94**
	GAMA [49]		59.78		58.16		**57.61**
CIFAR100	FGSM [30]	62.47	33.46	61.42 (0.001)	30.78	62.03	**28.03**
	PGD-20 [41]		30.78		28.31		**27.21**
	PGD-100 [41]		30.01		27.76		**27.02**
	CW [11]		29.03		27.08		**26.31**
	DLR [18]		29.21		26.95		**26.42**
	GAMA [49]		28.97		26.64		**26.08**
IN-100	FGSM [30]	68.03	32.06	67.36 (0.01)	29.47	67.84	**28.79**
	PGD-20 [41]		30.02		27.13		**26.41**
	PGD-100 [41]		29.75		27.03		**26.12**
	CW [11]		28.69		26.48		**25.81**
	DLR [18]		28.71		26.71		**25.90**
	GAMA [49]		28.07		26.15		**25.64**

LPIPS distance is maximized to perturb the pre-softmax scaling factors (L9) and later clamped to lie between $(10^{-r}, 1)$ (L10). For the remaining epochs, we perform the standard Trades [73] adversarial training where the KL-Divergence between the clean and the adversarial images is maximized to perturb the images (L12-L15). Finally, if the task was to perturb the pre-softmax output scaling factor, then the old scaling factors are reinitialized using the new perturbed ones (L17). Otherwise, standard Trades adversarial training occurs on the perturbed image (L19-20) where the cross-entropy loss on the clean images and KL Divergence between the clean and the adversarial images is minimized.

4.6 Experimental Results

In this section, we present the results of the proposed Adaptive Attention Scaling (AAS) attack and Adaptive Attention Scaling Adversarial Training (AAS-AT). The performance is evaluated on CIFAR10, CIFAR100 and ImageNet-100 datasets, where ImageNet-100 is a random 100 class subset of ImageNet-1K. In all the experiments, we use an ℓ_∞ norm threat model of perturbation bound 8/255. PGD-100, CW, DLR, GAMA use 100 iterations for generating the attack, whereas FGSM is a single-step attack. For training, we utilize a 10-step attack for all Adversarial Training methods and the training is done using the VIT-B16 model (unless specified) using additional



(a) (b)

(c)

Figure 4.4: **Ablation of AAS attack on CIFAR10. (a)** Effect of increasing the number of attention blocks used in AAS attack **(b)** Comparison between PGD and AAS attack on increasing the size of the model. Using a larger model with more attention blocks leads to larger gradient masking. **(c)** Comparison between PGD and AAS attack on increasing the number of attack iterations. Using a larger number of iterations also doesn't decrease the gap between the attack strengths.

synthetic data generated from Diffusion models (DDPM) [33] in the case of CIFAR10 and CIFAR100. For the evaluation of the attack, we use the VIT-B16 model trained on the standard PGD-AT [41] with AWP [67] for weight space smoothing. We use RTX-2080 and V100 gpus for all the experiments. Further training details of individual adversarial training methods, along with the reproducibility evaluations (reruns of AAS-AT), are present in the appendix.

4.6.1 Evaluation of the proposed AAS attack

The results of the proposed Adaptive Attention Scaling (AAS) attack on CIFAR10, CIFAR100 and ImageNet-100 datasets are presented in Table 4.3. The results are shown on incorporating AAS with different existing attack methods. In each of them w/o ASA/Scale represents the original accuracy achieved by the respective attack itself. Whereas + Scale represents the accuracy achieved on manually finding the best possible scaling factor for GAMA attack in the set $\{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The same scaling factor is used for all the attention blocks, and separate tuning for each attention block is not performed because it is $O(m^t)$ where m is the attention blocks and t is the size of the scaling factor set, which is computationally very expensive. Finally, + AAS represents the performance of the respective attack on combining with the proposed Adaptive Attention Scaling attack. Since on performing scaling (+ Scale) or the proposed attack (+ AAS), the function mapping of the model will change; therefore, clean accuracy will also change.

As observed in Table 4.3, the clean accuracy drops by upto 1% when finding the scaling factors manually. But on using AAS attack, this decrease is not more than 0.45%. On the other hand, on CIFAR10, the robust accuracy decreases by upto 5% in the case of FGSM and PGD-20 attacks. Even for stronger attacks like CW, DLR and GAMA, an improved attack strength of up to 2% is observed on CIFAR10. It is also observed that simply scaling the pre-softmax outputs can also help in improved attack strength by upto 1.6% in case of the strongest GAMA attack, but since the scaling factors found in this fashion might not be optimal, therefore finding them using gradient-based optimization as used in AAS attack helps in a further boost of around 0.51%. Even on CIFAR100, on combining the proposed AAS attack with GAMA, it shows 2.89% improved attack strength, whereas scaling shows 2.3% improvements. We also compare the performance of the proposed AAS attack on ImageNet-100 where we observe around 3.5% improvements over PGD-100 and 2.2% improved results over the GAMA attack. As can be seen from this analysis, by overcoming the floating point underflow errors, the proposed AAS attack gives consistent gains over the existing attacks.

• **Ablation experiments.** As shown in Figure 4.4 (a), on increasing the number of attention blocks in which the pre-softmax values are scaled using the proposed AAS attack, the robust accuracy on the CIFAR10 dataset falls continuously. Since floating point errors occur in each of the attention blocks, therefore when scaling is done for a larger number of attention blocks, the effect of gradient making is minimized, thus leading to stronger attacks. Further, as shown in Figure 4.4 (b), if the size of the model is increased by adding up more attention blocks, the drop in robust accuracy of the proposed AAS attack with respect to PGD-100 further increases. This demonstrates the effectiveness of overcoming gradient masking by using the proposed AAS attack. Finally, we present the effect of increasing the number of iterations of attack for PGD and AAS attacks in Figure 4.4 (c). As can be seen, AAS attack saturates earlier than PGD, and PGD is not able to close up the gap between the two attacks even on using 1000 iterations.

4.6.2 Evaluation of the proposed defense (AAS-AT)

As shown by Mo et al. [44] it is essential to use a pretrained initialization along with gradient clipping to enable stable and effective adversarial training of VITs. Therefore, we use ImageNet-1K initialization and gradient clipping in all our experiments. We utilize standard Pad-Crop along with Horizontal Flip as the augmentations. Training is done for 110 epochs with a max learning rate of 0.1, and a cosine learning rate schedule is used for all experiments except XCIT-S12 [22]. For XCIT-S12 and XCIT-S12 + Ours, we train for 300 epochs instead. Further, SGD, along with a momentum

Table 4.4: **Comparison of AAS-AT** with different adversarial training models.

	CIFAR10				CIFAR100			
Model	Clean	PGD-100	PGD-100 + AAS (Ours)	AA [18]	Clean	PGD-100	PGD-100 + AAS (Ours)	AA [18]
PGD-AT [41]	**86.14**	59.12	54.24	53.14	60.04	30.06	26.31	25.78
PGD-AT + Ours	85.32	58.12	57.78	**56.61**	**62.06**	29.03	28.34	**27.71**
Trades [73]	86.31	60.12	55.49	54.03	61.03	30.43	26.94	26.01
Trades + Ours	**87.46**	59.01	58.03	**57.34**	**63.06**	29.41	28.71	**28.06**
Trades + AWP [67]	86.21	60.48	56.84	56.03	62.78	31.86	27.94	27.03
Trades + AWP + Ours	**87.10**	59.78	58.41	**57.73**	**63.14**	31.76	30.43	**29.41**
MART [64]	**86.19**	59.13	54.76	54.12	62.41	32.61	27.12	26.47
MART + Ours	85.31	57.87	56.43	**55.78**	**62.84**	28.32	27.83	**27.01**
XCIT-S12 [22]	90.06	61.48	57.06	56.14	67.34	37.86	33.41	32.17
XCIT-S12 + Ours	**90.78**	59.94	57.84	**57.42**	**67.12**	35.35	33.97	**33.46**
Mo et al. [44]	86.43	60.03	56.12	55.03	61.76	31.3	27.84	27.01
Mo et al. [44] + Ours	**86.71**	58.46	57.44	**56.79**	**61.43**	30.86	29.16	**28.42**

of 0.9, is used as the optimizer in all the experiments.

The simplicity of the proposed AAS-AT allows it to combine effectively with any existing adversarial training method. The results of combining the proposed AAS-AT with different adversarial training methods on CIFAR10 and CIFAR100 are shown in Table 4.4. We use PGD-100, PGD-100 + AAS and AutoAttack [18] for evaluating the robustness of the defenses. As can be seen, AutoAttack remains the strongest white box attack. But the proposed PGD-100+AAS attack improves the attack strength by upto 4.5% over PGD-100. In the case of PGD-AT + AAS-AT, the difference between PGD-100 and PGD-100+AAS is significantly reduced to only 0.34%. This shows that since the scale of the softmax is inherently lowered on training using AAS-AT, even PGD-100 remains effective. This demonstrates that large scaling of pre-softmax outputs indeed leads to the generation of weaker attacks. Though PGD-100+AAS is weaker than AutoAttack [18], the difference between PGD-100 + AAS and AutoAttack is less than 1% in all cases. Further, PGD-100 + AAS is significantly cheaper in terms of compute as compared to AA. This demonstrates the effectiveness of the proposed AAS attack. As shown in Table 4.4, it can be observed that on CIFAR10 incorporating AAS-AT with standard adversarial training methods like PGD [41] and Trades can improve the performance by upto 3.47% on AA attack. Further on, incorporating AAS-AT with state-of-the-art adversarial training methods like Trades + AWP also gives an improved performance of upto 1.7%. On CIFAR100, we get even larger gains of upto 2.38% on combining with Trades + AWP. We also demonstrate that AAS-AT can improve the performance of existing adversarial training methods like [22, 44], which are specifically crafted for VITs. In the case of XCIT-S12 [22], AAS-AT improves the AA attack [18] performance by around 1.38%

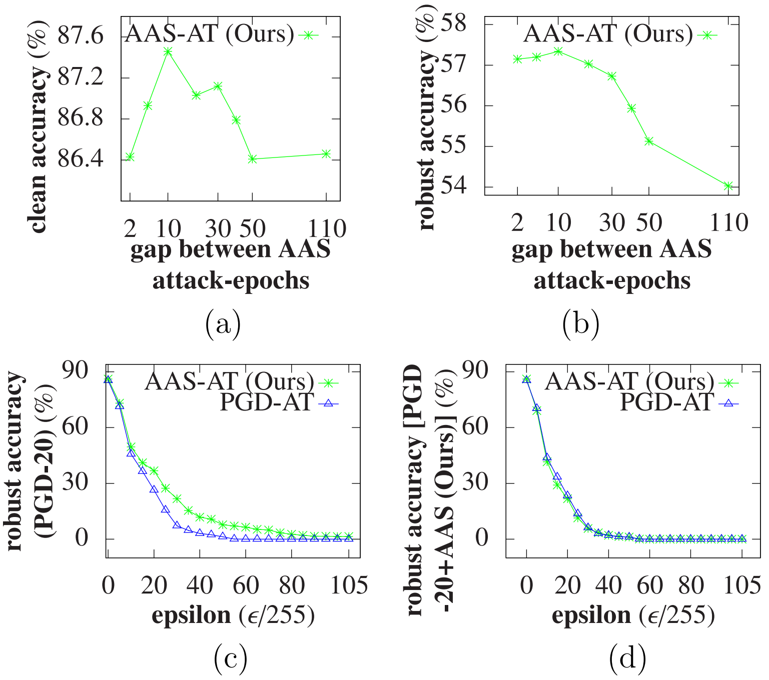


Figure 4.5: **Ablation of PGD-AT + AAS-AT on CIFAR10.** Variation of **(a)** clean and **(b)** PGD-100 (robust) accuracy with changing the gap between consecutive AAS attack epochs, respectively. **(c)** Lower robust accuracy (PGD-20) of AAS-AT and saturation to zero robustness of AAS-AT occurs at a much lower ϵ value as compared to PGD-AT. This indicates the absence of gradient masking in AAS-AT. **(d)** Using AAS attack on top of PGD-20 overcomes gradient masking, and the robust accuracy of PGD-AT decreases significantly as compared to **(c)**.

and 1.29% in case of CIFAR10 and CIFAR100 respectively. On combining AAS-AT with [44], it shows gains of 1.76% on CIFAR10 and 1.41% on CIFAR100.

• **Ablation experiments.** As shown in Figure 4.5 (a,b), the proposed defense PGD-AT + AAS-AT is stable to using AAS attack every $5 - 40$ epochs. Using AAS attack too frequently or using it only once/twice in the entire training leads to suboptimal performance. The effect of varying ϵ and performing PGD-20 attack during evaluation is shown in Figure 4.5 (c). Since the proposed AAS-AT does not have a large scale of pre-softmax outputs, PGD-20 attack is stronger for PGD-AT + AAS-AT (this is also evident from Table 4.4) as compared to the baseline PGD-AT. Since PGD-AT suffers from gradient masking, thus its accuracy does not reach 0% even on using an $\epsilon = 100/255$. But we get zero robustness on using $\epsilon = 65/255$. This shows AAS-AT does not suffer from gradient masking. Finally, in Figure 4.5 (d), we show that on using AAS attack along with PGD-20, the accuracy becomes zero for both PGD-AT as well as the proposed AAS-AT at ϵ close to 60/255. Thus the proposed AAS attack is able to overcome the gradient masking effect in PGD-AT model.

4.7 Additional Details on AAS attack

4.7.1 Details on Datasets

The performance of different defences is evaluated on CIFAR10, CIFAR100 and ImageNet-100 [23] datasets, comprising of 10, 100 and 100 classes, respectively. The resolution of images in the CIFAR10 and CIFAR100 datasets is 32x32, while it is 256X256 in the ImageNet-100 dataset. For all the experiments, VIT-B16 architecture is used. CIFAR100 is a more challenging dataset as compared to CIFAR10 because it has one-tenth the number of images in each class along with a larger number of classes. We use the ImageNet-100 dataset to show that the proposed attack is generalizable to larger-resolution images as well. For all the attacks and training, we consider an ℓ_∞ threat model with $\epsilon = 8/255$. For all our attack and adversarial training experiments, we use the codebase of Mo et al. [44] [1].

4.7.2 Details on Attacks and Evaluation

We train the VIT-B16 model using PGD-AT [41] along with AWP [67] to evaluate the existing and proposed AAS attacks. VIT-B16 is trained for 110 epochs using a cosine learning rate schedule with a maximum learning rate of 0.1. SGD with a momentum of 0.9 is used for training the model. We utilize additional data generated from DDPM [33] for all the experiments on CIFAR10 and CIFAR100 datasets.

We evaluate PGD-AT + AWP trained VIT-B16 model against several attacks like PGD [41], CW [11], AutoAttack [18] and GAMA [49] attack. While AutoAttack [18] is the strongest attack, it is computationally expensive. Amongst the single-attack methods, which are relatively cheaper in terms of compute, GAMA attack is the strongest one. Amongst the multistep attacks except for AutoAttack, we use 100 iterations for generating the attack.

4.8 Additional Details on AAS-AT

4.8.1 Details on Training and Baselines

We use the VIT-B16 model for all the experiments. VIT-B16 is trained for 110 epochs using a cosine learning rate schedule with a maximum learning rate of 0.1. SGD with a momentum of 0.9 is used for training the model. We utilize additional data generated from DDPM [33] for all the experiments on CIFAR10 and CIFAR100 datasets except for Debenedetti et al. [22] we train for 300 epochs. We use a pretrained ImageNet-1K initialization and also use gradient clipping in all the experiments. We utilize simple Pad-Crop-Horizontal Flip as the augmentations for training. We compare the performance of the proposed AAS-AT with the PGD-AT [41], Trades

[1] https://github.com/mo666666/When-Adversarial-Training-Meets-Vision-Transformers

[73], Trades-AWP[67], MART [64], Debenedetti et al. [22] and Mo et al. [44].

4.9 Other Details

4.9.1 Error Analysis

We perform multiple reruns of the proposed AAS-AT, and we observe small variations in the robust and standard accuracy across the reruns. We performed three reruns of Trades+AAS-AT and observed 87.23% as the mean accuracy and a standard deviation of 0.21% in clean accuracy. Against AutoAttack, we observed 57.61% as the mean adversarial accuracy and a standard deviation of 0.16%.

4.9.2 Limitations

In this work, though we propose a gradient-based optimization to get the scaling factors automatically, it is bound to give an approximate value. It is difficult to analyze how close this is with respect to the optimal scaling factors one can find by trying out all possible combinations of scaling factors. Further, on perturbing the scaling factors, we observe a drop in the clean accuracy. This is bound to happen since the model is not finetuned on these perturbed scaling factors. Though there is a drop in the clean accuracy, it is not significant.

4.9.3 Social Impact

By highlighting the reason for gradient masking in VITs, this work aims to improve the robustness of VITs and prevent the development of future defences, which might give a false sense of security because existing attacks are weak on VITs. We hope that this work will help in the development of more robust defences on VITs in the future.

4.9.4 Additonal Explanation on the choice of LPIPS

The following observations have been there in the literature and our experiments:

- O_1: Maximizing the LPIPS [76] distance gives perceptual gradients [1].
- O_2: Perceptual gradients imply adversarial robustness [27].
- O_3: Maximizing the LPIPS distance leads to enhanced robustness by overcoming gradient masking [Ours].

Let a relation R_1 be defined as $R_1 : a\mathcal{R}b$ where a : LPIPS distance and b : perceptual gradients [1], Similarly, let a relation R_2 be defined as $R_2 : b\mathcal{R}c$ where b : perceptual gradients and c: robustness [27], The observation O_3 defines a relation between LPIPS distance (a) and robustness (c). Therefore, our observation O_3 defines a transitive relation given by $R_3 : a\mathcal{R}c$, which can be inferred from R_1 and R_2

as: R_1 and $R_2 \implies R_3$. Thus, this justifies the choice of LPIPS metric to perturb the pre-softmax output scaling.

4.10 Conclusion

In this chapter, we demonstrate that the inherent design of attention blocks in VITs leads to floating point underflow errors, which causes weaker attack generation. To find the appropriate scaling factors for each attention block, we propose Adaptive Attention Scaling attack, which maximizes the LPIPS distance between the original and the perturbed model, where the perturbation is generated only on the pre-softmax output scaling factors. Since LPIPS distance is known as a good perceptual metric, maximizing it leads to perceptually aligned gradients, which is a characteristic of robust models [27]. We show that maximizing LPIPS distance indeed finds the appropriate scaling factors, thus overcoming gradient masking effect. We demonstrate that such an attack strategy can be integrated with any existing attack and leads to improved attack strength even on combining it with state-of-the-art single attacks, like GAMA attack. Further, we utilize this strategy in existing adversarial training methods and demonstrate improvements in robustness. Due to the simple design, the proposed method can be incorporated with any existing adversarial training method. Combining it with AT methods that are mainly designed for VITs also gives improved performance.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

In this thesis we try to analyze the issues with adversarial attacks on Graph Neural Networks (GNNs) and Vision Transformers (VITs) and motivated by our findings, we propose improved attacks on these class of deep neural networks (DNNs). In the first work (chapter), we highlight the need to rethink node injection attacks on GNNs. Firstly, we find that using cross-entropy loss is not the best choice and propose to use max-margin loss in the PGD attack formulation. Further, we observe that maximizing max-margin loss may lead to sub-optimal search within the constraint ball. Thus, in addition to max-margin loss maximization, we minimize cosine similarity between random initialization and attacked features of injected nodes to search the constraint space. Since transferability is important in the context of black box attacks, we also perform a weight perturbation in ℓ_2 norm before crafting the attack to get a model whose loss surface can generalize better. We show that the use of ℓ_∞ norm restricts the diversity of attack features since same magnitude of update is enforced in all feature dimensions. We propose to use ℓ_2 norm perturbation instead for gradient ascent. We demonstrate improved performance over graph robustness benchmark [77] models. We show that our method is generalizable to larger graphs, like Twitter and Amazon2M, and different classes of GNNs. It also improves black box attack strength in case of adversarially trained models and remains imperceptible making it difficult to detect it.

In the second work (chapter), we demonstrate that the inherent design of attention blocks in VITs leads to floating point underflow errors, which causes weaker attack generation. To find the appropriate scaling factors for each attention block, we propose Adaptive Attention Scaling attack, which maximizes the LPIPS distance between the original and the perturbed model, where the perturbation is generated only on the pre-softmax output scaling factors. Since LPIPS distance is known as a good perceptual metric, maximizing it leads to perceptually aligned gradients, which is a characteristic of robust models [27]. We show that maximizing LPIPS distance indeed finds the appropriate scaling factors, thus overcoming gradient masking effect. We demonstrate that such an attack strategy can be integrated with any existing attack and leads to improved attack strength even on combining it with state-of-the-art single attacks, like GAMA attack. Further, we utilize this strategy in existing adversarial training methods and demonstrate improvements in robustness. Due to the simple design, the proposed method can be incorporated with any existing adversarial training method. Combining it with AT methods that are mainly designed for VITs also gives improved performance. We hope that by providing a fundamental understanding of gradient masking in VITs and proposing appropriate insights to improve the attack strength of node injection attacks, this work will open new avenues of research in enhancing the robustness of VITs and GNNs even further.

5.2 Future Directions

The ℓ_p threat models are popular in literature because they are well-defined mathematically. It is difficult to define real-world robustness requirements mathematically. A hope is that achieving robustness to the union of ℓ_p norm-based threat models might suffice to achieve robustness to real-world attacks. But aiming to achieve robustness to the union of ℓ_p norm threat models leads to large degradation in the clean accuracy. Further, the real-world attacks are far from any ℓ_p threat model. Therefore, it is indeed necessary to design a threat model different from standard ℓ_p norm-based threat models, which can address the real-world robustness requirements. Further, it would be interesting to analyze how effective and efficient adversarial training methods can be designed on this threat model to achieve robustness to real-world attacks.

It is well known that adversarial training leads to a trade-off between clean and robust accuracy. Because of additional constraints in the training objective, the clean accuracy falls as compared to standard training. Trades [73] first demonstrated this fundamental tradeoff. Some works have demonstrated that perfect robustness can be achieved if the number of parameters in the model is increased to the scale of billions. Therefore while larger capacity models demonstrate improved robustness as well as clean accuracy, it is interesting to see if distilling the knowledge from the large models to the small models can help them in achieving improved robustness as well as clean accuracy. A general method for distilling the knowledge from a larger model to a smaller one in a student-teacher setting is using the KL Divergence loss and minimizing the distribution shift in the output space of the student and teacher models.

Bibliography

[1] S. Addepalli, S. Jain, G. Sriramanan, and V. B. Radhakrishnan. Scaling adversarial training to large perturbation bounds. In *The European Conference on Computer Vision (ECCV)*, 2022.

[2] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *The European Conference on Computer Vision (ECCV)*, 2020.

[3] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[4] A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *ArXiv*, abs/1802.00420:1–12, 2018.

[5] Y. Bai, J. Mei, A. L. Yuille, and C. Xie. Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, 34:26831–26843, 2021.

[6] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10231–10241, 2021.

[7] P. Boldi and S. Vigna. The webgraph framework i: Compression techniques. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, page 595–602, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 158113844X. doi: 10.1145/988672.988752. URL https://doi.org/10.1145/988672.988752.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot

learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S18Su--CW.

[10] N. Carlini and D. Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[11] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.

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

[13] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[14] H. Chang, Y. Rong, T. Xu, W. Huang, H. Zhang, P. Cui, W. Zhu, and J. Huang. A restricted black-box adversarial framework towards attacking graph embedding models. *Proc. of AAAI Conference on Artificial Intelligence*, 34(04):3389–3396, 2020.

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

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

[17] F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning (ICML)*, 2020.

[18] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proc. of Int. Conf. on Machine Learning*, pages 1–11, 2020.

[19] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[20] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[21] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song. Adversarial attack on graph structured data. In *International Conference on Machine Learning*, pages 1123–1132, 2018.

[22] E. Debenedetti, V. Sehwag, and P. Mittal. A light recipe to train robust vision transformers. *arXiv preprint arXiv:2209.07399*, 2022.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

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

[27] R. Ganz, B. Kawar, and M. Elad. Do perceptually aligned gradients imply adversarial robustness? *arXiv preprint arXiv:2207.11378*, 2022.

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

[29] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *CoRR*, abs/1412.6572:1–11, 2015.

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[31] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelović, T. A. Mann, and P. Kohli. On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models. *ArXiv*, pages 1–16, 2018.

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

[33] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. A. Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.

[34] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[35] D. Hitaj, G. Pagnotta, I. Masi, and L. V. Mancini. Evaluating the robustness of geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2103.01914*, 2021.

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

[37] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 591–600, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. doi: 10.1145/1772690.1772751. URL https://doi.org/10.1145/1772690.1772751.

[38] C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *International Conference on Learning Representations (ICLR)*, 2021.

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

[40] J. Ma, S. Ding, and Q. Mei. Towards More Practical Adversarial Attacks on Graph Neural Networks. *arXiv: Learning*, pages 1–11, 2020.

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

[42] K. Mahmood, R. Mahmood, and M. Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021.

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

[44] Y. Mo, D. Wu, Y. Wang, Y. Guo, and Y. Wang. When adversarial training meets vision transformers: Recipes from training to architecture. *arXiv preprint arXiv:2210.07540*, 2022.

[45] M. Naseer, K. Ranasinghe, S. Khan, F. S. Khan, and F. Porikli. On improving adversarial transferability of vision transformers. *arXiv preprint arXiv:2106.04169*, 2021.

[46] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[47] R. Shao, Z. Shi, J. Yi, P.-Y. Chen, and C.-J. Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[48] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

[49] G. Sriramanan, S. Addepalli, A. Baburaj, and R. Venkatesh Babu. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[50] Y. Sun, S. Wang, X. Tang, and T.-Y. Hsieh. Non-target-specific node injection attacks on graph neural networks: A hierarchical reinforcement learning approach. *ArXiv*, pages 1–15, 2020.

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

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

[53] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. of Int. Conf. on Learning Representations*, pages 1–10, 2013.

[54] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 990–998, 2008.

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

[56] S. Tao, Q. Cao, H. Shen, Y. Wu, L. Hou, and X. Cheng. Adversarial camouflage for node injection attack on graphs, 2022. URL https://arxiv.org/abs/2208.01819.

[57] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[58] F. Tramer, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

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

[63] J. Wang, M. Luo, F. Suya, J. Li, Z. Yang, and Q. Zheng. 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*, pages 1–20, 2020.

[64] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020.

[65] Z. Wei, J. Chen, M. Goldblum, Z. Wu, T. Goldstein, and Y.-G. Jiang. Towards transferable adversarial attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2668–2676, 2022.

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

[67] D. Wu, S.-T. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[68] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[69] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. *Proc. of Twenty-Eighth Int. Joint Conference on Artificial Intelligence*, pages 3961–3967, 2019.

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

[71] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. 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, 2014.

[72] Y. Yu and C.-Z. Xu. Efficient loss function by minimizing the detrimental effect of floating-point errors on gradient-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4056–4066, June 2023.

[73] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

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

[75] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=iAX0l6Cz8ub.

[76] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[78] D. Zhu, Z. Zhang, P. Cui, and W. Zhu. Robust Graph Convolutional Networks Against Adversarial Attacks. In *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pages 1399–1407, 2019.

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

[80] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial Attacks on Neural Networks for Graph Data. *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pages 1–10, 2018.

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

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

Publications from this thesis

[1] Samyak Jain & Tanima Dutta Rethinking node injection attacks on GNNs. In proceedings of AAAI Safe and Robust AI track, 2024.

[2] Samyak Jain & Tanima Dutta Understanding and improving adversarial robustness of vision transformers. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2024.*

Other Publications

[1] Samyak Jain, Sravanti Addepalli, Pawan Sahu, Priyam Dey & R.V. Radhakrishnan DART: Diversify-Aggregate-Repeat training improves generalization of neural networks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023.*

[2] Sravanti Addepalli, Samyak Jain, Gaurang Sriramanan & R.V. Radhakrishnan Scaling adversarial training to large perturbation bounds. *The European Conference on Computer Vision (ECCV), 2022.*

[3] Sravanti Addepalli, Samyak Jain & R.V. Radhakrishnan Efficient and effective augmentation strategy for adversarial training. *In Neural Information Processing Systems (NeurIPS), 2022.*