If You Want to Be Robust, Be Wary of Initialization

Sofiane Ennadir*
KTH
Stockholm, Sweden

Johannes F. Lutzeyer
LIX, Ecole Polytechnique
Paris, France

Michalis Vazirgiannis
KTH & Ecole Polytechnique
Stockholm, Sweden

El Houcine Bergou
UM6P
Benguerir, Morocco

Abstract

Graph Neural Networks (GNNs) have demonstrated remarkable performance across a spectrum of graph-related tasks, however concerns persist regarding their vulnerability to adversarial perturbations. While prevailing defense strategies focus primarily on pre-processing techniques and adaptive message-passing schemes, this study delves into an under-explored dimension: the impact of weight initialization and associated hyper-parameters, such as training epochs, on a model's robustness. We introduce a theoretical framework bridging the connection between initialization strategies and a network's resilience to adversarial perturbations. Our analysis reveals a direct relationship between initial weights, number of training epochs and the model's vulnerability, offering new insights into adversarial robustness beyond conventional defense mechanisms. While our primary focus is on GNNs, we extend our theoretical framework, providing a general upper-bound applicable to Deep Neural Networks. Extensive experiments, spanning diverse models and real-world datasets subjected to various adversarial attacks, validate our findings. We illustrate that selecting appropriate initialization not only ensures performance on clean datasets but also enhances model robustness against adversarial perturbations, with observed gaps of up to 50% compared to alternative initialization approaches.

1 Introduction

Neural networks have demonstrated remarkable prowess across various domains, ranging from computer vision [7] to natural language processing [28], proving their ability to model and extract complex insights from real-world datasets. Recently, Graph Neural Networks (GNNs) [20, 35, 29] have emerged as a powerful extension of neural networks specifically tailored to tackle graph-structured data. These models have led to rapid progress in solving tasks such as node and graph classification where their application have spanned from drug design [19], protein resistance analysis [23] to session-based recommendations [32]. Concurrently with their success, deep learning architectures have been shown to be unstable when subject to adversarial perturbations [14], resulting in unreliable predictions, consequently questioning these models' applicability in critical domains. While most adversarial robustness studies focus on the domain of computer vision, recent work [15] studying the robustness of GNNs has emerged. Given their rich nature, graphs allow different attack schemes, where the attacker can either choose to edit the graph structure (by adding/deleting edges) or edit the node/edge features. In parallel, recent studies have been devoted to studying approaches to defend against these attacks and enhance GNN's robustness, such as input pre-processing techniques [31], low-rank approximation [10], edge-pruning [37] or adapting the message-passing schemes [1].

*Corresponding Author: `ennadir@kth.se`

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

The majority of available defense studies focus on understanding the inner dynamics of GNNs to pinpoint and mitigate adversarial vulnerabilities. While analyzing the message-passing mechanism and implementing input pre-processing techniques remains a viable direction, comprehensive understanding necessitates exploration beyond traditional avenues. In this sense, investigating factors such as weight initialization strategies and the impact of other hyperparameters, notably those associated with optimization mechanisms, can offer new insights and perspectives on achieving GNN's global robustness. Hyperparameter choices and tuning play a critical role in striking a balance between learning the underlying signals in the data and preventing overfitting to ensure the model's generalization. Hence, existing studies on initialization mainly evolves around understanding its effect on the model's convergence, stability and performance [33, 22]. In contrast, the current work primarily focuses on examining the effect of initialization on a model's underlying adversarial robustness, representing to the best of our knowledge the first exploration of its kind. Our main objective is to provide a theoretical understanding of the link between weight initialization and other dynamics such as the number of training steps and the resulting model's robustness. With this perspective in mind, we start by formalizing robustness in the context of GNNs when subjected to structural and node feature-based adversarial attacks. Subsequently, we derive an upper bound that connects the model's robustness to the weight initialization strategies. Specifically, we illustrate that this bound depends on the initial weight norms and the number of training epochs. Finally, we validate our theoretical findings by demonstrating the effects of employing various initialization strategies on the model's robustness using benchmark adversarial attacks on real-world datasets. Note that while our analysis primarily focuses on the widely used Graph Convolutional Networks (GCNs) [20] and Graph Isomorphism Networks (GIN) [35], we highlight the versatility of our approach by providing a general upper bound applicable to any Deep Neural Networks in Section 5. This underlines the potential for extending our analysis to a wide range of architectures, showcasing its broad applicability in understanding and enhancing adversarial robustness in neural networks. We summarize our contributions as follows:

- We provide a theoretical analysis that links weight initialization strategies with adversarial robustness in GNNs. We specifically derive an upper bound connecting a model's robustness to weight initialization and the number of training epochs, demonstrating that the initialization strategy can significantly influence the network's adversarial robustness.
- We validate our theoretical findings by conducting extensive experiments across various models using different benchmark adversarial attacks on real-world datasets. These experiments demonstrate that certain weight initialization strategies can enhance the model's defense against adversarial attacks, without degrading its performance on clean datasets.
- While our primary focus is on GNNs, we extend our analysis to Deep Neural Networks, illustrating the broader applicability of our theoretical analysis and its corresponding insights.

2 Related Work

Graph Adversarial Attacks. Multiple studies focus on designing adversarial attacks capable of fooling a graph-based classifier [15, 34, 9]. The majority of these methods [41, 36] approach the adversarial aim as an optimization problem and employs different methods to solving it such as meta-learning [40]. Furthermore, Nettack [39] constrained the problem by preserving degree distribution and imposing constraints on feature co-occurrence to generate unnoticeable perturbations. Finally, reinforcement learning was proposed recently as a mean to generate graph adversarial attacks [6].

Graph Adversarial Defenses. Recent efforts have emerged to defend against the aforementioned adversarial attacks. In particular, methods such as low-rank matrix approximation coupled with graph anomaly detection [21] have been used. For example, GNN-Jaccard [31] proposed to pre-process the graph's adjacency matrix to detect potential manipulation of edges. Other methods such as edge pruning [37] and transfer learning [27] have been leveraged to limit the effect of poisoning attacks. Additionally, adaptations of the message-passing scheme, such as employing orthogonal weights [1] or introducing noise during training [8], have been shown to perform well in term of defense. Furthermore, there is a growing interest in exploring robustness certificates [41, 3] as a means of ensuring model robustness. For instance, [4] used randomized smoothing to provide a highly scalable model-agnostic certificate for graphs. Additionally, other robustness certificates for GCN-based graph classification under topological perturbations have been proposed [18].

Weight Initialization. The impact of weight initialization has been extensively studied both theoretically and empirically where the main line of study consists of understanding the interplay between initialization techniques and the implicit regularization they induce, thereby elucidating their influence on a model's generalization capabilities [33, 22]. For instance, it has been showcased that sampling initial weights from the orthogonal group can speed up convergence [17]. Similarly, alternative initialization approaches such as Glorot Initialization [12] and Kaiming Initialization [16] have been proposed in efforts to improve the model's performance.

Our work stands apart from existing research on adversarial robustness as it represents, to the best of our knowledge, the first attempt to theoretically investigate the impact of initialization on a model's underlying robustness. Moreover, our approach diverges fundamentally from existing literature on weight initialization as our focus lies in theoretically understanding the effect of initialization on a model's robustness rather than its implications for generalization or convergence.

3 Graph Adversarial Robustness

In this section, we start by introducing the notation and some fundamental concepts related to GNNs. We afterwards establish the problem setup together with the set of considered assumptions. We finally lay out a GNN's robustness formalization on which we will build our theoretical analysis.

3.1 Preliminaries

Let $G = (V, E)$ be a graph where V ($|V| = n$) is its set of vertices and E its set of edges. We denote $A \in \mathcal{A} \triangleq \{0,1\}^{n \times n}$ its adjacency matrix. The graph nodes are annotated with feature vectors $X \in \mathcal{X} \subseteq \mathbb{R}^{n \times d}$ (the i-th row of X corresponds to the feature of node i). We denote by $\mathcal{N}(i)$ the neighbors of node $i \in V$ and $\|\cdot\|_2$ the Euclidean (resp., spectral) norm for vectors (resp., matrices).

In this work, we consider the task of node classification. In this task, every node is assigned exactly one class from $\mathcal{C} = \{1, 2, \ldots, C\} \subset \mathcal{Y}$ and we consider $d_{\mathcal{Y}}$ as a distance within the output space $\mathcal{Y}$. The learning objective is to find a function f_W, parameterized by W, that assigns each node $i \in V$ a class $c \in \mathcal{C}$ while minimizing some classification loss (e. g., cross-entropy loss), denoted as $\mathcal{L}$.

GNNs. A GNN model consists of a series of neighborhood aggregation layers that use the graph structure and the node features from the previous layers to generate new nodes representations. Specifically, GNNs update node feature vectors by aggregating local neighborhood information. In the particular case of GCNs, this process is described by the following iterative propagation:

$$h^{(\ell)} = \phi^{(\ell)}(\widehat{A} h^{(\ell-1)} W^{(\ell)}), \tag{1}$$

with $W^{(\ell)} \in \mathbb{R}^{p \times q}$ being the weight matrix in the ℓ-th layer, q is the embedding dimension and $\phi^{(\ell)}$ is a non-linear activation function. Moreover, $\widehat{A} \in \mathbb{R}^{n \times n}$ denotes the normalized adjacency matrix $\widehat{A} = D^{-1/2} A D^{-1/2}$ where $D = \mathrm{diag}(|\mathcal{N}(1)|, |\mathcal{N}(2)|, \ldots, |\mathcal{N}(n)|)$ denotes the degree matrix.

Problem Setup. For our theoretical analysis, we assume that the model is based on 1-Lipschitz activation functions (which is a characteristic of commonly used activation functions such as TanH). Additionally, we consider the training loss function $\mathcal{L}$ to be L-smooth and that it is minimized using gradient descent. We denote by W_* the local optimum towards which gradient descent iterates converge. Specifically, for a learning rate $\eta < \frac{1}{L}$, the update at time step t for a layer i is:

$$W_{t+1}^{(i)} = W_t^{(i)} - \eta \nabla \mathcal{L}(W_t^{(i)}).$$

It is worth emphasizing that although we focus on the node classification task, which is prevalent and well-studied in the literature of adversarial robustness, our analysis is equally applicable to other tasks such as graph classification. Moreover, while our theoretical analysis predominantly centers around using the gradient descent as the optimizer, this choice doesn't limit the generality of our findings. One can employ a different optimizer and still yield the same insights and results by following a similar approach as the one outlined in this paper. Consequently, this specific setup should not be perceived as a limitation but rather as an analytical choice.

3.2 Adversarial Robustness for Graph Neural Networks

Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ be a GNN-classifier following the framework outlined in Section 3.1. An adversarial attacks consists of generating an alternative graph $(\tilde{A}, \tilde{X})$ that perturbs the original prediction $f(A, X)$ while not being far (semantically) from the original graph. Typically, this generated graph must adhere to a number of constraints related to its similarity to the original graph, defined by a perturbation budget ϵ controlling the number of edited edges or features. The set of these graphs is written as $B([A, X]; \epsilon) = \{(\tilde{A}, \tilde{X}) : \min_{P\in\Pi}(\|A - P\tilde{A}P^T\|_2 + \|X - P\tilde{X}\|_2) \leq \epsilon\}$, where Π represents the set of permutations of the adjacency matrix. While the previous formulation relies on the ℓ_2 norm, other norms may be used depending on the domain of application and the specific use case. Building on previous work [8], the adversarial risk of a GNN can be defined as the expected error of adjacent graphs within the considered graph's neighborhood defined by ϵ written as:

$$\mathcal{R}_\epsilon[f] = \mathop{\mathbb{E}}_{(A,X)\sim\mathcal{D}} \left[\sup_{(\tilde{A},\tilde{X})\in B([A,X];\epsilon)} d_{\mathcal{Y}}(f(\tilde{A}, \tilde{X}), f(A, X)) \right]. \quad (2)$$

In the current analysis, we focus on the ℓ_2 norm as our output distance $d_{\mathcal{Y}}$ (which can be substituted by any norm – giving the existence of norm's equivalence). We theoretically approach the introduced adversarial risk by deriving an upper-bound, which reflects the model's expected error under input perturbation. Intuitively, a smaller upper bound reflects a smaller adversarial risk which in turn suggests a robust behavior locally. In this perspective, Definition 1 draws the link between the considered risk quantity and a model's robustness.

Definition 1. (Adversarial Robustness). The graph-based function $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ is said to be $(\epsilon, \gamma) -$ robust if its adversarial risk is upper-bounded by γ, i. e., $\mathcal{R}_\epsilon[f] \leq \gamma$.

The current definition addresses adversarial risk from a worst-case scenario perspective, which is the most prevalent approach in the literature. This means we aim to identify the neighbor graph that maximizes the harm (i. e., causes the greatest deviation from the original prediction). By upper-bounding the risk associated with this "worst-case" graph, we inherently account for all other potential adversaries within the same neighborhood, as their risk will be less than or equal to that of the worst-case scenario. We note that the nuances between the "average" and "worst-case" approaches have been thoroughly examined and justified in previous research [24].

4 On the Effect of Initialization

We start by considering the Graph Convolutional Networks (GCNs) within the broader context of Message Passing Neural Networks for node classification. This study investigates how initialization and other hyper parameters impact the final model's robustness. In this context, we aim to establish a connection between the introduced adversarial risk (Equation 2) and the initial weight distribution and its evolution during training. Specifically, we seek to demonstrate that different choices in the initialization distribution and other relevant parameters lead to varying levels of model robustness, offering new insights into the potential trade-offs between initialization strategies and robustness. In this sense, we derive an upper-bound (denoted as γ in Definition 1) on the stability of a GCN-based classifier when the input graph's node features are subject to adversarial attacks.

Theorem 2. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ denote a graph-based function composed of T GCN layers, where the initial weight matrix of the i-th layer is denoted by $W_0^{(i)}$. For adversarial attacks only targeting node features of the input graph, with a budget ϵ, we have (in respect to Definition 1):*

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \left\| W_0^{(i)} \right\| + 2^{t+1} \left\| W_*^{(i)} \right\| \right) \left(\sum_{u\in\mathcal{V}} \hat{w_u} \right)$$

with t being the number of training epochs and $\hat{w_u}$ denoting the sum of normalized walks of length $(T - 1)$ starting from node u.

Theorem's proof is provided in Section A of the Appendix. Theorem 2 provides a formal connection between the robustness of a GCN-based classifier and its initial weights, offering valuable insights into their effects. From a first perspective, the derived upper-bound depends on the initial weight's norm. Specifically, a lower norm corresponds to a smaller upper-bound, indicative of a more robust model. However, while setting all initial weights to zero theoretically yields the smallest upper-bound and consequently the optimum robustness, this direction can detrimentally affect the model's performance on the learning task. Empirical evidence suggests that initializing weights to zero (or a constant) often leads to poor learning outcomes, as it constrains weight behavior during propagation, limiting subsequent back-propagation operations and resulting in convergence to unsatisfactory local minima (e. g., see page 301 in [13]). From a second perspective, it appears that a higher number of training epochs leads to the looseness of the upper-bound, resulting in increased adversarial vulnerability. This latter observation provides proofs and highlights on the existence of the usually discussed trade-off between clean and attacked accuracy. Achieving a balance between increasing the number of epochs to achieve satisfactory clean accuracy and limiting them to attain a robust model is hence essential. While theoretically challenging to identify this equilibrium point, our experimental results demonstrate its existence. We note that the dependence of γ on t can be sharpened by having $(1+\eta L)^t$ instead of 2^t. With small η (which is the case usually in practice), $(1+\eta L)^t \approx 1 + t\eta L$ resulting in a bound which depends linearly in t. The same remark applies for the remaining bounds derived in the paper. These insights, in the case of node-feature-based adversarial attacks, also extends to structural perturbations where Theorem 3 provides the exact bound for this case.

Theorem 3. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ denote a graph-based function composed of T GCN layers, where the initial weight matrix of the i-th layer is denoted by $W_0^{(i)}$. Let f be the number of used training epochs. When f is subject to structural attacks, with a budget ϵ, we have (in respect to Definition 1):*

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \left\| W_0^{(i)} \right\| + 2^{t+1} \left\| W_*^{(i)} \right\| \right) \|X\| \left(1 + T \prod_{i=1}^{T} \left(2^t \left\| W_0^{(i)} \right\| + 2^{t+1} \left\| W_*^{(i)} \right\| \right) \right)$$

The computed upper-bound suggests that the effect of initialization is more important in the case of structural perturbations. This emphasis is resulting from the distinct dynamics within the message passing mechanism, where the influence of the adjacency matrix and node features varies during each propagation step. Precisely, for structural perturbations, the effect of the attack is considered at each propagation step through the perturbed adjacency matrix (in the aggregation step). Moreover, the impact is also amplified by the affected residual layers from previous iterations, resulting in a more significant attack result. This is different in the case of node-feature based adversarial attacks, since the node features are only taken into account in the first propagation. Overall, the main takeaway of the provided analysis in Theorem 2 and 3 is that "approximately-free" robustness enhancements can be derived from choosing the right initial weight's distribution and the right number of training epochs. We illustrate this specific point by analyzing the effect of the initial distributions choices on the model's robustness. Specifically, we consider the case of the Gaussian distribution, where Lemma 4 studies how the parameters of this distribution – namely, the mean and variance – exert an influence on the expected (in respect to the initial distribution) value of the adversarial risk.

Lemma 4. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ denote a graph-based function composed of T GCN layers for which the initial weight are drawn from the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. When subject to node features based adversarial attacks, we have the following:*

$$\mathbb{E}_{W_0 \sim \mathcal{N}(\mu,\Sigma)}[\mathcal{R}_\epsilon[f]] \leq \epsilon \prod_{i=1}^{T} \left(2^t \sqrt{\mu^2 + tr(\Sigma)} + 2^{t+1} \left\| W_*^{(i)} \right\| \right) \left(\sum_{u \in \mathcal{V}} \hat{w_u} \right)$$

Proof is provided in Section C.Given that a tighter upper bound inherently results in a higher level of robustness, the results derived in Lemma 4 illustrate the clear effect of initialization in the case of the Gaussian distribution. The derived bound shows that increasing the distribution parameters, both the mean and variance values, leads to a decrease in the victim model's underlying robustness. While one might intuitively aim to set these parameters as low as possible to achieve optimal robustness, doing so could potentially compromise the model's performance on clean datasets. Therefore, as previously mentioned, striking the right balance between clean accuracy and adversarial robustness is crucial.

Extending the results to the GIN. The same previously applied analysis for the GCN-based models can be extended to take into account GIN-based classifiers. We consider the same set of assumptions and the same problem setup considered during the previously studied GCN case. We additionally assume that the input node feature space to be bounded, i. e., $\|X\| \leq B$. We note that this bound is a realistic assumption and that the value B can be easily computed for any real-world dataset.

Theorem 5. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ denote a graph-based function composed of T GIN layers, where the initial weight matrix of the i-th layer is denoted by $W_0^{(i)}$. For adversarial attacks only targeting node features of the input graph, with a budget ϵ, we have:*

$$\gamma = \prod_{l=1}^{T} \left(2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\|\right) \left[BT \max_{u \in \mathcal{V}} deg(u) + \epsilon\right]$$

with t being the number of training epochs and $deg(u)$ is the degree of node u.

Proof of the theorem is provided in the appendix (Section D). Theorem 5 establishes an upper bound on the robustness of a GIN-based classifier against adversarial attacks targeting node features. We observe analogous insights, to the ones derived for a GCN-based classifier, regarding the influence of the initialization distribution and number of training epoch on the model's underlying robustness.

5 Generalization to Other Models

While our primary research focus lies within the domain of graph representation learning, a sub-field of the broader landscape of Deep Learning models, the fundamental principles of our theoretical analysis hold applicability across various model architectures. Notably, and to our knowledge, the absence of a comparable study in current adversarial literature motivates our endeavor to bridge this gap. In this section, we aim to fill this void by presenting a comprehensive analytical framework that provide the connection between weight initialization and the robustness of neural networks.

Let $x \in \mathbb{R}^{n_0}$ denote an input vector where n_0 is the input dimension. Let $W^{(l)} \in \mathbb{R}^{n_{l-1}, n_l}$ be the weight matrix and $b_l \in \mathbb{R}^{n_l}$ the bias of the l^{th} layer with n_l being its dimensionality. We focus on the general family of neural networks for which the computation during layer l, using an activation function $\phi^{(l)}$, can be written as :

$$h^{(l)} = \phi^{(l)}(W^{(l)} h^{(l-1)} + b^{(l)}).$$

We consider the same set of assumptions (stated in Section 3.1) as the one from previous section. We consider the ℓ_2 norm as our input and output distances within the metric space $\mathbb{R}^{n_0}$ and we consider an input attack budget $epsilon$. The introduced adversarial risk in Equation 2 can be easily extended and tailored to the family of considered neural networks discussed in this section. Further clarification on this extension is provided in the Appendix (Section G.1). From this standpoint, by adapting the Definition 1, analogous effects of the weight initialization, provided in Theorem 6, can be observed.

Theorem 6. *Let $f : \mathcal{X} \subseteq \mathbf{R}^{in} \rightarrow \mathcal{Y} \subseteq \mathbf{R}^{out}$ be a T-layers neural network with $W_0^{(i)}$ denoting the initial weight matrix of the i-th layer. When subject to adversarial attacks, f is $(\epsilon, \gamma) -$ robust with:*

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\|\right)$$

The proof of Theorem 6 can be found in Section E of the Appendix. Similar to previous findings, the upper bound relies on key elements of the initialization process, specifically the initial weight norm and the number of training epochs. These results validate and extend the established link between initialization and a model's robustness in neural networks, highlighting the importance of selecting appropriate parameters. From the derived upper bound, which is also applicable to GCN and GIN cases, we observe that the number of training epochs exerts an effect on the bound. Specifically, while increasing the number of epochs can improve the model's performance on a clean dataset, it simultaneously leads to a deterioration in the model's adversarial robustness. Ideally, adversarial defense strategies aim to avoid this trade-off between clean and attacked accuracy, striving for robust models that do not compromise the initial performance. In this context, considering the strong-convexity of the loss function $\mathcal{L}$, in addition to the previously made assumptions, we observe that the effect of the number of training epochs becomes less pronounced. Lemma 7 specifically provides the computed bound under these assumptions.

Lemma 7. *Let $f : \mathcal{X} \subseteq \mathbf{R}^{in} \rightarrow \mathcal{Y} \subseteq \mathbf{R}^{out}$ be a T-layers neural network trained with a μ-strongly convex and L-smooth loss function. Let $W_0^{(i)}$ denote the initial weight matrix of the i-th layer. When* *subject to adversarial attacks, with a budget ϵ, we have that f is $(\epsilon, \gamma) -$ robust with:*

$$\gamma = \epsilon \prod_{i=1}^{T} \left((1 - \mu/L)^t \| W_0^{(i)} \| + 2 \| W_*^{(i)} \| \right)$$

The proof of the Lemma is provided in Section F of the Appendix. Since $\mu \leq L$, increasing the number of training epochs results in the diminishing influence of the initialization weights. In this scenario, the bound depends solely on the final weights, a phenomenon previously explored in works such as Parseval networks [5] for neural networks and GCORN [1] for GNNs. This observation highlights the necessity of convexity in the loss function when training a neural network, as it plays a crucial role in enhancing the model's robustness, beyond the traditional considerations of classical training optimization perspectives.

6 Experimental Results

This section aims to empirically validate our theoretical findings using real-world benchmark datasets. We start by laying out the used experimental, then we study the impact of various initialization strategies on a GCN's robustness. Next, we analyze the influence of training epochs on adversarial robustness. Finally, we extend our experimentation to considered family of DNNs in Section 5.

6.1 Experimental Setting

Experimental Setup. Consistent with our theoretical analysis, this section focuses on the node classification task. We leverage the citation networks Cora and CiteSeer [26], with additional results on other datasets provided in the Appendix G. To mitigate the impact of randomness during training, each experiment was repeated 10 times, using the train/validation/test splits provided with the datasets. A 2-layers GCN classifier with identical hyperparameters and activation functions was employed across all the experiments. The models were trained using the cross-entropy loss function, and consistent values for the number of epochs and learning rate were maintained across all analysis. Further implementation details can be found in Appendix H and the code implementation to replicate our experiments is provided in the supplementary material.

Adversarial Attacks. We consider two main gradient-based structural adversarial attacks: **(i)** 'Mettack' (with the 'Meta-Self' training strategy) [40] that formulates the problem as a bi-level problem solved using meta-gradients **(ii)** and the Proximal Gradient Descent (PGD) [34] which consists of iteratively adding small crafted perturbations using the gradient of the classifier's loss. We additionally provide results for the 'Dice' attack [40] in Appendix G. For our experiments, we considered perturbation rates ranging from 10% (i. e., $0.1E$) to 40% (i. e., $0.4E$).

Evaluation Metrics. We report the experimental findings in terms of the 'Attacked Accuracy', which is the model's test accuracy when subject to the attacks. Additionally, given that initialization have an impact on the model's generalization and performance, solely reporting the attacked accuracy fails in some specific cases to provide a comprehensive perspective. Thus, we adopt for some experiments the "Success Rate" metric, also commonly employed in adversarial literature, which encompasses the number of successfully attacked nodes while taking into account the model's initial clean accuracy.

6.2 Effect Of Training Epochs

The theoretical analysis presented in Section 4 established a connection between the number of training epochs and the model's resultant robustness. The derived bound suggests that increasing the number of epochs results in the model becoming more vulnerable to adversarial attacks. The objective of this experimental section is to empirically validate this assertion using real-world datasets. To this end, at each training epoch, we assess the model's performance on the test set, considering both its clean accuracy and its accuracy under adversarial attacks.

Figure 1 illustrates the results of this analysis. The initial two subplots (a,b) displays the findings on the Cora dataset, while the subsequent (c,d) subplots presents results from the CiteSeer dataset. For each dataset, the first plot showcases the clean and attacked accuracy, while the second plot shows the Success Rate (the discrepancy between the clean and attacked accuracy for each budget). The

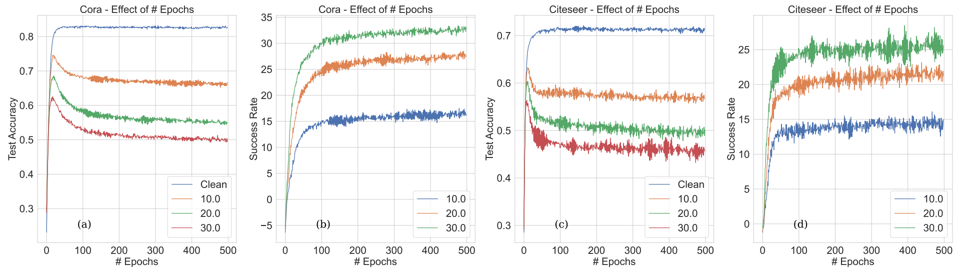


Figure 1: Effect of training epochs on the model's robustness on Cora (a,b) and CiteSeer (c,d).

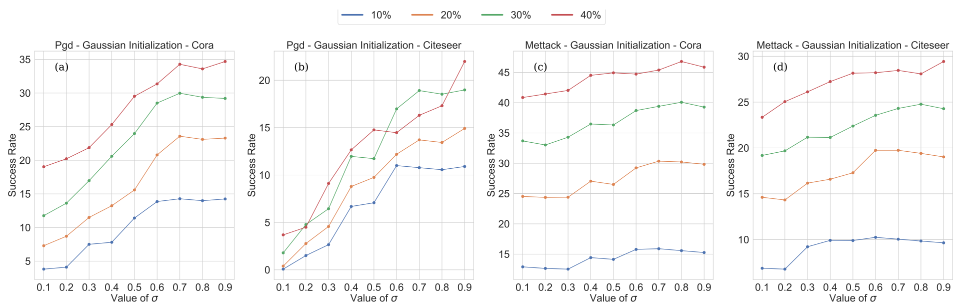


Figure 2: Effect of the variance parameter on the model's robustness in the case of Gaussian Initialization using on PGD [on Cora (a) and Citeseer (b)] and Mettack [on Cora (a) and Citeseer (b)].

experimental results demonstrate the existence of the previously discussed trade-off between clean and robust accuracies. Specifically, as anticipated, the clean accuracy exhibits a continual increase until reaching a plateau, corresponding to the convergence of the loss function to a minimum. Conversely, the attacked accuracy demonstrates a rising trend until reaching an inflection point, beyond which it begins to decline. These findings confirms the observations from the derived upper-bound, indicating that a higher number of epochs leads to increased vulnerability in the model. Ideally, users would aim to stop training at the inflection point, where the attacked accuracy is maximized while the clean accuracy remains proximal to its convergence point.

6.3 Effect Of Initial Weight Distribution

We aim to validate the impact of the initial weight norms on the model's adversarial robustness. As previously discussed in Section 4, a larger weight norm leads to the relaxation of the upper-bound, potentially resulting in the model being more susceptible to adversarial attacks.

In this perspective, we start by investigating the effect of sampling from a Gaussian distribution, as outlined in Lemma 4. We hence consider this latter by setting the mean value μ to a constant, and analyzing the impact of the variance parameter σ. Intuitively, based on the upper-bound analysis, a higher variance value is anticipated to result in reduced model robustness. Figure 2 illustrates the resultant Success Rate across various variance values for both the "PGD" and "Mettack" methods, applied to the Cora and Citeseer datasets. The findings unequivocally validate the theoretical insights, demonstrating a direct correlation between increasing the variance (σ) and a higher Success Rates, indicating heightened vulnerability and reduced robustness of the model. Moreover, the impact of initialization becomes more pronounced when considering larger attack budgets, as outlined in the computed upper-bound. Notably, for certain budgets (e.g., 30% and 40%), the observed gap ranges between 5% and 15%, underscoring the initial weights significant implications on the robustness.

Within the same context, we explore alternative initialization strategies, focusing on two primary cases. First, we investigate sampling initial weights from a uniform distribution $\mathcal{U}(-\beta, \beta)$, where β can be seen as a scaling parameter for weight norms. Second, we consider employing a scaled orthogonal weight initialization strategy. While this our aim can be approached by sampling weights from a scaled random Gaussian distribution, we adopt the orthogonal initialization strategy proposed in prior work [25], which we further rescale by a factor β to examine the impact on weight norms.

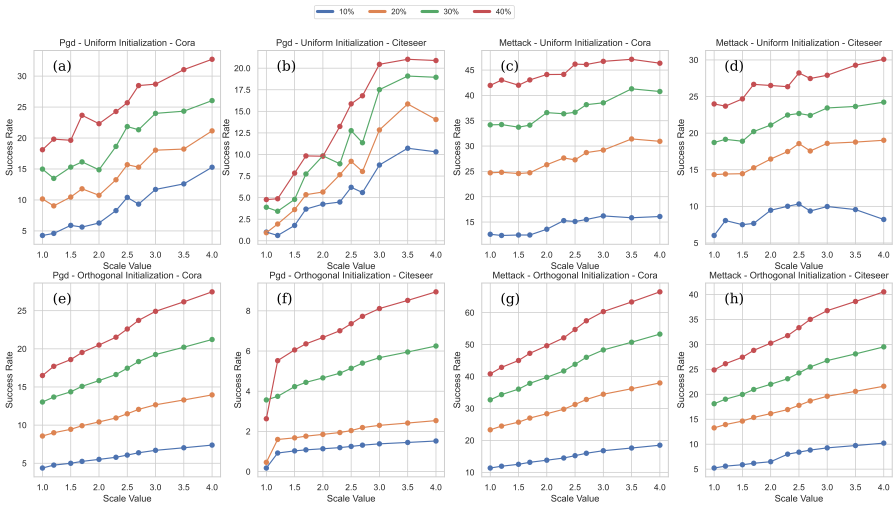


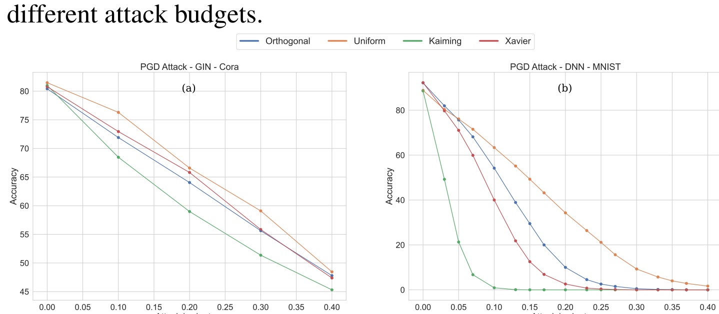
Figure 3: Effect of the scaling parameter β on the model's robustness in the case of Uniform (a-d) and Orthogonal (e-h) Initialization when subject to PGD and Mettack using Cora and CiteSeer.

In both cases, higher scaling parameter values of β are anticipated to theoretically yield higher upper-bounds and consequently render the model more vulnerable, as indicated by our computed bounds. We conduct numerical computations on both the Cora and Citeseer datasets to assess the resulting adversarial robustness of a GCN across various β values, as provided in Figure 3. The experimental results are exactly aligned with our theoretical findings showcasing the effect of the weight norm in the adversarial robustness. To summarize, while traditionally overlooked in prior studies on adversarial robustness, our experimentation underscores the critical importance of selecting appropriate initialization distributions and strategies for enhancing model robustness.

6.4 Experimental Generalization

We extend our experimentation to empirically validate the theoretical generalizations provided in both Section 4 for the GINs and Section 5 for a DNNs. To this end, we consider these two models with various initialization schemes, including the previously used Orthogonal [25] and Uniform initialization in addition to the Kaiming [16] and Xavier Initialization [12]. Our analysis primarily focuses on the PGD adversarial attack, using identical attack budgets as in the previous sections. Figure 4 presents the results on the GIN (a) using the Cora dataset and (b) on the DNN using the MNIST dataset. Notably, we observe that the different initialization methods yield similar clean accuracy ($\epsilon = 0$), yet as the attack budget increases, the discrepancy in attacked accuracy between them also grows. For instance, in the case of DNNs, the accuracy gap between the best and worst initialization methods for $\epsilon = 0.1$ ranges around 60%, proving our main assumption related to the impact of initialization on the model's robustness.

Figure 4: Effect of initialization on the GIN (a) and DNN (b) for different attack budgets.



7 Conclusion & Limitations

The current study shows that the dynamics of learning in GNNs and DNNs have an important effect on the model's final robustness. Specifically, we theoretically showed that the model's robustness is connected to the weight initialization and the number of training epochs. We empirically validate our findings, where we can see that choosing the right initialization can yield huge "almost-free" robustness improvement. We additionally showed the existence of a trade-off between choosing the right number of epochs to have the best clean accuracy and the most robust model. While the current work didn't propose an alternative or a solution, it has introduced a new perspective, which in our knowledge, was absent from the adversarial literature, opening the door to new research direction either by proposing new initialization scheme to improve robustness while guaranteeing a good generalization or new gradient-based weight updates to enforce the robustness of the model.

References

[1] Yassine Abbahaddou, Sofiane Ennadir, Johannes F. Lutzeyer, Michalis Vazirgiannis, and Henrik Boström. Bounding the expected robustness of graph neural networks subject to node feature attacks. In *The Twelfth International Conference on Learning Representations*, 2024.

[2] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[3] Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations, 2019.

[4] Aleksandar Bojchevski, Johannes Klicpera, and Stephan Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more, 2020.

[5] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pages 854–863. PMLR, 2017.

[6] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial Attack on Graph Structured Data. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1115–1124, 2018.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Sofiane Ennadir, Yassine Abbahaddou, Johannes F. Lutzeyer, Michalis Vazirgiannis, and Henrik Boström. A simple and yet fairly effective defense for graph neural networks, 2024.

[9] Sofiane Ennadir, Amr Alkhatib, Giannis Nikolentzos, Michalis Vazirgiannis, and Henrik Boström. Unboundattack: Generating unbounded adversarial attacks to graph neural networks. In *International Conference on Complex Networks and Their Applications*, pages 100–111. Springer, 2023.

[10] Negin Entezari, Saba A Al-Sayouri, Amirali Darvishzadeh, and Evangelos E Papalexakis. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th international conference on web search and data mining*, pages 169–177, 2020.

[11] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[15] Stephan Günnemann. Graph neural networks: Adversarial robustness. In *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 149–176. Springer, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[17] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *International Conference on Learning Representations*, 2020.

[18] Hongwei Jin, Zhan Shi, Venkata Jaya Shankar Ashish Peruri, and Xinhua Zhang. Certified robustness of graph convolution networks for graph classification under topological attacks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8463–8474. Curran Associates, Inc., 2020.

[19] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, 2016.

[20] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[21] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[22] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1924–1932. PMLR, 2018.

[23] Aymen Qabel, Sofiane Ennadir, Giannis Nikolentzos, Johannes F. Lutzeyer, Michail Chatzianastasis, Henrik Boström, and Michalis Vazirgiannis. Advancing antibiotic resistance classification with deep learning using protein sequence and structure. *bioRxiv*, 2023.

[24] Leslie Rice, Anna Bair, Huan Zhang, and J Zico Kolter. Robustness between the worst and average case. *Advances in Neural Information Processing Systems*, 34:27840–27851, 2021.

[25] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014.

[26] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[27] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM, jan 2020.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

[30] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

[31] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples for graph data: Deep insights into attack and defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4816–4823. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[32] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 346–353, 2019.

[33] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pages 5393–5402. PMLR, 2018.

[34] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. 2019.

[35] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations*, 2019.

[36] Haoxi Zhan and Xiaobing Pei. Black-box Gradient Attack on Graph Neural Networks: Deeper Insights in Graph-based Attack and Defense. *arXiv preprint arXiv:2104.15061*, 2021.

[37] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks, 2020.

[38] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1399–1407, 2019.

[39] Daniel Zügner, Amir Akbarnejad, and Stephan 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.

[40] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *7th International Conference on Learning Representations*, 2019.

[41] Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, jul 2019.

Supplementary Material: If You Want to Be Robust, Be Wary of Initialization

A Proof Of Theorem 2

Theorem. *Let $f : (\mathcal{A}, \mathcal{X}) \to \mathcal{Y}$ denote a graph-based function composed of T GCN layers, where the initial weight matrix of the i-th layer is denoted by $W_0^{(i)}$. For adversarial attacks only targeting node features of the input graph, with a budget ϵ, we have (in respect to Definition 1):*

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1} \left\|W_*^{(i)}\right\|\right) \left(\sum_{u \in \mathcal{V}} \hat{w_u}\right)$$

with t being the number of training epochs and $\hat{w}_u$ denoting the sum of normalized walks of length $(T-1)$ starting from node u.

Proof. Let's consider a graph-function f that is based on T GCN-layers. The gradient descent update at epoch t for a layer i is written as:

$$W_{t+1}^{(i)} = W_t^{(i)} - \eta \nabla \mathcal{L}(W_t^{(i)})$$

Since we consider that our loss function $\mathcal{L}$ to be L-smooth, we have the following result:

$$\|\nabla \mathcal{L}(W_t^{(i)})\| \leq L \|W_t^{(i)} - W_*^{(i)}\|$$

Consequently, after t training epochs, we can write:

$$\begin{aligned}
\|W_t^{(i)}\| &= \|W_{t-1}^{(i)} - \eta \nabla \mathcal{L}(W_{t-1}^{(i)})\| \\
&\leq \|W_{t-1}^{(i)}\| + \eta L \|W_{t-1}^{(i)} - W_*^{(i)}\| \\
&\leq (1 + \eta L) \|W_{t-1}^{(i)}\| + \eta L \|W_*^{(i)}\|
\end{aligned}$$

In addition, we have that $\eta \leq \frac{1}{L}$. Hence, by recursion, we find that:

$$\|W_t^{(i)}\| \leq (1 + \eta L)^t \|W_0^{(i)}\| + \sum_{h=0}^{t} 2^h \|W_*^{(i)}\| \tag{3}$$

$$\leq (1 + \eta L)^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\| \tag{4}$$

Giving that we are considering feature-based adversarial attacks, let X denote the original node features and X' denote the perturbed adversarial features. With an attack budget ϵ, from the work [1], we have the following result:

$$\forall [A, X'] \in B([A, X], \epsilon), \|f(A, X) - f(A, X')\| \leq \prod_{i=1}^{T} \|W_t^{(i)}\| \epsilon (\sum_{u \in \mathcal{V}} \hat{w}_u). \tag{5}$$

with $\hat{w}_u$ denoting the sum of normalized walks of length $(T-1)$ starting from node u. Consequently:

$$\sup_{[A, X'] \in B([A, X], \epsilon)} \|f(A, X) - f(A, X')\| \leq \prod_{i=1}^{T} \|W_t^{(i)}\| \epsilon (\sum_{u \in \mathcal{V}} \hat{w}_u). \tag{6}$$

From Result 3 and 6, we conclude that:

$$\sup_{[A,X']\in B([A,X],\epsilon)} \|f(A,X)-f(A,X')\| \leq \epsilon \prod_{i=1}^{T} \left[2^t\|W_0^{(i)}\| + 2^{t+1}\|W_*^{(i)}\|\right] (\sum_{u\in\mathcal{V}} \hat{w_u})$$

We conclude that f is (ϵ;γ)-robust with:

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1}\left\|W_*^{(i)}\right\|\right)\left(\sum_{u\in\mathcal{V}} \hat{w_u}\right)$$

□

B Proof Of Theorem 3

Theorem. *Let $f : (\mathcal{A},\mathcal{X}) \to \mathcal{Y}$ denote a graph-based function composed of T GCN layers, where the initial weight matrix of the i-th layer is denoted by $W_0^{(i)}$. Let f be the number of used training epochs. When f is subject to structural attacks, with a budget ϵ, we have (in respect to Definition 1):*

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1}\left\|W_*^{(i)}\right\|\right)\|X\|\left(1 + T\prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1}\left\|W_*^{(i)}\right\|\right)\right)$$

Proof. Similar to the previous proof, let's consider a graph-function f that is based on T GCN-layers and trained using gradient descent for t epochs. We have the following result from Equation 3:

$$\|W_t^{(i)}\| \leq 2^t\|W_0^{(i)}\| + 2^{t+1}\|W_*^{(i)}\| \tag{7}$$

For this proof, we are considering the model f to be subject to structural perturbations. In this perspective, let $\tilde{A}$ denote the input non-attacked adjacency and $\tilde{A}'$ denote the attacked/perturbed adjacency, with h' denoting its corresponding hidden representation. From the work [1], we have:

$$\forall [A',X] \in B([A,X],\epsilon), \|f(\tilde{A},X) - f(\tilde{A}',X)\| \leq \prod_{i=1}^{T}\|W^{(i)}\|\|X\|\epsilon(1 + T\prod_{i=1}^{T}\|W^{(i)}\|)$$

By combining the two previous results, we get that following inequality and hence the desired result:

$$\sup_{[A',X]\in B([A,X],\epsilon)} \|f(\tilde{A},X)-f(\tilde{A}',X)\| \leq \epsilon \prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1}\left\|W_*^{(i)}\right\|\right)\|X\| \left(1 + T\prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1}\left\|W_*^{(i)}\right\|\right)\right).$$

□

C Proof Of Lemma 4

Lemma. *Let $f : (\mathcal{A},\mathcal{X}) \to \mathcal{Y}$ denote a graph-based function composed of T GCN layers for which the initial weight are drawn from the Gaussian distribution $\mathcal{N}(\mu,\Sigma)$. When subject to node features based adversarial attacks, we have the following:*

$$\mathbb{E}_{W_0\sim\mathcal{N}(\mu,\Sigma)}[\mathcal{R}_\epsilon[f]] \leq \epsilon \prod_{i=1}^{T} \left(2^t\sqrt{\mu^2 + tr(\Sigma)} + 2^{t+1}\left\|W_*^{(i)}\right\|\right)\left(\sum_{u\in\mathcal{V}} \hat{w_u}\right)$$

Proof. Let's consider f to be a graph classifier based on T-GCN layers for which the initial weight are drawn from the Gaussian distribution. Specifically, $\forall i \leq L, W_0^{(i)} \sim \mathcal{N}(\mu,\Sigma)$. We have that:

$$\mathbb{E}[\|W_0^{(i)}\|] \leq \sqrt{\|\mu\|^2 + \mathrm{tr}(\Sigma)}$$

From Theorem 2, we have the following:

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \left\|W_0^{(i)}\right\| + 2^{t+1} \left\|W_*^{(i)}\right\|\right) \left(\sum_{u\in\mathcal{V}} \hat{w_u}\right)$$

Hence, combining the two elements results in the following:

$$\mathbb{E}_{W_0\sim\mathcal{N}(\mu,\Sigma)}[\mathcal{R}_\epsilon[f]] \leq \epsilon \prod_{i=1}^{T} \left(2^t \sqrt{\mu^2 + \mathrm{tr}(\Sigma)} + 2^{t+1} \left\|W_*^{(i)}\right\|\right) \left(\sum_{u\in\mathcal{V}} \hat{w_u}\right)$$

□

D Proof Of Theorem 5

Theorem. *Let $f : (\mathcal{A}, \mathcal{X}) \to \mathcal{Y}$ denote a graph-based function composed of T GIN layers, where the initial weight matrix of the i-th layer is denoted by $W_0^{(i)}$. For adversarial attacks only targeting node features of the input graph, with a budget ϵ, we have:*

$$\gamma = \prod_{l=1}^{T} \left(2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\|\right) \left[BT \max_{u\in\mathcal{V}} deg(u) + \epsilon\right]$$

with t being the number of training epochs and $deg(u)$ is the degree of node u.

Proof. Let's consider a graph-function f that is based on T GIN-layers and trained using gradient descent for t epochs. We have the following result from Equation 3:

$$\|W_t^{(i)}\| \leq (1+\eta L)^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\| \leq 2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\| \tag{8}$$

Let X denote the original node features and X' the perturbed adversarial features. For an attack budget ϵ, from the work [1], we have the following:

$$\forall [A', X] \in B([A, X], \epsilon), \|f(A, X) - f(A, X')\| \leq \prod_{l=1}^{T} \|W^{(l)}\| [B \times T \times \max_{u\in\mathcal{V}} deg(u) + \epsilon] \tag{9}$$

Consequently, we can merge the two inequalities resulting in the following:

$$\gamma = \prod_{l=1}^{T} \left(2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\|\right) \left[B \times T \times \max_{u\in\mathcal{V}} deg(u) + \epsilon\right]$$

□

E Proof Of Theorem 6

Theorem. *Let $f : \mathcal{X} \subseteq \mathbf{R}^{in} \to \mathcal{Y} \subseteq \mathbf{R}^{out}$ be a T-layers neural network with $W_0^{(i)}$ denoting the initial weight matrix of the i-th layer. When subject to adversarial attacks, f is $(\epsilon, \gamma)-$robust with:*

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\|\right)$$

Proof. Let f be a T-layers neural network. We additionally assume that its corresponding activation functions are 1-Lipschitz. Let x (with h its hidden representation) be an input vector and x' (corresp. h') its corresponding crafted adversarial input (corresp. hidden representation). For an adversarial attack with budget ϵ, we have the following:

$$\begin{aligned}\forall x' \in \mathcal{X} : \|x - x'\| \le \epsilon, \|f(x) - f(x')\| &= \|h^{(l)} - h'^{(l)}\| \\ &= \|\phi^{(l)}(W^{(l)}h^{(l-1)} + b^{(l)}) - \phi^{(l)}(W^{(l)}h'^{(l-1)} + b^{(l)})\| \\ &\le \|W^{(l)}\|\|h^{(l-1)} - h'^{(l-1)}\|\end{aligned}$$

Recurrently, we find the final result as:

$$\sup_{x' \in \mathcal{X}: \|x - x'\| \le \epsilon} \|f(x) - f(x')\| \le \prod_{l=1}^{T} \|W^{(l)}\|\epsilon \tag{10}$$

Note that similar results and analysis have been provided in previous work [5, 2]. By using the result derived in Equation 3, we have:

$$\|W_t^{(i)}\| \le 2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\| \tag{11}$$

By merging these two inequalities, and applying the Markov Inequality, we find the following upper-bound:

$$\gamma = \epsilon \prod_{i=1}^{T} \left(2^t \|W_0^{(i)}\| + 2^{t+1} \|W_*^{(i)}\|\right)$$

□

F On the case of strong-convexity - Proof of Lemma 7

Lemma. *Let $f : \mathcal{X} \subseteq \mathbf{R}^{in} \to \mathcal{Y} \subseteq \mathbf{R}^{out}$ be a T-layers neural network trained with a μ-strongly convex and L-smooth loss function. Let $W_0^{(i)}$ denote the initial weight matrix of the i-th layer. When subject to adversarial attacks, with a budget ϵ, we have that f is $(\epsilon, \gamma) -$ robust with:*

$$\gamma = \epsilon \prod_{i=1}^{T} \left((1 - \mu/L)^t \|W_0^{(i)}\| + 2\|W_*^{(i)}\|\right)$$

Proof. We consider f to be a T-layers neural network (following the same propagation as equation the one presented in Section 5). From Section E, we have the following:

$$\|f(x) - f(x')\| \le \prod_{l=1}^{T} \|W^{(l)}\|\epsilon$$

In addition to the previous assumption of L-smoothness of the loss function, we consider that its μ-strongly convex. Hence, for the layer (l), we have the following result:

$$\begin{aligned}\|W_t^{(l)}\| &\le (1 - \mu/L)^t \|W_0^{(l)} - W_*^{(l)}\| + \|W_*^{(l)}\| &(12)\\ &\le (1 - \mu/L)^t \|W_0^{(l)}\| + 2\|W_*^{(l)}\| &(13)\end{aligned}$$

When subject to adversarial attacks, we can use the previous result from E, specifically from Equation10:

$$\sup_{x' \in \mathcal{X}: \|x - x'\| \le \epsilon} \|f(x) - f(x')\| \le \prod_{l=1}^{T} \|W^{(l)}\|\epsilon \tag{14}$$

Hence, by merging the two previous results, we deduce that:

$$\gamma = \epsilon \prod_{i=1}^{T} \left((1-\mu/L)^t \|W_0^{(i)}\| + 2\|W_*^{(i)}\| \right) \tag{15}$$

$\square$

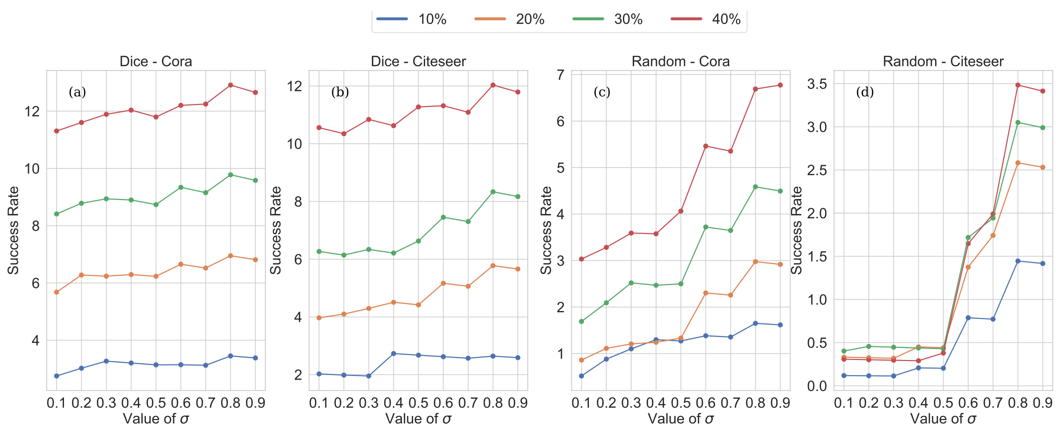


Figure 5: Effect of the variance on the model's robustness in the case of Gaussian Initialization when subject to DICE (a,b) and Random Attacks (c,d) for both Cora and CiteSeer.

G Additional Results

G.1 Adversarial Robustness of Deep Neural Networks

We consider the general family of neural networks for which the computation during layer l, using an activation function $\phi^{(l)}$, can be written as :

$$h^{(l)} = \phi^{(l)}(W^{(l)} h^{(l-1)} + b^{(l)}).$$

with $W^{(l)} \in \mathbb{R}^{n_{l-1}, n_l}$ being the weight matrix and $b_l \in \mathbb{R}^{n_l}$ the bias of the l^{th} layer.

In this perspective, let $f : \mathbb{R}^{n_0} \to \mathbb{R}$ be a neural network n_0 being the input dimension. The adversarial task in this case consists of finding a perturbed input $\tilde{x}$ for which the prediction differs from the original prediction $f(x)$. The perturbed input $\tilde{x}$ should hence adhere to the similarity constraints defined by a perturbation budget ϵ. Let's consider the ℓ_2 norm within both the input space $\mathbb{R}^{n_0}$ and the output space $\mathbb{R}$, we can hence define the set of valid adversarial perturbation as:

$$B(x;\epsilon) = \{\tilde{x} : \|x - \tilde{x}\| \leq \epsilon\}$$

Similar to Section 3, we can introduce the adversarial risk of a DNN within the input's neighborhood defined by the budget ϵ as the following:

$$\mathcal{R}_\epsilon[f] = \mathop{\mathbb{E}}_{\substack{x \sim \mathcal{D} \\ \tilde{x} \in B(x;\epsilon)}} [\|(f(\tilde{x}) - f(x)\|]. \tag{16}$$

From this adapted adversarial risk, we can introduce the notion of a DNN's adversarial robustness

Definition 8. (DNN - Adversarial Robustness). The neural network $f : \mathbb{R}^{n_0} \to \mathbb{R}$ is said to be (ϵ, γ) – robust if its adversarial risk is upper-bounded by γ, i. e., $\mathcal{R}_\epsilon[f] \leq \gamma$.

G.2 Additional Adversarial Attacks

In addition to the previously reported Mettack and PGD adversarial attack, we consider two additional adversarial attacks. Notably, we first consider "DICE" which involves iteratively perturbing a graph's structure by adding or removing edges while ensuring connectivity, and then adjusting the perturbation based on the gradient of the graph neural network's loss function to generate an adversarial example. The process aims to find a minimal perturbation that misleads the network's predictions while keeping the perturbation size small. We additionally consider a "Random" attack which consists of randomly perturbing the adjacency matrix by dropping or adding edges. Figure 5 shows the adversarial accuracy results on the Cora and CiteSeer dataset when subject to DICE and Random attacks for different values of σ of the Gaussian initialization. Similarly, Figure 6 shows the effect of scaling both a uniform initialization and an Orthogonal one as previously explained in Section 6.

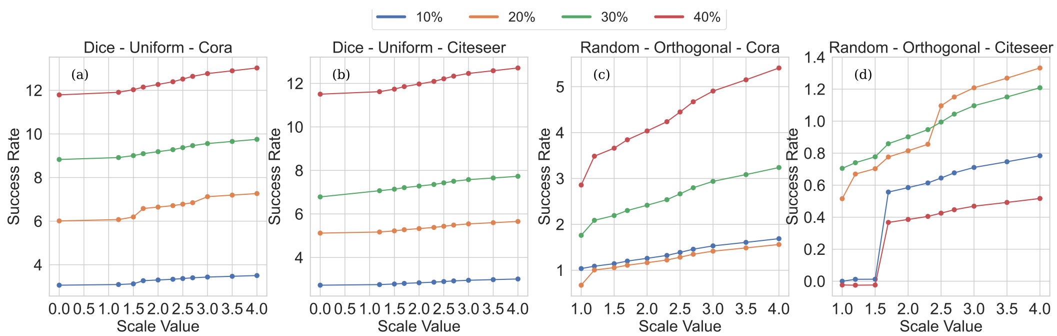


Figure 6: Effect of Uniform and Orthogonal Initialization on the model's robustness in the case of DICE Attack on Cora (a,c) and CiteSeer (b,d).

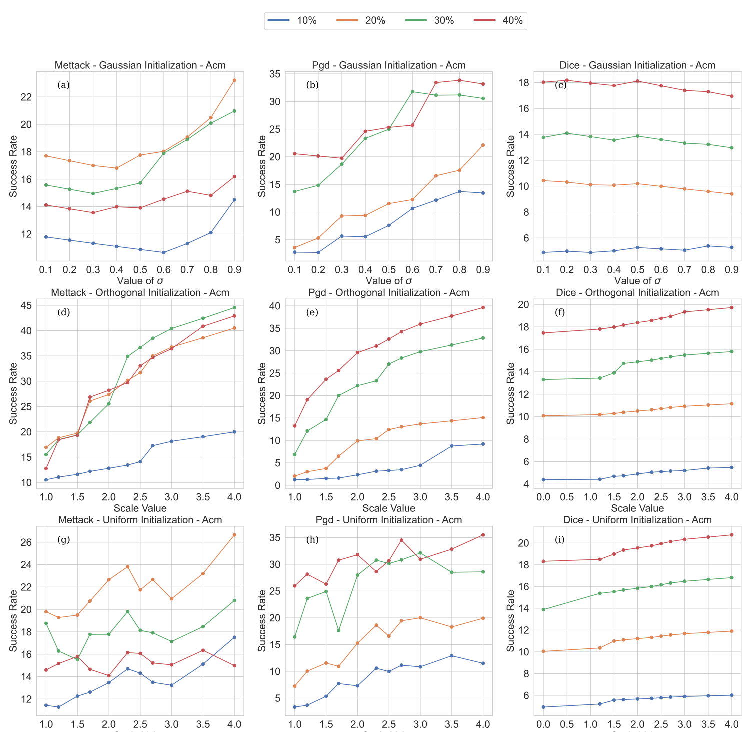


Figure 7: Effect of the Gaussian (a; b; c), Orthogonal (d; e; f) and Uniform (g;h;i) Initialization on the ACM dataset

G.3 Additional Datasets

We additionally extend the results to the ACM Dataset [30] within the node classification setting. Figure 7 presents the results using the Mettack, PGD and DICE for the ACM dataset for the Gaussian initialization (effect of σ), the Uniform and Orthogonal initialization.

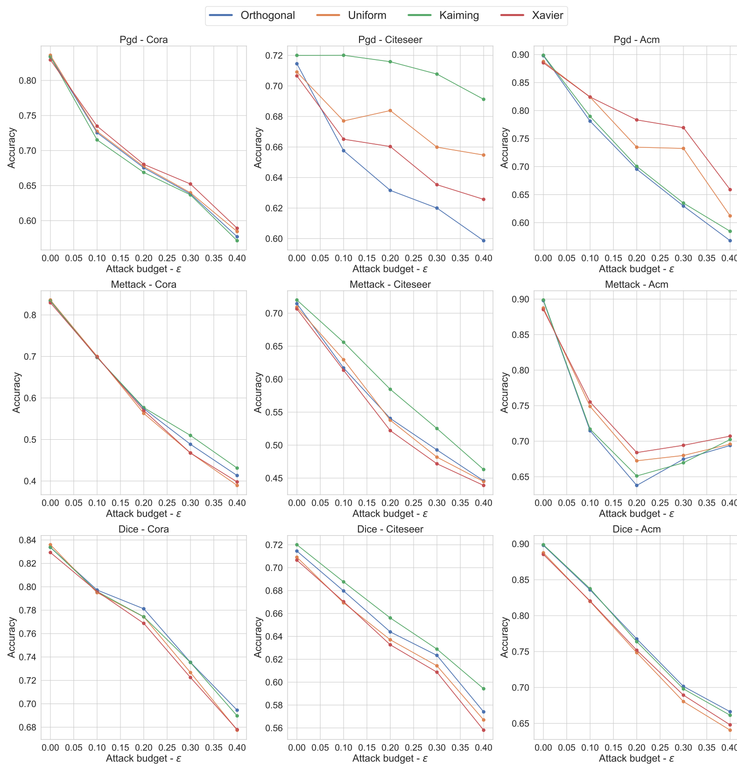


Figure 8: Effect of the initial distribution on RGCN's robustness and performance when subject to structural adversarial attacks.

G.4 Additional Models

As previously explained in Section 5, while our theoretical analysis primarily focuses on GCN, GIN, and DNN models, the derived insights extend to other models as well. To illustrate this point, we examine the effect of initialization distribution on the performance of defense methodologies. Specifically, we first consider RGCN [38], which employs Gaussian distributions in its hidden layers to mitigate the effects of adversarial attacks. We additionally consider GCN-Jaccard [31] which preprocesses the network by eliminating edges that connect nodes with jaccard similarity of features smaller than a certain level. We use various initialization schemes, similar to those in our previous experiments, and evaluate against the same adversarial attacks (PGD, Mettack, and DICE). Figure 8 (resp. Figure 9) presents the adversarial accuracy and defense performance of RGCN (resp. GCN-Jaccard) on the Cora, CiteSeer, and ACM datasets. Although the performance gap is not very pronounced for Cora, it is clearly observed for CiteSeer and ACM. This demonstrates the broader applicability of our insights across different models but also defense methods.

H Datasets and Implementation details

Datasets Characteristics and information about the node classification datasets used in our experimental study are presented in Table 1. As outlined in the main paper, we conduct experiments on a set of citation networks, including Cora, CiteSeer (in the main paper), and ACM dataset (Appendix G) [30]. For all these datasets, we adhere to the train/valid/test splits provided by with the dataset.

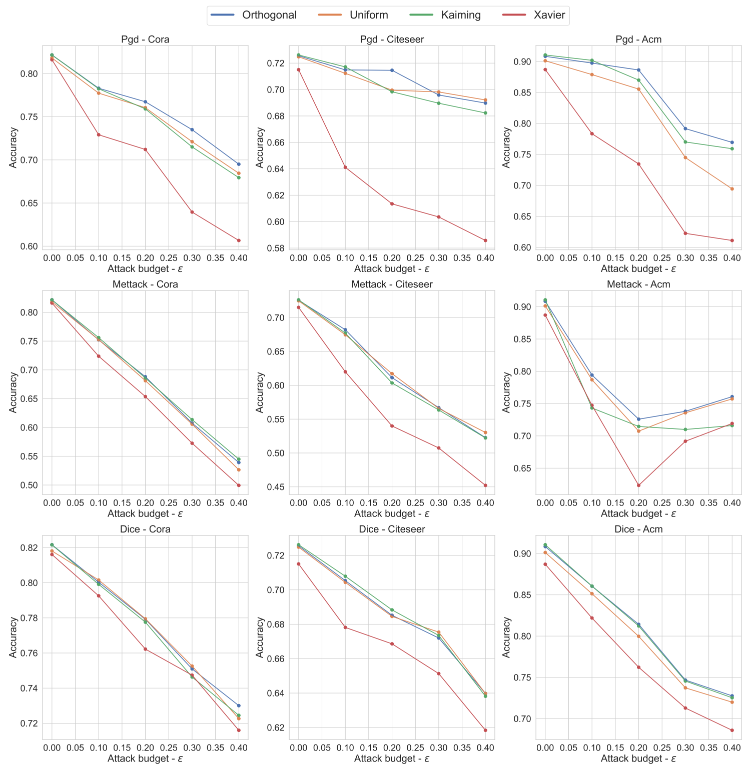


Figure 9: Effect of the initial distribution on GCN-Jaccard's robustness and performance when subject to structural adversarial attacks.

About the architectures. In all of the experiments, the models employed a 2-layer convolutional architecture (consisting of two iterations of message passing and updating) stacked with a Multi-Layer Perception (MLP) as a readout. The intent was to compare the models in an iso-architectural setting, to ensure a fair evaluation of their robustness. We maintained the same hyperparameters, including a learning rate of 1e-2, 300 epochs, and a hidden feature dimension of 16 have been. To account for the impact of random initialization, each experiment was repeated 10 times.

Reproducibility of the experiments. We emphasize that all experiments should be easily reproducible by directly using the provided code. The archive contains a ReadMe file containing a small documentation on how to run the experiments.

Table 1: Statistics of the node classification datasets used in our experiments.

DATASET	#FEATURES	#NODES	#EDGES	#CLASSES
CORA	1433	2708	5208	7
CITESEER	3703	3327	4552	6

On the adversarial attacks. For the PGD attack on the MNIST dataset, we used a step-size of 0.1 and we set the number of iterations to 100 (which was observed to be enough for the attack convergence). Note that we set these parameters for all the considered initializations in Figure 4 as our aim is to compare the effect of the different distribution on the final robustness.

Implementation details. Our implementation is available in the supplementary materials (and will be publicly available afterwards). It is built using the open-source library *PyTorch Geometric* (PyG) under the MIT license [11]. We used the publicly available implementation of the adversarial attacks provided in the DeepRobust package (https://github.com/DSE-MSU/DeepRobust). For RGCN, we used the implementation from the same package. The experiments have been run on both a NVIDIA A100 GPU where training a GCN takes around $1.2(\pm 0.2)$ s.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In addition to stating the novelty of our proposed approach, we used our abstract and introduction to summarize our main findings and contributions related to the effect of initialization on the adversarial robustness (as theoretically justified and empirically tested in the following sections).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Together with our conclusion, we presented the set of limitations of work. Specifically, we stated that while our work is innovative, we didn't provide a solution to the initialization problem from an adversarial defense perspective. We also discussed in the "problem setup" section our different theoretical choices (the smoothness of the loss function) and how realistic they are.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For each Theorem, Lemma and theoretical claim, we provide the proof in the Appendix and point out to the corresponding section in the main paper. We also stated all the assumptions and analytical choices in the Preliminaries (Section 3.1)

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In addition to providing the code as supplementary materials, we have provided all the implementations details that are sufficient to reproduce the results. These details include the used hyper-parameters (the architecture, learning rate ...) and also for the used adversarial attacks we provide the different parameters used. We also point out the dataset that we used (which are public) and that we used the same public folds as the one provided with the datasets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the anonymized code following the Neurips guidelines. Specifically, we submitted the code with the supplementary material section and we clearly state the steps to run it using a ReadMe file. Please note that for this question, we consider "open source" as providing the code to the reviewers and making it public afterwards for the public.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provided all the details about the architecture, the used hyper-parameters for the considered models (Section H of the Appendix) and all the hyper-parameters used for our adversarial attacks. Note that our work's goal is to provide comprehensive overview of the effect of initialization on the robustness, hence making sure that the same choice of hyper-parameters is enough to ensure the fairness of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We reproduce each experiment 10 times to take into account the factor of randomization and we report the mean value. Note that since we use mainly figures (which are appropriate for our setting – given the different attack budgets we are using), this seemed as the perfect approach. For the train/test folds, we use the public folds provided with each dataset and hence reducing the effect of randomization.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We reported the details of implementation in Section H of the Appendix, where we specified the GPU that was used and the average time to do the experiments. Note that while we have chosen to use a GPU, our experiments can be easily done using a CPU.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the guidelines of the Neurips Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provided overview on the harm that adversarial attacks can have on the applications of Deep Learning models. The main goal of our paper is to identify new potential factors related to adversarial attacks and hence should rather have a positive impact on the society.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: In this work, we study the theoretical effect of initialization on the adversarial robustness. We don't provide any new pre-trained model nor new datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We made sure to cite the papers that are relevant to our work and that were used to justify some theoretical or empirical insights. For the different code implementations, we cited clearly the license and the owner of the used function/code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have provided the implementation code together with all the experimental details to reproduce our work. We also clearly justify the use of the packages and their license. Note that the code have been anonymized and provided as a supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: There is no crowdsourcing nor research with human subjects in our case.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There is no crowdsourcing nor research with human subjects in our case.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.