

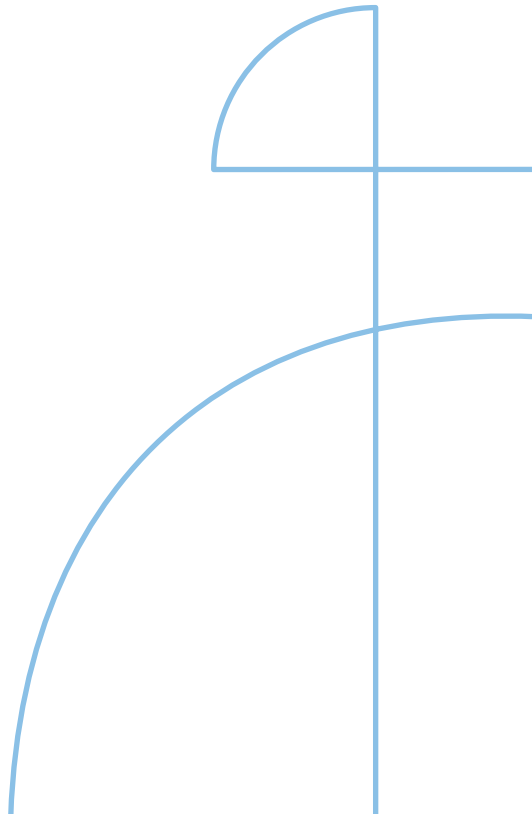


Doctoral Thesis in Information and Communication Technology

On the Adversarial Robustness of Graph Neural Networks

SOFIANE ENNADIR

KTH ROYAL INSTITUTE OF TECHNOLOGY



On the Adversarial Robustness of Graph Neural Networks

SOFIANE ENNADIR

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Thursday the 23rd of April 2026, at 2:00 p.m. in room Kollegiesalen, Brinellvägen 8, Stockholm.

Doctoral Thesis in Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden 2026

© Sofiane Ennadir

TRITA-EECS-AVL-2026:29
ISBN 978-91-8106-573-2

Printed by: Universitetservice US-AB, Sweden 2026

Abstract

Graph Neural Networks (GNNs) have emerged as the standard paradigm for machine learning on graph-structured data, demonstrating remarkable success in diverse applications such as molecular design, anomaly detection within networks, and recommendation systems. However, despite their effectiveness in learning meaningful representations for nodes and graphs, GNNs remain vulnerable to adversarial attacks. These attacks, which are small strategically crafted perturbations to the input graph, can result in unreliable predictions. This specific vulnerability raises serious concerns regarding the deployment of GNNs in safety-critical domains like finance and healthcare, where ensuring robustness is crucial. Consequently, understanding and enhancing the adversarial robustness of GNNs has become a critical research focus, involving both the design of potent attack strategies and the development of resilient defense mechanisms.

Many existing defense methods rely on pre-processing techniques or modifications to the message-passing framework to mitigate attacks, often by discarding or re-weighting parts of the input graph. Although these defenses have shown great results, they are frequently based on heuristic reasoning and lack strong theoretical guarantees. Specifically, given the input graphs' rich topological aspect, a deeper understanding of their vulnerabilities and internal behaviors is essential, especially regarding how an attack can propagate through the network. Moreover, current defense methodologies are typically evaluated only against the state-of-the-art attacks available at the evaluation time; in the absence of theoretical guarantees, these defenses remain susceptible to more advanced or previously unseen attack strategies. This gap underscores the need for mechanisms that not only exhibit robust empirical performance but also provide certifiable robustness for long-term effectiveness. Furthermore, most current approaches entail high computational overhead, limiting their practical feasibility in real-world applications.

In this thesis, we address key challenges in GNN adversarial robustness, focusing on the aforementioned drawbacks. First, we introduce defense mechanisms that are both empirically effective and grounded in solid theoretical analysis, thereby offering provable robustness against evolving attacks. Second,

we investigate how to reconcile strong defense performance with computational efficiency, which is an essential requirement in multiple domains such as applications in the mobile and online platforms. Achieving this balance is critical for broadening the deployment of robust GNNs in practical settings. Finally, we explore often overlooked factors related to the training dynamics, such as weight initialization and the number of training epochs, that can substantially influence a model's underlying robustness, illustrating how effective parameter selection can bolster resilience with very limited costs.

The contributions of this thesis are organized around four core pillars. In the first, we propose an adaptation of Graph Convolutional Networks (GCNs) using orthogonal weight matrices, showing both theoretically and empirically that this design can significantly enhance model robustness. In the second contribution, we present a simple yet powerful technique for injecting noise into hidden representations during training, which substantially improves robustness with minimal additional computational cost, consequently offering a more lightweight alternative to many existing, high-complexity defense methods. The third work examines the neglected interplay between training dynamics (e.g., number of epochs, initialization strategies) and model vulnerability, demonstrating how careful tuning of these parameters can enhance a model's underlying robustness. Finally, we propose a novel adversarial attack approach that generates adversarial graphs from scratch via a learnable generator, rather than merely perturbing existing graphs, thereby introducing new perspectives on attack methodologies.

Through these contributions, the current thesis aims to provide theoretical insights and tools that could help advance the current understanding of adversarial attacks in the context of GNNs. These contributions and insights can advance the development of robust GNNs, paving the way for safer and more reliable graph-based machine learning systems.

Sammanfattning

Graph Neural Networks (GNNs) har etablerat sig som ett standardparadigm för maskininlärning på grafstrukturerad data och har visat stor framgång inom tillämpningar som molekyldesign, anomalidetektion i nätverk och rekommendationssystem. Trots deras förmåga att lära sig meningsfulla representationer för noder och grafer är GNNs sårbara för adversarial attacks, det vill säga små, strategiskt utformade perturbationer i indata som kan leda till opålitliga prediktioner. Denna sårbarhet väcker allvarliga farhågor vid användning i säkerhetskritiska domäner såsom finans och sjukvård, där robusthet är avgörande. Följaktligen har förståelsen och förbättringen av GNNs adversarial robustness blivit ett centralt forskningsområde, innefattande både utveckling av effektiva attackstrategier och motståndskraftiga försvarsmekanismer.

Många befintliga defense methods bygger på preprocessing-tekniker eller modifieringar av message passing-ramverket, ofta genom att filtrera eller omvikta delar av grafen. Trots god empirisk prestanda baseras dessa metoder ofta på heuristik och saknar starka teoretiska garantier. Givet grafernas rika topologiska struktur krävs en djupare förståelse av deras sårbarheter och interna dynamik, särskilt hur en attack kan spridas genom nätverket. Dessutom utvärderas försvar vanligtvis endast mot state-of-the-art attacks vid utvärderingstillfället, vilket gör dem sårbara för mer avancerade eller tidigare okända strategier. Detta belyser behovet av metoder som kombinerar god empirisk prestanda med certifierbar robusthet. Samtidigt innebär många nuvarande angreppssätt hög computational overhead, vilket begränsar deras praktiska användbarhet.

Denna avhandling adresserar centrala utmaningar inom adversarial robustness för GNNs. För det första introduceras defense mechanisms som är både empiriskt effektiva och teoretiskt grundade, med provable robustness mot föränderliga attacker. För det andra undersöks hur stark defense performance kan förenas med computational efficiency, vilket är avgörande för tillämpningar i exempelvis mobila och onlinebaserade system. För det tredje analyseras ofta förbisedda faktorer i training dynamics, såsom weight initialization och antal training epochs, och hur dessa påverkar modellens robusthet, där noggrant parameterurval kan ge betydande förbättringar till låg kostnad.

Avhandlingens bidrag organiseras kring fyra huvudområden. För det första föreslås en modifiering av Graph Convolutional Networks (GCNs) med orthogonal weight matrices som teoretiskt och empiriskt förbättrar robustheten. För det andra presenteras en enkel men effektiv metod för att injicera noise i hidden representations under träning, vilket ger ökad robusthet med låg computational cost. För det tredje analyseras samspelet mellan training dynamics och sårbarhet, vilket visar hur parametrering påverkar robustheten. Slutligen introduceras en ny adversarial attack-metod där grafer genereras från grunden via en learnable generator, snarare än att enbart perturb befintliga grafer.

Sammanfattningsvis bidrar avhandlingen med teoretiska insikter och praktiska verktyg som fördjupar förståelsen av adversarial attacks i GNNs och främjar utvecklingen av mer robusta och tillförlitliga grafbaserade maskininlärningssystem.

Acknowledgements

Embarking on the path toward a Ph.D. has been a deeply transformative journey, marked by challenges, hardships, and moments of solitude that were sometimes difficult to face. Nonetheless, these moments were often outweighed by the joy of reaching meaningful milestones, whether through research papers or valuable implementations. I feel incredibly fortunate to have followed this path. Beyond gaining knowledge in the field of Artificial Intelligence, I have also learned how to pursue impactful and meaningful research. This thesis is the culmination of several years of work, and it would not have been possible without the support of many people I met during my engineering and graduate studies. With sincere gratitude, I thank everyone who played a role in this journey.

First, I would like to thank my two supervisors. I am grateful to my main supervisor, Professor Michalis Vazirgiannis, for his guidance in shaping the overall research direction of this thesis and for helping me reach key milestones along the way. I would also like to express my deep gratitude to Professor Henrik Boström, from whom I learned not only about research but also about the broader perspective of academic life. Thank you for your support throughout my doctoral studies, as well as for the many engaging discussions, meetings, and lunches we shared.

To my friends and colleagues at KTH, thank you for the enriching collaborations and discussions, both within and beyond my research. I would particularly like to thank Dr. Amr Alkhatib, with whom I had the pleasure of collaborating on several papers and traveling to conferences and study trips. Our many conversations about research, work, and life have been a constant source of inspiration and support. I would also like to thank Tianze Wang, who has been a great friend over the years and with whom I have had the pleasure of collaborating after joining King.

I was also fortunate to be very close to the DaScim team at École Polytechnique. I would like to warmly thank all its members for their time and the enriching discussions during my visits. In particular, I am grateful to Professor Johannes Lutzeyer, with whom I collaborated on several research papers. Through this collaboration, I learned how theoretical ideas can evolve into practical and impactful insights. I would also like to thank Dr. Yassine Abbahaddou, my close collaborator, for the many insightful discussions that led to meaningful research outcomes. Additionally, I would like to express my sincere gratitude to Professor El Houcine Bergou from UM6P, with whom I had the pleasure of collaborating and with whom I continue to work on several research topics.

I gratefully acknowledge the support of the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation. This support allowed me to participate in leading European and international machine learning venues.

During my Ph.D., I also had the opportunity to undertake several internships and research visits. I am thankful to the Simons Foundation and Polymathic AI, where I had the pleasure of working with Siavash Golkar and Leopoldo Sarra, whom I thank for their excellent collaboration and mentorship. I also had the opportunity to work with King, first as an intern and now as a full-time employee, where I was able to explore the practical applications of my research on Graph Neural Networks. I would therefore like to express my sincere gratitude to the AI Labs team for their support throughout this journey.

Finally, I am profoundly grateful to my small and beloved family. To my brother Mehdi and my father, thank you for your constant support throughout my life and studies. I would like to express my deepest gratitude to my mother, who has played the most important role in my life. From teaching me the basics of mathematics to sharing valuable life lessons, her unwavering support has always helped me overcome challenges and stay motivated. This thesis and all the work behind it are dedicated to you. It is difficult to fully express how grateful I am for everything you have done for me.

Lastly, I would like to thank all my fellow Moroccan friends (Yassir, Amine, Soukaina, Youness, Zakaria, and his family), whom I have had the pleasure of knowing throughout my life, from my engineering school years to my time in Sweden.

List of Contributions

List of included papers

Paper A

Ennadir, S.* , Abbahadou, Y.* , Lutzeyer, J., Vazirgiannis, M., Boström, H.
“Bounding the Expected Robustness of Graph Neural Networks Subject to Node Feature Attacks”.

In the Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria.

Paper B

Ennadir, S., Abbahadou, Y., Lutzeyer, J., Vazirgiannis, M., Boström, H.
“A Simple and Yet Fairly Effective Defense for Graph Neural Networks”.

In the Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Vancouver, Canada.

Paper C

Ennadir, S., Lutzeyer, J., Vazirgiannis, M., Bergou, E.
“If You Want to Be Robust, Be Wary of Initialization”.

In the Thirty-Eighth Annual Conference on Neural Information Processing Systems, Neurips 2024, Vancouver, Canada.

Paper D

Ennadir, S., Alkhatib, A., Nikolentzos, G., Vazirgiannis, M., Boström, H.
“UnboundAttack : Generating Unbounded Adversarial Attacks to Graph Neural Networks”.

In the 12th International Conference on Complex Networks, CNA 2023, Menton, France.

Paper E

Ennadir, S., Alkhatib, A., Boström, H., Vazirgiannis, M.
“Conformalized Adversarial Attack Detection for Graph Neural Networks”.

In Proceedings of the 12th Symposium on Conformal and Probabilistic Prediction with Applications.

Volume 204 of Proceedings of Machine Learning Research, pages 450–469.
PMLR, 13–15 Sep 2023

List of Excluded Papers

Paper F

Ennadir, S.* , Gandler, G.* , Cornell, F.* , Cao, L., Smirnov, O., Wang, T., Zolyomi, L., Brinne, B., Asadi, S.

“Expressivity of Representation Learning on Continuous-Time Dynamic Graphs: An Information-Flow Centric Review”.

Transactions on Machine Learning Research (TMLR), April 2025.

Paper G

Ennadir, S., Golkar, S., Sarra, L.

“Joint Embedding go Temporal”.

NeurIPS Workshop on Time Series in the Age of Large Models, December 2024.

Contents

1	Introduction	1
1.1	Research Landscape	3
1.2	Research Challenges	5
1.3	Research Objectives	7
1.3.1	Research Questions	7
1.3.2	Thesis Contributions	8
1.4	Author Contribution	12
1.5	Outline	12
2	Background	13
2.1	Graph Neural Networks	13
2.1.1	Formalizing the Classification Problem	13
2.1.2	Graph Neural Networks	14
2.2	Adversarial Attacks	18
2.2.1	Effect of Perturbations	18
2.2.2	Crafting the Attacks	21
2.3	Research Methodology	23
2.3.1	Datasets	24
2.3.2	Evaluation Metrics	24
3	Contributions - Adversarial Defenses	27
3.1	Adversarial Risk and Vulnerability	27
3.2	Paper A – Connecting Topology to Adversarial Robustness	29
3.2.1	Paper A - On the Effect of Orthonormal Weights	31
3.3	Paper B - Injecting Noise as a Defense	33
3.4	Paper C – On the Effect of Training Dynamics	36
4	Other Contributions	41
4.1	Paper D - Adversarial Unbounded Attacks	41
4.2	Paper E - Conformalized Adversarial Detection	44
5	Conclusions and Future Work	47

Abbreviations

Graph Related

GAT Graph Attention Network

GCN Graph Convolutional Networks

GCORN Graph Convolutional Orthonormal Robust Network

GIN Graph Isomorphism Network

GNN Graph Neural Networks

NoisyGNN Noise-Injected Graph Neural Network (proposed defense)

PGD Proximal Gradient Descent

General Elements

DL Deep Learning

LLM Large Language Models

ML Machine Learning

NLP Natural Language Processing

RL Reinforcement Learning

Chapter 1

Introduction

Machine learning has experienced rapid advances over the past decade, driven by the advent of innovative architectures and sophisticated optimization techniques. These innovations have enabled deep learning (DL) methods to attain state-of-the-art performance in numerous tasks, from Natural Language Processing (NLP), where Large Language Models (LLMs) such as GPT-4 [1] have revolutionized language understanding and generation, to computer vision [2], where Transformer-based models have pushed the frontiers of image recognition, object detection, and more. In parallel, classical neural architectures have continued to evolve, enhancing their representational capacity and enabling them to tackle increasingly complex real-world applications.

Amid these developments, Graph Neural Networks (GNNs) [3], [4], [5] have emerged as a particularly powerful class of models designed for learning on graph-structured data. Unlike standard neural networks that operate on fixed-size vector inputs or images, GNNs excel at capturing the relational and topological information encoded in graphs. In their generic form, these models are based on aggregation and transformation of node features through a series of message-passing layers [6], enabling them to learn representations that incorporate both node attributes and the topological structure through k -hop neighbors. This ability makes GNNs ideally suited for tasks such as node classification, graph classification, and link prediction, with demonstrated success across various domains. For instance, these models have been used in social networks to identify spammers, detect fake accounts, or predict community membership [7]. Additionally, they have been employed in modeling molecular structures [8] to predict different properties aiding in drug discovery [9], and finally predicting user-item relationships to forecast the following item a user is likely to purchase in the context of recommendation systems [10].

As Deep Learning-based models continue to excel in terms of performance and success, their potential for real-world deployment grows, but so do concerns about their reliability, safety, and ethical implications. Additional aware-

CHAPTER 1. INTRODUCTION

ness of these issues has led to the emergence of Trustworthy Machine Learning, a research perspective dedicated to ensuring that Machine Learning models are generally transparent, fair, and robust. Transparency and interpretability, for instance, allow the researchers and the end-users to understand and trust a model's decision-making process. At the same time, fairness ensures that sensitive demographic groups are not disproportionately harmed or disadvantaged. Another critical aspect of trustworthy ML is adversarial robustness, which addresses how reliably a model behaves under "adversarial" or "maliciously crafted" perturbations.

Deep Learning-based models, including those used in computer vision, have been found vulnerable to adversarial attacks [11], which are small and carefully crafted input modifications that can dramatically alter the model's predictions. Graph Neural Networks, which can be seen as a generalization of Deep Learning to graph modality, are no exception, and it has been shown [12] that they could likewise be misled by slight edits to node features or the graph topology. Such vulnerabilities raise pressing concerns for safety-critical applications (e.g., finance, healthcare, security), where errors or manipulated predictions can cause significant harm. In the specific case of graphs, different attack scenarios are possible, where the adversarial modifications may include adding or deleting edges, injecting or deleting nodes, or strategically tweaking node features to force the GNN into making incorrect predictions.

Understanding and mitigating these adversarial risks has consequently become a major research directive within the GNN community. Broadly speaking, the literature on GNN adversarial robustness can be divided into two main directions:

- **Attack Mechanisms:** Research in this direction aims to develop more effective adversarial strategies, like gradient-based or heuristic-based, that exploit the structural and feature properties of graphs. These methods continually evolve, seeking ways to remain undetected while maximizing the impact on a model's predictions.
- **Defense Techniques:** On the other side, researchers strive to design defenses that safeguard GNNs from adversarial perturbations, ensuring stable performance despite malicious edits.

In an ideal scenario, the two-player interplay between attacks and defenses would converge toward an equilibrium, in which a robust, trustworthy GNN emerges that can withstand the strongest available attack strategies while maintaining high accuracy on clean data. In this perspective, the current thesis aims to explore the adversarial robustness of GNNs by theoretically investigating the effect of a perturbation on a GNN and, therefore, deriving new defense methods that can be very effective while taking into account possible added complexity.

1.1 Research Landscape

As previously introduced, the research landscape within the context of studying the adversarial robustness of GNNs has recently attracted significant attention [13]. Specifically, a variety of attack mechanisms [14], [15], [16], [17], [18], [19], [20], [21] have been proposed to investigate how robust an underlying GNN model is against carefully crafted perturbations. We note that the attack research landscape can be broadly divided into three categories based on the level of knowledge available to the attacker. First, in white-box attacks, the adversary has complete access to the model's internal workings, including its architecture, parameters, and gradients, which allows for highly precise perturbations. For instance, in applications such as financial fraud detection based on monitoring network traffic, an insider or a highly knowledgeable external adversary might exploit full system access to craft adversarial modifications that directly target the vulnerabilities of the deployed GNN model. This complete transparency enables attackers to use gradient-based methods to compute the optimal perturbations that maximize the misclassification rate with minimal modifications. Second, gray-box attacks assume that the attacker has only partial information about the target model. This might include knowledge of the model's architecture or access to a surrogate model that approximates the target's behavior. A practical example of a gray-box scenario can be seen in recommendation systems, where an attacker might not know the full details of the recommendation algorithm but could gain insights by analyzing similar public models or historical data to construct a surrogate. With this approximation, the attacker can craft adversarial examples that are likely to transfer to the real system, thereby subtly manipulating product recommendations or influencing consumer behavior. Finally, black-box attacks are conducted under minimal knowledge conditions, where the attacker has no insight into the internal parameters or structure of the model and may only observe its outputs or predicted labels. A concrete application of black-box attacks is in social network analysis, where external entities may only see the outcomes of content moderation or community detection algorithms. In such scenarios, adversaries might use query-based strategies in an iterative fashion to modify the network structure or node attributes based solely on the observable predictions. This iterative process can eventually converge and induce significant misclassification, even without knowing the inner workings of the algorithm.

Most attack methods treat the problem of compromising an input graph as an optimization task, aiming to produce an altered graph that remains almost indistinguishable from the original while eliciting incorrect predictions from the target model. For instance, Mettack [15] casts this challenge as a bilevel optimization problem where the goal is to degrade the overall performance of the GNN, not merely to misclassify individual nodes. In this framework, the

CHAPTER 1. INTRODUCTION

graph structure is considered a hyperparameter, and meta-gradients, which are computed during backpropagation through the training process, are used to identify the minimal edge insertions or deletions that will most adversely affect the model's training loss. This approach demonstrates that even with limited information (such as access to the observed graph structure and a subset of node labels), an attacker can iteratively select perturbations that preserve key graph properties (like the degree distribution) while substantially impairing the model's predictive capability. Another notable example of this formulation is Nettack [14], which proceeds through a surrogate model based on a linearized graph convolutional network. In the proposed framework, the attacker iteratively selects minimal perturbations, either by adding or removing edges or by modifying node features, that maximize a surrogate loss aimed at a specific target node. In parallel and to ensure that the changes remain unnoticeable, the method enforces constraints that preserve essential graph properties such as the degree distribution and feature co-occurrence patterns. From another perspective, approaching the adversarial aim can be performed using Reinforcement learning. Specifically, the adversarial task could be framed as a sequential decision-making problem. For instance, RL-S2V [17] casts the attack following a Markov decision process, where the attacker learns a policy to add or delete edges from the graph iteratively. In this framework, the state is defined by the current modified graph and the target node, while the reward, which is received only at the final step, reflects whether the attack has successfully induced a misclassification. The authors leveraged Q-learning with a hierarchical decomposition of the action space to showcase that we can efficiently navigate the combinatorial nature of graph modifications, significantly reducing computational complexity. Additional approaches have proposed to inject new nodes within the graph instead to propagate noisy representations [18], [19], [20], [21]. These methods, referred to as node injection attacks, introduce malicious nodes instead of modifying existing nodes or edges, thereby affecting the model's performance and showcasing.

In parallel to advancements in attack strategies, various defenses have been proposed to counter these vulnerabilities [13]. Overall, current adversarial defenses for graph data can be broadly categorized into two classes. The first, known as pre-processing methods [22], [23], aims to defend by filtering the input graph itself. The key idea is that adversarial modifications, such as the addition or deletion of edges, often alter the inherent semantic properties of the graph. By applying appropriate filtering scores, one can decide whether to disregard a given edge. An illustrative example is the GCN-Jaccard defense [23], which leverages the Jaccard similarity measure to assess the overlap between the feature sets of connected nodes. In this approach, edges connecting nodes with very low similarity (as indicated by a low Jaccard score) are considered suspicious and are removed prior to model training. Empirical findings have

shown that such pre-processing addition can significantly mitigate the impact of targeted adversarial attacks, thereby enhancing the robustness of graph convolutional networks without compromising their accuracy.

The second family of defense methods focuses on enhancing the robustness of a GNN through editing the internal workings. For instance, since GNNs are based on information aggregation within a neighborhood, by controlling the propagation, we can control the effect of the input perturbation. For instance, GNNGUARD [24] defends against attacks by dynamically reconfiguring the message propagation process by estimating the importance of each neighbor using similarity measures (such as cosine similarity) between node features, so that edges connecting similar nodes are given higher weights while those connecting dissimilar nodes are downweighted or pruned. This dynamic reweighting directly controls how information is aggregated, ensuring that adversarial noise has a minimal disruptive effect on the learned representations. In addition, the authors have proposed a layer-wise graph memory mechanism that smooths changes in these importance weights across successive layers, thus giving more stability to the propagation process even when the input graph has been perturbed. Empirical results show that this approach effectively restores model performance to levels comparable with clean graphs and outperforms many existing defense strategies. Another interesting example in this direction is the one provided by RGCN [25], which enhances the robustness of GCNs against adversarial attacks by fundamentally rethinking the hidden representations and message-passing mechanisms. Instead of representing nodes with fixed vectors, RGCN models each node's hidden representation as a Gaussian distribution. This design allows the network to naturally capture uncertainty, enabling it to absorb and mitigate the effects of adversarial perturbations by adjusting the variances. Moreover, RGCN introduces a variance-based attention mechanism that assigns lower weights to neighbors with high uncertainty, typically indicative of adversarial tampering, consequently preventing the harmful propagation of noisy or misleading information across the graph.

1.2 Research Challenges

As discussed in the previous section, the study of adversarial robustness in Graph Neural Networks (GNNs), encompassing both attack and defense strategies, has evolved into a distinct research field. Despite significant advances, several critical challenges remain unresolved. These challenges, spanning the spectrum from theoretical guarantees to practical deployment considerations, are pivotal for advancing the reliability of GNNs in safety-critical applications and are therefore the basis of the current thesis.

Research Challenge 1: Reconciling Strong Empirical Results with Provable Theoretical Guarantees

Many state-of-the-art defense methods, such as those that filter suspicious edges or adjust message-passing mechanisms, have shown promising empirical performance. However, these methods are primarily based on heuristic reasoning and lack rigorous theoretical underpinnings. This shortfall means that while they may effectively counter current adversarial attacks (e. g., those current state-of-the-art benchmarks such as Mettack), they could be vulnerable to future, more sophisticated strategies. The primary challenge is to develop defense frameworks that not only exhibit robust empirical performance but also provide certifiable guarantees of robustness across a broad range of adversarial behaviors.

Research Challenge 2: Understanding Attack Propagation in Rich Topological Structures

Graph-structured data possess unique topological characteristics, such as connectivity patterns and multi-hop relationships, that are not present in other domains like images or text. GNNs leverage these rich topological features through message-passing, which aggregates information from multi-hop neighborhoods. However, this exact mechanism can inadvertently amplify adversarial perturbations, allowing small, localized attacks to propagate and cause widespread degradation in performance. Although some approaches mitigate this by reweighting edges or incorporating noise at intermediate layers, a deeper theoretical and empirical understanding of how adversarial noise diffuses through complex, heterogeneous, and dynamic graphs is still lacking. Addressing this challenge is critical for designing defenses that effectively limit the spread of adversarial effects by taking into account the input graph's underlying topology.

Research Challenge 3: Incorporating Training Dynamics and Other Factors into Adversarial Robustness

Beyond the inherent structure of graphs, various training dynamics, such as weight initialization, the number of training epochs, and optimizer choices, play a crucial role in a model's generalization capabilities. These factors also significantly influence a model's vulnerability to adversarial attacks. For example, different initialization strategies can affect both the clean performance and the robustness of a GNN. Despite their importance, these aspects have often been overlooked in the context of adversarial robustness. A key research challenge, therefore, is to systematically integrate these training dynamics into the design and analysis of robust GNN architectures, providing insights into

how careful parameter selection can bolster a model's resilience with minimal additional cost.

Research Challenge 4: Balancing Robustness with Practical Feasibility

Many current adversarial defense methods incur substantial computational overhead, whether through intensive graph pre-processing, repeated computation of graph statistics, or costly retraining procedures under adversarial constraints. Such computational demands can be prohibitive for real-time or large-scale applications, including online social networks and mobile platforms. An essential research goal is to develop lightweight yet effective defense techniques that ensure robust performance without sacrificing speed or scalability. This requires the design of efficient algorithms and training protocols that can be seamlessly integrated into existing GNN architectures while maintaining competitive performance on both clean and adversarially perturbed graphs.

1.3 Research Objectives

1.3.1 Research Questions

Building on the research challenges outlined previously, this thesis seeks to advance the development of robust and reliable graph-based learning systems, particularly for safety-critical applications. Our aim is to address the need for provable robustness, deepening our understanding of adversarial propagation in complex topologies, incorporating critical training dynamics, and ensuring computational efficiency. We therefore reformulate the previous challenges as a set of specific, well-defined, and precise research questions that we aim to answer throughout this thesis:

- Q1 Topology and Propagation:** How does the input graph's topology influence the propagation of adversarial perturbations throughout the network?
- Q2 Theoretical and Practical Defenses:** Based on insights into adversarial propagation, can we develop novel defense methodologies that provide theoretical robustness guarantees while maintaining high performance and imposing minimal computational overhead?
- Q3 Training Dynamics and Robustness:** In what ways do training dynamics, such as initialization strategies, affect the final adversarial robustness of a model, and how can these factors be optimized to enhance resilience?

Q4 Adaptive Attack Strategies: Can we derive new adversarial attack methods that are capable of bypassing current defenses, thereby providing a more comprehensive understanding of the GNN’s vulnerabilities?

1.3.2 Thesis Contributions

The contributions in this thesis are all related to the study of the adversarial robustness of GNNs. Progress has been achieved in accordance with the outlined research questions, with various milestones reached throughout the study.

Figure 1.1 provides an overview linking the previously cited research questions with the corresponding contribution through the included papers in this thesis. Specifically, during the course of the thesis, the aim was to validate our main *research statement*, which we summarize as follows:

Research Statement

Given that GNNs rely on multi-hop message-passing, unlike modalities such as images, where data points are typically independent, the topology of the input graph is central to a model’s robustness. Consequently, understanding how adversarial attacks propagate through a graph’s connectivity is key to accurately assessing and enhancing the robustness of GNN-based classifiers.

In order to reach and confirm our research statement, a number of contributions have been proposed, which we present in the following points:

1. **C1 – Understanding the Topological Influence on Attack Propagation:** The first research question focuses on understanding how the graph’s topology affects the propagation of adversarial perturbations. In *Paper A* [26], we conduct an in-depth analysis within the context of Graph Convolutional Networks (GCNs). Our results show that a GCN’s adversarial robustness depends on two key factors: (i) the product of the weight norms of the message-passing layers, and (ii) the structure of the underlying graph, captured through normalized walks. Consequently, and in line with the intuition, densely connected graphs are more susceptible to adversarial manipulations, whereas sparser topologies tend to confine and mitigate the spread of adversarial noise. We additionally expanded the results on GCN to take into account Graph Isomorphic Networks (GIN), where we find a similar bound, with the difference that topology is represented through the node degrees. Specifically, if a graph has nodes that are highly connected, then these nodes can act as a “propagation-centers” of the attack, helping to propagate it easily and faster. Hence, by definition, a first conclusion is that when dealing with networks with

high individual degrees, it is easier to attack a GIN than a GCN. Overall, the insights provided in this paper provide important theoretical and empirical insights into why certain graphs or GNNs are inherently more robust against adversarial attacks.

2. **C2 – Designing Theoretically Grounded Defenses:** This research direction is divided into two complementary components: (i) developing defense methods that combine strong empirical performance with rigorous theoretical guarantees, and (ii) ensuring that these defenses incur only a minimal computational complexity overhead, making them practical for large-scale applications. In *Paper A* [26], we propose an adaptation of Graph Convolutional Networks (GCNs) that employs orthogonal (or nearly orthogonal) weight matrices to constrain the spectral norm growth. By iteratively projecting the weight matrices onto the orthogonal manifold after each forward pass, our method, denoted as GCORN, effectively bounds the maximum amplification of adversarial perturbations. The theoretical analysis in Paper A establishes a direct relationship between the product of weight norms and adversarial robustness, indicating that tighter control over these norms leads to a more resilient model. Empirical evaluations demonstrate that GCORN not only achieves higher accuracy under adversarial attacks compared to the original vanilla GCN and other defense benchmarks on both structural and node features attacks, but also maintains strong performance on clean data, which is a crucial metric in practice, since we do not know a priori if an input graph is corrupted or not.

Despite these promising results, the orthogonal projection process introduces a computational cost that may be prohibitive in specific real-time or large-scale scenarios, limiting the applicability of GCORN. Addressing this issue, *Paper B* [27] introduces NoisyGNN, a lightweight defense mechanism that injects controlled random noise into the hidden layers during training. As previously witnessed in Vision, adding noise or adversarial training has had great success in increasing a model’s robustness. Since we are operating in a discrete space (the graph is a set of edges), we consider the smoothing in the hidden representation space, which is continuous. Our theoretical analysis provides an upper bound on the model’s adversarial risk, showing that robustness is proportional to the amount of noise injected. However, because excessive noise can impair the model’s ability to learn meaningful representations, we should aim to find the trade-off between noise injection and clean-data performance. Empirical results confirm that NoisyGNN significantly improves resistance against various adversarial attack strategies while adding only a small computational overhead, mainly related to sampling from a Gaussian distribution,

thus making it an efficient solution.

Together, these two contributions offer a dual strategy for enhancing GNN robustness and tackling the considered research question. GCORN provides a mathematically grounded approach to control the sensitivity of the model through orthogonal weight constraints, while NoisyGNN delivers a computationally efficient alternative that achieves robustness by carefully balancing noise injection with model performance.

3. **C3 – On the effect of Training Dynamics:** The third research perspective is related to studying other parameters and factors that are not directly related to the architecture of the model, but that can have an effect on the final model’s underlying robustness. In *Paper C* [28], we provide a comprehensive theoretical framework that connects weight initialization strategies and the number of training epochs to a GNN’s adversarial robustness. To our knowledge, such theoretical studies have been missing from the literature, and such choices and their importance have previously been overlooked. Specifically, we prove that the model’s final vulnerability can be bounded in terms of its *initial weight norms* and the chosen number of training epochs, thereby establishing a direct link between initialization choices (e. g., Gaussian vs. orthogonal and others) and resilience under adversarial perturbations. Our analysis shows that while larger weight norms or longer training can often improve clean-data accuracy, they simultaneously degrade adversarial robustness, a phenomenon we confirm empirically on multiple real-world benchmarks. Notably, experiments reveal that certain initial distributions can yield up to a 50% difference in robustness when compared to less favorable initializations, all without sacrificing performance on clean inputs. Beyond GNNs, we further demonstrate the broader applicability of these insights by extending the same bounding techniques to general deep neural networks, consequently generalizing our contributions to the general community studying adversarial robustness.
4. **C4 – Adaptive Attack Strategies:** The fourth and final research question investigates the ability to generate new adversarial attacks that are capable of fooling available adversarial defense methods. In *Paper D* [29], we introduce a new perspective on adversarial attacks called *Unbound-Attack*, which shifts from the conventional, perturbation-based paradigm to a more flexible framework that generates adversarial graphs entirely “from scratch.” The central hypothesis in this work was that adversarial defenses adapt to limit the effect of a perturbation by pre-processing or other techniques to identify and disregard suspicious nodes; therefore, available adversarial attacks that consider a single point and perturb it are easily detectable. Additionally, the produced graph is not always valid,

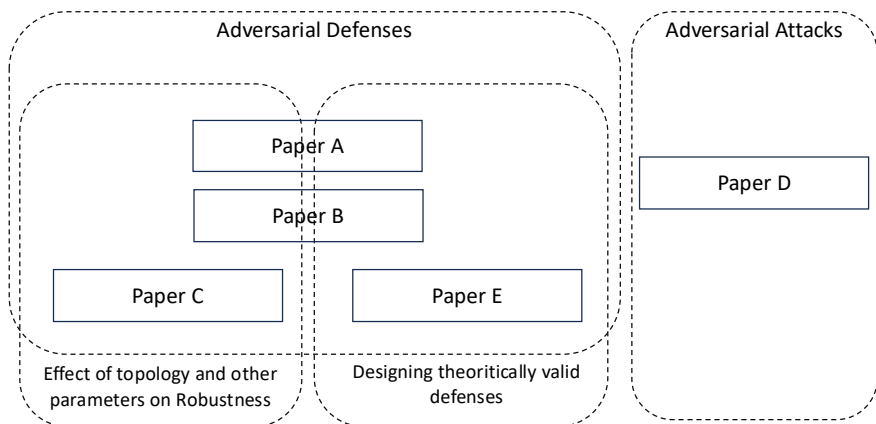


Figure 1.1: Overview of the considered research questions and the overlapping contributions of the included papers in this thesis.

specifically in domains in which editing an edge has a high cost. Consequently, and unlike classical attacks that operate under tight budgets (e.g., only a few edge or feature modifications) and often leave detectable traces, UnboundAttack synthesizes graphs with properties similar to those in the training data, such as comparable degree distributions or subgraph motifs, while fundamentally bypassing many detection methods that rely on identifying slight deviations from an original graph. This approach is particularly relevant for domains like molecular design, where adding or removing chemical bonds in an existing molecule can easily invalidate its realism or be caught by simple heuristics. By contrast, UnboundAttack leverages a generative adversarial model to produce chemically plausible but malicious molecular graphs that fool a victim GNN without relying on incremental perturbations. Our empirical results demonstrate that these unbounded adversarial examples consistently degrade the performance of popular GNN architectures (e.g., GCN and GIN) more effectively than state-of-the-art perturbation-based attacks. Consequently, *Paper D* underscores how unrestricted graph generation can expose vulnerabilities in GNNs that standard, budget-limited methods may overlook, emphasizing the necessity for defenses capable of addressing these broader and more elusive threat vectors.

1.4 Author Contribution

The author of this thesis is the primary contributor and first author of all the included research papers. The author has led all aspects of the work, including proposing all the methods, writing the papers, performing the experiments, and implementing the source code for each contribution. The co-authors of the papers have contributed to the development of the methods and assisted in the writing process.

1.5 Outline

The remainder of this thesis is organized as follows:

- **Chapter 2: Background** provides the necessary information, covering fundamental concepts and definitions relevant to the thesis. We start by introducing graph neural networks (GNNs), and we delve into topics and definitions regarding adversarial robustness with a focus on graph adversarial attacks. We additionally provide some elements about the research methodology, and specifically the experimental approach we have followed.
- **Chapter 3: Contributions** building upon the contents of Section 1.3, this chapter offers an extended summary of the contributions made in the context of adversarial defenses. It highlights the key contributions, methodologies, and findings presented in the research papers included in this thesis.
- **Chapter 4: Additional Contributions** In addition to the previous contributions, we present in this section contributions that are related to the adversarial attack side, and some additional contributions that do not necessarily fit within the considered research questions.
- **Chapter 5: Conclusions and Future Work** outlines potential directions for future research and provides a comprehensive conclusion to the thesis. It reflects on the implications of the findings, discusses limitations, and suggests areas for further exploration.
- **Appendix:** Following Chapter 4, the complete publications referenced in Section 1.3 are appended.

Chapter 2

Background

This chapter begins by introducing foundational concepts in representation learning, with a particular emphasis on Graph Neural Networks (GNNs), which form the core of this thesis. We then explore adversarial attacks tailored to graph-structured data, presenting the formal definitions and mathematical tools that underpin the methodologies developed in the following chapters.

Afterwards, we outline key aspects of the research strategy and experimental framework adopted in this work, highlighting the principles of reproducibility, reliability, and validity that support the robustness of our findings.

2.1 Graph Neural Networks

Since extending adversarial attacks to tasks such as unsupervised learning is not straightforward, our investigation focuses primarily on supervised classification, in line with the current research landscape. We begin by formalizing key concepts related to supervised learning, before narrowing our focus to methods based on Graph Neural Networks (GNNs).

2.1.1 Formalizing the Classification Problem

Let \mathcal{X} denote an input space from which data points $x \in \mathcal{X}$ are drawn (e.g., images, graph-structured data, or any other domain). In the standard supervised learning setup, we assume there is an output space \mathcal{Y} consisting of a finite set of classes, typically represented by $\{1, 2, \dots, K\}$ for some integer number of classes K . We consider that we are provided with a set of training samples $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. We assume these pairs are drawn *i.i.d.* from the original data distribution $D_{\mathcal{X}, \mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}$.

The aim of supervised learning is to learn a classifier, that is, a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ chosen from a family of functions \mathcal{F} , whose goal is to predict the label y of a new, unseen input x in a way that minimizes its misclassification error with respect to $D_{\mathcal{X},\mathcal{Y}}$. Formally, we define the risk of a classifier f as:

$$R[f_\theta] = \mathbb{E}_{(x,y) \sim D_{\mathcal{X},\mathcal{Y}}} \left[\mathbf{1}\{f(x) \neq y\} \right],$$

where $\mathbf{1}\cdot$ is the indicator function, which equals 1 when the model's prediction is incorrect and 0 otherwise. Equivalently, the quantity $R[f]$ represents the probability that $f(x)$ disagrees with the true label y under distribution $D_{\mathcal{X},\mathcal{Y}}$. Since $D_{\mathcal{X},\mathcal{Y}}$ is unknown, in practice we approximate this risk by minimizing an empirical loss on the training sample S :

$$\hat{R}[f_\theta] = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{f(x_i) \neq y_i\}$$

Typically, training involves minimizing this empirical quantity by selecting the parameters θ of the function f_θ from the chosen function class \mathcal{F} . This objective serves as an estimator of the actual risk, with the expectation that the model will exhibit similar performance on unseen test data. This notion is closely tied to the concept of generalization, a fundamental topic of interest in the machine learning literature.

2.1.2 Graph Neural Networks

Having introduced the standard supervised classification framework in the previous section, we now specialize these notions in the graph domain. In this setting, each input $x \in \mathcal{X}$ is a graph represented by both its structure (either adjacency or any other shift operator) containing information about the different links/edges between the nodes, by its node features which are individual information about each node and in some special cases edges features, which are information about the edges (such as the type of the link between different molecules). Consequently, the learning process must account for these different elements, taking into account the topological structure of the nodes, edges, and the different available node/edge features. In this aspect, Graph Neural Networks (GNNs) [6] have emerged as a powerful class of models precisely tailored to capture this relational structure.

Let $G = (V, E)$ be a graph where V is its set of vertices and E its set of edges. We will denote by $n = |V|$ and $m = |E|$ the number of vertices and the number of edges, respectively. Let $\mathcal{N}(v)$ denote the set of neighbors of a node $v \in V$, i. e., $\mathcal{N}(v) = \{u : (v, u) \in E\}$. The degree of a node is equal to its number of neighbors, i. e., equal to $|\mathcal{N}(v)|$ for a node $v \in V$. A graph

2.1. GRAPH NEURAL NETWORKS

is commonly represented by its adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, which encodes edge information. The (i, j) -th element of the adjacency matrix is equal to the weight of the edge between the i -th and j -th node of the graph, and a weight of 0 in case the edge does not exist. As previously mentioned, in some settings, the nodes of a graph might be annotated with feature vectors. We use $\mathbf{X} \in \mathbb{R}^{n \times D}$ to denote the node features, where D is the feature dimensionality. The feature of the i -th node of the graph corresponds to the i -th row of \mathbf{X} .

In this perspective, our function's input is a graph $G \in \mathcal{G}$, represented by its adjacency matrix $\mathbf{A} \in \mathcal{A}$ and its node features $\mathbf{X} \in \mathcal{X}$. Analogous to the standard supervised setup, we receive a training sample:

$$S = \{(G_1, y_1), (G_2, y_2), \dots, (G_m, y_m)\},$$

where each $G_i \in \mathcal{G}$ is associated with a class label $y_i \in \mathcal{Y} = \{1, 2, \dots, Y\}$ with Y being the number of classes which are drawn from the underlying distribution $D_{\mathcal{G}, \mathcal{Y}}$. Our goal is to learn a classifier $f_\theta : \mathcal{G} \rightarrow \mathcal{Y}$ from the set of GNN model families F_{GNN} such that f_θ achieves a low misclassification error with respect to $D_{\mathcal{G}, \mathcal{Y}}$. In particular, if $f_\theta(G)$ denotes the predicted class for graph G , then the population risk of graph classifier f_θ is:

$$R[f_\theta] = \mathbb{E}_{(G, y) \sim D_{\mathcal{G}, \mathcal{Y}}} [\mathbf{1}\{f_\theta(G) \neq y\}],$$

Moreover, the training process can be formulated as minimizing the empirical risk (or a differentiable surrogate loss):

$$\min_{f_\theta \in F_{GNN}} \sum_{i=1}^m \mathbf{1}\{f_\theta(G_i) \neq y_i\}.$$

The main idea of a GNN is that a node's representation is dependent on itself but also on its neighborhood, and iteratively on the neighbors of its neighbors. From this perspective, a GNN model consists of a series of neighborhood aggregation layers that use the graph structure and the nodes' feature vectors from the previous layer to generate new representations for the nodes. The GNN maintains a *node hidden embedding* $\mathbf{h}_v^{(t)}$ for each node v at each layer $t = 0, 1, \dots, T$, where T is the number of message-passing steps, initializing:

$$\mathbf{h}_v^{(0)} = \mathbf{x}_v,$$

where \mathbf{x}_v is the initial d -dimensional feature vector of node v . Each subsequent layer $t \in \{1, \dots, T\}$ updates these embeddings via two operations: *aggregation* and *update*.

Aggregation. In layer t , each node v collects “messages” from its neighbors $\mathcal{N}(v)$:

$$\mathbf{m}_v^{(t)} = \text{Agg}^{(t)}\left(\{\mathbf{h}_u^{(t-1)} : u \in \mathcal{N}(v)\}\right),$$

where $\text{Agg}^{(t)}$ is a permutation invariant function that maps the feature vectors of the neighbors of a node v to an aggregated vector. Typical choices include sum, mean, and max aggregation. The aggregated vector is passed along with the previous representation of v ($\mathbf{h}_v^{(t-1)}$) to the Update function, which combines those two vectors and produces the new representation of v .

Update. Having aggregated information from its neighborhood, node v updates its representation as follows:

$$\mathbf{h}_v^{(t)} = \text{Upd}^{(t)}(\mathbf{h}_v^{(t-1)}, \mathbf{m}_v^{(t)}),$$

where $\text{Upd}^{(t)}$ is typically implemented as a learnable transformation (e.g., a fully connected layer followed by a nonlinear activation). After T rounds, the final embeddings $\mathbf{h}_v^{(T)}$ encode the local T -hop neighborhood structure and features of each node.

Task Diversity in GNNs. While in the previous formulation we assumed that each graph G is associated with a label y , in reality, graphs offer a multitude of downstream tasks and a variety of supervised learning instances. In our thesis, we focus on the two following tasks:

- **Node Classification:** In this task, each node in a given graph is assigned a label. For instance, in citation networks like CORA or CITESEER, nodes represent papers and the goal is to classify them into research topics based on both their attributes and citation links. Here, the learned node embeddings $\{\mathbf{h}_v^{(T)}\}$ serve as features for a downstream classifier (often through a softmax layer) to predict the label for each node. One such example of ReadOut in the context of Node classification can be formulated as follows:

$$\hat{y}_v = \text{softmax}\left(\mathbf{W}^{(T)}\mathbf{h}_v^{(T)}\right),$$

where $\mathbf{W}^{(T)}$ is a trainable weight matrix mapping the final hidden state to a K -dimensional vector representing class probabilities for node v .

- **Graph Classification:** Unlike node classification, graph classification assigns a single label to an entire graph. This is common in applications

such as molecular property prediction, where each graph represents a molecule and the objective is to predict its biological activity or chemical properties. In these cases, after T iterations of neighborhood aggregation, to produce a graph-level representation, GNNs apply a permutation invariant readout function (for instance, sum operator, mean operator) to the feature vectors of all nodes of the graph as follows:

$$\mathbf{h}_G = \text{READOUT}\left(\{\mathbf{h}_v^{(T)} : v \in V\}\right),$$

from which the final prediction can be computed as follows:

$$\hat{y}_G = \text{softmax}\left(\mathbf{W}^{(\text{out})}\mathbf{h}_G\right).$$

In addition to the main formalism of message-passing that we provided previously, we illustrate this with two key concrete approaches that were the basis of our theoretical study aiming to understand their adversarial robustness. Specifically, we start with Graph Convolutional Networks (GCN) [3], which offer a simple and efficient mechanism for capturing neighborhood information through normalized aggregation, and Graph Isomorphic Networks (GIN) [4] that push the boundaries of representational power by incorporating learnable scaling and more complex update functions. These differences in architecture also impact their susceptibility to adversarial perturbations, a topic that will be explored in depth in subsequent chapters.

(i) Graph Convolutional Networks (GCN): The main idea of the model is that each layer aggregates feature information from a node’s neighborhood using a normalized adjacency matrix. The propagation rule of the model can be formally written for the t -th layer as:

$$H^{(t)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(t-1)}W^{(t-1)}\right),$$

where $\tilde{A} = A + I$ is the adjacency matrix with added self-loops, \tilde{D} is the diagonal degree matrix of \tilde{A} , $H^{(t-1)} \in \mathbb{R}^{n \times d_{t-1}}$ is the matrix of node embeddings from the previous layer, $W^{(t-1)}$ is a trainable weight matrix, and $\sigma(\cdot)$ is a nonlinear activation function (such as ReLU). The normalization $\tilde{D}^{-\frac{1}{2}}$ ensures that each node’s feature contributions are appropriately scaled, which helps mitigate issues arising from nodes with very high degrees. GCNs have become popular due to their simplicity and efficiency, yet their aggregation process can sometimes lead to over-smoothing, where distinct node features become too similar after several layers.

(ii) Graph Isomorphism Networks (GIN): To address some limitations of GCNs, particularly the tendency to over-smooth and lose discriminative power, the Graph Isomorphism Network (GIN) was proposed by Xu *et al.* [4]. GIN

aims to achieve maximal discriminative power, comparable to the 1-dimensional Weisfeiler-Lehman (1-WL) test for graph isomorphism. Its layer-wise update can be written as follows:

$$h_v^{(t)} = \text{MLP}^{(t)} \left((1 + \epsilon^{(t)}) \cdot h_v^{(t-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(t-1)} \right),$$

where $h_v^{(t)} \in \mathbb{R}^{d_t}$ is the representation of node v at layer t , $\epsilon^{(t)}$ is a learnable (or fixed) scalar that adjusts the relative importance of the node's own features versus those of its neighbors, and $\text{MLP}^{(t)}$ denotes a multi-layer perceptron applied at layer t . This formulation allows GIN to capture better subtle structural differences by preserving a more balanced mix of self-information and neighborhood aggregation. Empirical results have shown that GINs achieve state-of-the-art performance on tasks requiring fine-grained discrimination of graph structures, albeit with potentially higher computational costs due to the increased complexity of the MLPs.

Permutation Invariance. We note that an important property of the above message-passing procedure is that it is *permutation-equivariant*: relabeling the nodes of a graph in a consistent manner does not change the structure of the output embeddings, except for a corresponding reordering of indices. In graph-level tasks, a further *permutation-invariant* readout ensures that any reordering of nodes in V leads to the exact final prediction. This is very relevant in the context of adversarial attacks, since by definition, in the case where we aim to attack the structure, this latter change should be taken into account within the attack budget.

2.2 Adversarial Attacks

Adversarial attacks have recently emerged as a prominent topic in machine learning, initially in computer vision [11] and later extending to other modalities such as Natural Language Processing [30] and, more recently, to graph-structured data [12]. In this section, we provide an overview of adversarial attacks, starting with a motivating example and then formally defining various attack scenarios in the context of graphs.

2.2.1 Effect of Perturbations

Consider the well-known adversarial example in computer vision: a panda image that, after the addition of imperceptible noise, is misclassified as a gibbon by a deep neural network. This specific example showcases an input perturbation's

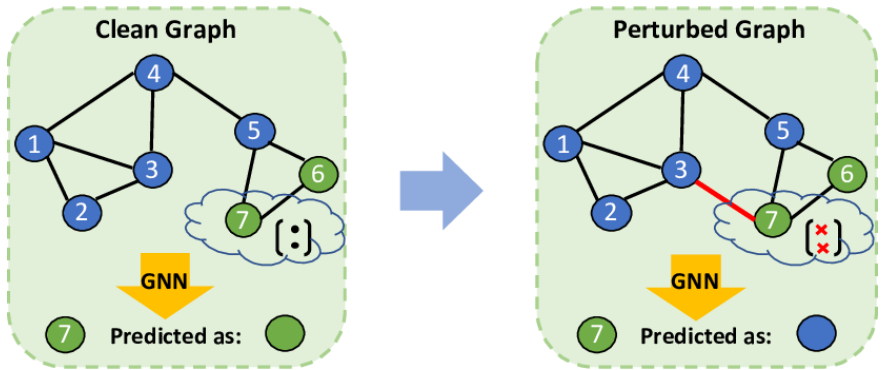


Figure 2.1: An Illustration of an adversarial attack on graph data. By adding a simple edge, the GNN's prediction on node 7 changed. Source: The illustration is borrowed from previous work [13].

ability to fool a model and lead to drastic changes in the output, with sometimes higher predicted confidence from the model. Analogously, in the context of graphs, small and strategically crafted changes to the graph structure or node features can mislead the GNNs. Figure 2.1 gives an illustration of such an effect, where adding a simple edge changes the model's prediction. For example, imagine a citation network where a single corrupted edge or a minor alteration in a paper's feature vector causes the network to classify a paper's research field incorrectly. Although these changes may be nearly invisible when viewing the entire network, their cumulative effect can severely degrade the model's performance. In general, adversarial attacks can be categorized based on the level of information the attacker has regarding the target model. We define three primary scenarios:

White-Box Attacks. In a white-box setting, the attacker has complete access to the target model's architecture, parameters, and gradients. This complete transparency allows the adversary to compute precise gradients with respect to both the node features and the graph structure. This access to the information makes the adversarial aim easier since the attack can continually adapt to the parts of the inputs that harm the model the most (through a gradient assessment, for instance). While less frequent in terms of real-world application (given that the majority of available applications do not allow it), assessing a model's vulnerability through these settings can give an idea of the worst-case adversarial robustness.

CHAPTER 2. BACKGROUND

Gray-Box Attacks. Gray-box attacks occur when the adversary has only partial knowledge of the target model. In such scenarios, the attacker might know the general architecture of the GNN or have access to a surrogate model that approximates the target's behavior. For example, in a recommendation system, an attacker may not have direct access to the internal parameters of the deployed GNN but may have information about the used training data and, therefore, recreate a surrogate model, which in turn can be leveraged to generate adversarial examples that are likely to transfer to the actual system. Techniques developed under this setting often rely on approximating gradients or other model outputs to inform the adversarial modifications.

Black-Box Attacks. The final setting, which is the most common and most important to study, is the black-box attack, where the adversary has minimal information about the target model, typically restricted to observing the model's output or prediction labels. Without access to internal gradients, the attacker must rely on query-based strategies, iteratively modifying the input graph based solely on the observed changes in the predictions. For example, in a social network analysis scenario, an external adversary might only have access to the final community detection or content moderation outputs. By systematically querying the model with slightly altered versions of the graph (e.g., by adding or removing edges) and monitoring the responses, the attacker can eventually infer which modifications cause misclassification, even though the internal workings of the model remain opaque.

In addition to the previous settings, there are two other settings that should be considered depending on when the attacker intervenes. Specifically, we can summarize them within the following two points:

Evasion Attacks. Evasion attacks are carried out during the test phase, where the adversary manipulates the input graph to force a misclassification without altering the training process. In this scenario, the model parameters remain fixed, and the attacker focuses solely on crafting adversarial examples that lead to incorrect predictions at inference time.

Poisoning Attacks. In contrast to evasion attacks, poisoning attacks occur during the training phase. Here, the adversary deliberately manipulates the training data by injecting carefully crafted perturbations into the graph structure or node features in order to compromise the learning process. This may involve adding, deleting, or modifying edges and features such that the re-trained model learns incorrect or biased representations, ultimately leading to poor performance on unseen data.

In the current thesis, we focus on evasion attacks, as we consider them to be the most relevant in real-world scenarios where the attacker can only access the model during inference time. We additionally focus on white-box adversarial attacks since, by definition, they give us the worst-case vulnerabilities. Therefore, by defending against this type of attack, we are sure to perform well in black-box settings.

2.2.2 Crafting the Attacks

In all the scenarios, the adversarial aim can be summarized as the problem of finding a perturbed graph \tilde{G} from an input graph $G \in \mathcal{G}$ that remains close to the original graph while inducing a different class from the attacker classifier. More formally, given a graph-based function $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$, and some input $(A, X) \in (\mathcal{A}, \mathcal{X})$ with its corresponding label $y \in \mathcal{Y}$ where $f(A, X) = y$, the goal of an adversarial attack is to produce a perturbed graph \tilde{G} and its corresponding features \tilde{X} which are ‘slightly’ different from the original input (G, X) such that the predicted class of (\tilde{A}, \tilde{X}) is different from the predicted class of (A, X) . From this perspective, defining closeness between the adversarial graph and the original input assumes the existence of a distance within our input space. In this aspect, we consider that our structural input space \mathcal{A} and the node features space \mathcal{X} are both measurable, and there exists a suitable distance metric (for instance, the number of modified edges) that could be used to quantify the similarity. We therefore introduce a distance over our input metric spaces, which takes both the graph and its corresponding features into account:

$$d^{\alpha, \beta}([G, X], [\tilde{G}, \tilde{X}]) = \alpha \|G - \tilde{G}\|_{\mathcal{G}} + \beta \|X - \tilde{X}\|_{\mathcal{X}}.$$

In practice, a graph is represented by its adjacency matrix (or some other graph shift operator) and its feature matrix. We can therefore, without loss of generality, consider the distance:

$$d_2^{\alpha, \beta}([A, X], [\tilde{A}, \tilde{X}]) = \min_{P \in \Pi} \{ \alpha \|A - P\tilde{A}P^T\|_2 + \beta \|X - P\tilde{X}\|_2 \}, \quad (2.1)$$

where α, β are hyper-parameters. Given the permutation invariance of GNNs, as previously discussed, we consider the distance with respect to the set of permutation matrices Π . We note that in the spatial case of unattributed graphs, the distance in 2.1 aligns with the commonly used edit distance on graphs, which is a measure of similarity between two graphs, quantifying the minimal number of edges that need to be edited to convert one graph into another while taking into account graph isomorphism. The proximity between the adversarially generated graph and the original input graph is controlled by the ‘‘attack budget’’, which is the maximum distance between the two latter

CHAPTER 2. BACKGROUND

elements. We formalize the attack neighborhood of an input graph G for an attack budget ϵ as:

$$B^{\alpha,\beta}(G, \epsilon) = \{(\tilde{A}, \tilde{X}) : d^{\alpha,\beta}([A, X], [\tilde{A}, \tilde{X}]) < \epsilon\}$$

Based on these previous elements, given the input graph G and the considered neighborhood corresponding to the attack budget ϵ , we can define the set of adversarial graphs as:

$$\tilde{G} = \{\tilde{G} \in B^{\alpha,\beta}(G, \epsilon) \mid f(A, X) \neq f(\tilde{A}, \tilde{X})\}$$

We note that the previous formulation is applicable to both structural adversarial attacks, aiming to perturb the adjacency, and node-features based adversarial attacks, which target the node features, and settings where both are attacked. According to the attacker's aim and desire to choose one part of the graph, the parameters α and β are set accordingly. One venue to formalize the adversarial aim is through an optimization problem of the form:

$$\begin{aligned} \min_{\tilde{G} \in B^{\alpha,\beta}(G, \epsilon)} \quad & d_2^{\alpha,\beta}([A, X], [\tilde{A}, \tilde{X}]) \\ \text{subject to} \quad & f(A, X) \neq f(\tilde{A}, \tilde{X}) \end{aligned}$$

This general formulation can be adapted in different ways to reflect the adversarial aim. In white-box settings, where the attacker has access to the model's gradients, this problem can typically be solved using classical gradient optimization such as PGD [31]. For instance, previous work *Mettack* [15] employs a bilevel formulation in which the graph structure itself is treated as a hyperparameter. Specifically, the meta-gradients are computed by unrolling the training procedure and backpropagating through the entire optimization process. This approach allows the attacker to identify the minimal edge insertions or deletions that will cause the most significant increase in the training loss (or decrease in the prediction margin) once the model has been retrained on the perturbed graph. From another perspective, *Nettack* [14] adopts a surrogate-based approach by leveraging a linearized version of a GCN to approximate the behavior of the target model, allowing it to operate in the black-box setting. In this framework, the attacker uses the surrogate model to calculate the influence of individual perturbations on the loss function, consequently identifying the minimal set of modifications targeting the edges or altering the node features that maximizes the surrogate loss. The provided optimization problem can also be framed from a Reinforcement Learning perspective. For instance, other approaches such as *RL-S2V* [17] recast the attack as a sequential decision-making process modeled by a Markov Decision Process (MDP). In RL-S2V, the adversary learns a policy to modify the graph iteratively, selecting actions that add or delete edges based on the immediate effect on the loss function, until

a misclassification is achieved. This reinforcement learning–based framework is instrumental in scenarios where gradients are unavailable or unreliable, as it relies on feedback from the model’s output rather than direct gradient information. For additional methods that explore these attacks, we refer the reader to previous work [32], which provides a general, comprehensive survey on the subject.

2.3 Research Methodology

This thesis focuses on investigating the adversarial robustness of various GNNs, with particular emphasis on models based on GCNs and GINs. To evaluate robustness, our methodology consists of two primary stages: first, training the models under standard supervised learning settings; and second, evaluating their robustness against adversarial perturbations. It is important to note that we focus on supervised learning settings, as the presence of labels is essential for defining adversarial attacks. Our experimental methodology is therefore structured around two main components: the training of the models and the evaluation of their adversarial robustness.

In general, Machine Learning (ML) and Deep Learning (DL) methods rely on numerical, structured data to train models. The experimental phase of such research typically involves data collection and pre-processing to ensure quality and relevance, followed by model training and evaluation. In supervised learning, model performance is commonly assessed using quantitative metrics such as accuracy, precision, recall, and F1-score. Researchers often focus on both fitting the training data and achieving strong generalization to unseen test data. This generalization capability is often assessed via cross-validation, where the dataset is partitioned into multiple folds. The model is iteratively trained on subsets of the data and validated on the remaining folds to mitigate overfitting and ensure robustness. Given that this thesis primarily focuses on adversarial robustness rather than model architecture innovation, we employ hyperparameter configurations consistent with those reported in the literature to ensure fair comparisons. For example, we use a 2-layer GCN architecture, which has been shown to achieve performance comparable to state-of-the-art methods across various benchmarks.

From the second perspective, evaluating adversarial robustness, our goal is to assess a model’s stability under input perturbations. In theory, this involves exploring all possible perturbations within a given neighborhood of an input point x , which is an intractably large and infinite space. One approach might involve randomly sampling perturbations and evaluating their effect on model predictions. However, this strategy requires assumptions about the underlying distribution of attacks, and random sampling may not capture worst-case perturbations effectively. Therefore, while many of our proposed defenses include

theoretical guarantees within defined attack neighborhoods or budgets, our empirical evaluations primarily assess performance against established, state-of-the-art adversarial attacks from the literature, such as Mettack, DICE [15], and PGD [31]. These methods are designed to approximate worst-case attacks, and while they cannot fully guarantee adversarial optimality, they offer meaningful insights into the practical robustness of the models. Nonetheless, we acknowledge that such empirical evaluation does not ensure robustness against future or unseen attack strategies, a broader challenge that falls under the domain of certified robustness, which is beyond the scope of this thesis.

2.3.1 Datasets

All experiments presented in this thesis are conducted using publicly available benchmark datasets, which offer multiple advantages. These include fostering transparency, enabling reproducibility of results, and facilitating consistent benchmarking across different methods. The primary task under consideration is node classification [33]. Specifically, we use citation networks such as Cora, CiteSeer, and PubMed, as well as the Co-author CS network [34]. We additionally use the arXiv Computer Science citation network from the Open Graph Benchmark (OGBN-Arxiv) [35], to showcase the effect of the proposed approach on larger graphs. Additionally, in some papers, to evaluate our methods on the graph classification task, we include results from datasets in the TU Dataset benchmark [36], particularly those from bioinformatics and cheminformatics domains (e. g., PROTEINS, NCI1, and D&D). Most of the datasets provide standard train/validation/test splits, which we adhered to in our experiments. In specific cases, such as the CS dataset, we followed the protocol from [37], randomly selecting 20 nodes per class for training and allocating 500/1000 nodes for validation and testing. For the graph classification task, we adopted the evaluation protocol proposed by [38], conducting 10-fold cross-validation using the predefined folds provided by the authors. This approach ensures fair comparison, reproducibility, and robustness in evaluating generalization performance.

2.3.2 Evaluation Metrics

To assess the empirical robustness of the models on the selected benchmark datasets, we use two interconnected evaluation metrics. The first is the attack success rate, defined as the proportion of input graphs for which an adversarial perturbation within the allowed neighborhood results in a change in classification. In other words, this measures how often an attack is successful in flipping the model's prediction. The second is the attacked accuracy, which quantifies the overall classification accuracy of the model when subjected to adversarial

2.3. RESEARCH METHODOLOGY

perturbations. In both cases, we only consider inputs that are correctly classified in the clean (unperturbed) setting, ensuring that our robustness evaluation is not confounded by baseline model errors.

When evaluating defense strategies, we also report the clean accuracy under defense, which measures the model's performance on unperturbed data when the defense mechanism is active. This metric is essential because, in real-world scenarios, we typically do not know in advance whether an input has been attacked. A good defense should thus strike a balance between maintaining clean accuracy and improving robustness under attack.

Chapter 3

Contributions - Adversarial Defenses

In this chapter, we present the core contributions of this thesis, which revolve around both quantifying the adversarial robustness of Graph Neural Networks (GNNs) and proposing effective defense mechanisms. We specifically address the first three research questions that explore how topology, model initialization, and training dynamics impact a GNN’s susceptibility to adversarial attacks. We start by formalizing a general framework for adversarial robustness on which we will build the rest of our contributions. Following this, we detail each contribution within each direction in dedicated sections aligned with the respective research questions.

3.1 Adversarial Risk and Vulnerability

As introduced in Section 2.2, adversarial attacks on graphs involve crafting perturbed versions of an input graph that remain close to the original in structure or features, but lead the classifier to make incorrect predictions. Let \mathcal{G} denote the space of graphs. Given an input graph $G \in \mathcal{G}$, the attacker aims to generate a perturbed graph \tilde{G} , within a defined neighborhood of G , yielding a different classification than the original one provided by the model.

We consider three metric spaces with associated norms: the adjacency matrices space $(\mathcal{A}, \|\cdot\|_{\mathcal{A}})$, node features space $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$, and output labels space $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}})$. Let \mathcal{D} be a probability distribution over $(\mathcal{A}, \mathcal{X}, \mathcal{Y})$, and let’s consider $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ to be a graph-based classifier. Given an input (A, X) with true label $y = f(A, X)$, we define a joint graph distance metric over adjacency and features as:

$$d_2([A, X], [\tilde{A}, \tilde{X}]) = \min_{P \in \Pi} \{ \|A - P\tilde{A}P^T\|_2 + \|X - P\tilde{X}\|_2 \}, \quad (3.1)$$

where Π denotes the set of permutation matrices, capturing the permutation invariance of graph representations. The ϵ -neighborhood of an input graph is

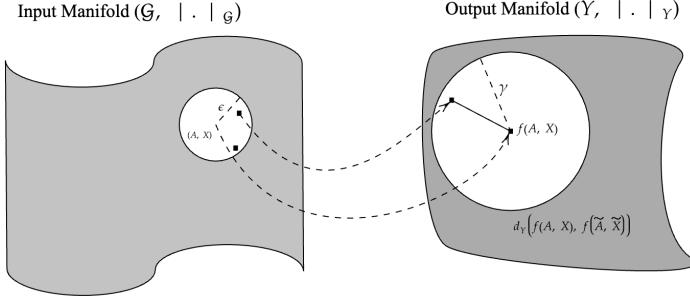


Figure 3.1: Illustration of bounding the adversarial vulnerability of a graph-based classifier f .

then defined as:

$$B_\epsilon(A, X) = \{(\tilde{A}, \tilde{X}) : d_2([A, X], [\tilde{A}, \tilde{X}]) < \epsilon\}.$$

Within this neighborhood, an adversarial example is one for which the classifier prediction changes, i. e., $f(\tilde{A}, \tilde{X}) \neq f(A, X)$. Since attackers typically do not have access to ground-truth labels, we define validity based on changes in model predictions. Based on the previous elements, we now formalize the expected adversarial risk of a graph classifier:

$$\mathcal{R}_\epsilon[f] = \mathbb{E}_{\substack{(A, X) \sim \mathcal{D} \\ (\tilde{A}, \tilde{X}) \in B_\epsilon(A, X)}} [d_Y(f(\tilde{A}, \tilde{X}), f(A, X))], \quad (3.2)$$

where d_Y denotes a suitable distance metric on the output space. Throughout our analysis, we use the ℓ_2 distance. A detailed discussion on metric equivalence is provided in Appendix A of Paper A.

Given that computing this expectation exactly is infeasible, we instead aim to derive an upper bound on $\mathcal{R}_\epsilon[f]$. Such bounds are sufficient and provide a principled way to estimate model vulnerability and form the theoretical backbone of our defense strategies. An illustration of this paradigm is provided in Figure 3.1, where we have an input space in which we consider a neighborhood, and then an output space in which we aim to measure the upper-bound behavior of the considered neighborhood. Typically, we want the vulnerability quantity to be minimal, such that we are sure that all points within the considered input's neighborhood have similar behavior based on the considered underlying classifier and shall therefore reflect the same classification. From this perspective, we introduce the notion of a GNN's robustness as follows.

Definition 1 (Adversarial Robustness). A graph-based function $f : (\mathcal{A}, \mathcal{X}) \rightarrow$

3.2. PAPER A – CONNECTING TOPOLOGY TO ADVERSARIAL ROBUSTNESS

\mathcal{Y} is said to be (ϵ, γ) -robust if:

$$\mathcal{R}_\epsilon[f] \leq \gamma,$$

with respect to the defined input and output distance metrics.

This formulation differs from the conventional "worst-case" robustness commonly studied in adversarial settings. Instead of focusing solely on the most damaging (worst) perturbation, our definition captures the average-case behavior over the neighborhood. We consider that this shift leads to a more comprehensive understanding of model robustness. Importantly, the following proposition bridges our average-case robustness to worst-case guarantees:

Proposition 2 (Average Implies Worst-Case Robustness). *If a graph function f is (ϵ, γ) -robust in the average sense, then it is also (ϵ, γ) -robust in the worst-case sense.*

The proof, provided in Appendix A of Paper B, ensures that our methodology generalizes and encompasses worst-case robustness evaluation, making it suitable for both theoretical analysis and empirical assessment.

3.2 Paper A – Connecting Topology to Adversarial Robustness

Having introduced a general framework for quantifying adversarial robustness, we now explore how the topology of the input graph influences a model's vulnerability. Unlike image data, graph data is characterized by its rich topology aspect, making it crucial to understand how structural factors impact robustness.

This investigation is the central focus of Paper A, where we derive theoretical bounds that relate a GNN's robustness to both the model parameters and the underlying graph structure. Specifically, we show that the adversarial vulnerability of a GNN is dependent on two key factors: (1) the norm of the model's weights and (2) the propagation dynamics within the graph, captured via normalized walks. The main result of Paper A is summarized in the following theorem, which addresses node feature-based attacks:

Theorem 3. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ be a GCN with L layers, where $W^{(i)}$ denotes the weight matrix of the i -th layer. For node feature perturbations with a budget ϵ , f is (ϵ, γ) -robust with:*

$$\gamma = \epsilon \cdot \left(\prod_{i=1}^L \|W^{(i)}\| \right) \left(\sum_{u \in \mathcal{V}} \hat{w}_u \right),$$

where \hat{w}_u denotes the total normalized walk weights of length $(L-1)$ originating from node u , and \mathcal{V} is the node set.

This result formally shows that the GCN's vulnerability is shaped by the interaction between the model's parameter norms and the connectivity of the graph. In particular, the more densely connected the graph (i.e., the more walks between nodes), the more an adversarial perturbation on a single node can propagate and affect predictions, resulting in a higher upper bound and thus lower robustness. On the other hand, sparsely connected graphs restrict information propagation, thereby limiting the impact of perturbations. This aligns with intuitive expectations and supports the view that graph sparsity can act as a form of implicit regularization against adversarial attacks.

While the primary focus of this analysis is node feature perturbations, which is a relatively underexplored area in the graph adversarial literature research, we extend the theoretical framework also to accommodate structural attacks, as described in the following result.

Theorem 4. *Let f be a GCN with L layers, and $W^{(i)}$ the weight matrix of the i -th layer. For perturbations targeting the graph structure (edges), f is (ϵ, γ) -robust with:*

$$\gamma = \left(\prod_{i=1}^L \|W^{(i)}\|_2 \right) \cdot \|X\|_2 \cdot \epsilon \cdot \left(1 + L \cdot \prod_{i=1}^L \|W^{(i)}\|_2 \right). \quad (3.3)$$

This bound captures the intuition that larger graphs (in terms of feature magnitude or node count) inherently offer more surface area for structural perturbations, increasing their susceptibility to attacks. Complete proofs of Theorems 3 and 4 are provided in Appendix C of Paper A.

We extend our analysis to GINs, which follow a different message-passing paradigm. The following result provides the corresponding upper bound for this case when subject to node feature-based adversarial perturbations.

Theorem 5. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ be a GIN with L layers (with $\zeta = 0$) and MLP weights $W^{(i)}$. Assume bounded node features with $\|X\|_2 < B$ and maximum node degree Δ_G . For node feature perturbations with budget ϵ , the model is (ϵ, γ) -robust with:*

$$\gamma = \left(\prod_{i=1}^L \|W^{(i)}\|_\infty \right) \cdot (B \cdot L \cdot \Delta_G + \epsilon).$$

This result highlights the role of node degree in GIN robustness. Nodes with high degree act as information hubs; perturbations to their features can affect many neighbors via the aggregation mechanism, making the model more vulnerable. Hence, robustness in GINs is closely linked to both feature magnitude and the graph's degree distribution.

3.2. PAPER A – CONNECTING TOPOLOGY TO ADVERSARIAL ROBUSTNESS

Table 3.1: Attacked classification accuracy (\pm standard deviation) of the models on different benchmark node classification dataset on feature based-adversarial attacks.

Attack	Dataset	GCN	GCN-k	AirGNN	RGCN	ParsevalR	GCORN
Random ($\psi = 0.5$)	Cora	68.4 \pm 1.9	69.2 \pm 2.6	73.5 \pm 1.9	71.6 \pm 0.3	72.9 \pm 0.9	77.1 \pm 1.8
	CiteSeer	57.8 \pm 1.5	62.3 \pm 1.2	64.6 \pm 1.6	63.7 \pm 0.6	65.1 \pm 0.8	67.8 \pm 1.4
	PubMed	68.3 \pm 1.2	71.2 \pm 1.1	70.9 \pm 1.3	71.4 \pm 0.5	71.8 \pm 0.8	73.1 \pm 1.1
	CS	85.3 \pm 1.1	86.7 \pm 1.1	87.5 \pm 1.6	88.2 \pm 0.9	87.6 \pm 0.6	89.8 \pm 1.2
	OGBN-Arxiv	68.2 \pm 1.5	52.8 \pm 0.5	66.5 \pm 1.3	63.8 \pm 1.9	68.3 \pm 1.9	69.1 \pm 1.8
PGD	Cora	54.1 \pm 2.4	58.3 \pm 1.6	68.2 \pm 1.8	62.5 \pm 1.2	68.6 \pm 1.7	71.1 \pm 1.4
	CiteSeer	52.3 \pm 1.1	59.6 \pm 1.6	59.3 \pm 2.1	61.9 \pm 1.1	62.1 \pm 1.5	65.6 \pm 1.4
	PubMed	66.1 \pm 2.1	67.3 \pm 1.3	70.8 \pm 1.7	69.5 \pm 0.9	68.9 \pm 2.1	72.3 \pm 1.3
	CS	71.3 \pm 1.1	74.1 \pm 0.8	76.3 \pm 2.1	76.6 \pm 1.2	77.3 \pm 0.6	79.6 \pm 1.2
	OGBN-Arxiv	67.5 \pm 0.9	49.9 \pm 0.7	55.7 \pm 0.9	63.6 \pm 0.7	67.6 \pm 1.2	68.1 \pm 1.1

3.2.1 Paper A - On the Effect of Orthonormal Weights

From our previous analysis, it is evident that the model’s robustness is highly linked to the norm of the weight matrices. Motivated by this insight, and in response to our second research question aiming to develop theoretically sound defenses, we propose a novel approach denoted Graph Convolutional Orthonormal Robust Networks (GCORNs).

GCORNs improve robustness by explicitly controlling weight norms through orthonormalization. Specifically, we enforce orthonormality of the layer weights during training via an iterative projection method introduced in [39], which maintains model expressiveness while encouraging stable gradient flow.

Given a weight matrix W from the message-passing scheme, we iteratively compute its orthonormal approximation \hat{W}_k via a Taylor expansion:

$$\hat{W}_{k+1} = \hat{W}_k \left(I + \frac{1}{2}Q_k + \dots + (-1)^p \binom{-1/2}{p} Q_k^p \right), \quad (3.4)$$

where $Q_k = I - \hat{W}_k^T \hat{W}_k$, and $p \geq 1$ is the chosen order of approximation. This process is differentiable and can be integrated directly into the training loop, enabling end-to-end learning of robust representations.

In addition to improved adversarial robustness, orthonormal weight matrices also mitigate exploding/vanishing gradients, a known issue in deep GNNs. From the convergence theory in [39], the process is guaranteed to converge when $\|W^T W - I\| \leq 1$, which we enforce via spectral norm scaling.

The experimental results demonstrate the effectiveness of GCORN in enhancing adversarial robustness across a wide range of attack settings. As presented in Table 3.1, GCORN consistently achieves higher performance under adversarial conditions compared to several competitive baseline methods. In particular, it improves upon the GCN-k [40] baseline by an average of approximately 12%

Table 3.2: Attacked classification accuracy (\pm standard deviation) of the models on different benchmark node classification datasets after the structural attacks.

Attack	Dataset	GCN	GCN-Jaccard	RGCN	GNN-SVD	GNN-Guard	ParsevalR	GCORN
Metattack	Cora	73.0 \pm 0.7	75.4 \pm 1.8	69.2 \pm 0.3	73.6 \pm 0.9	74.4 \pm 0.8	71.9 \pm 0.7	77.3 \pm 0.5
	CiteSeer	63.2 \pm 0.9	69.5 \pm 1.9	68.9 \pm 0.6	65.8 \pm 0.6	68.8 \pm 1.5	68.3 \pm 0.8	73.7 \pm 0.3
	PubMed	60.7 \pm 0.7	62.9 \pm 1.8	65.1 \pm 0.4	82.1 \pm 0.8	84.8 \pm 0.3	69.5 \pm 1.1	71.8 \pm 0.4
	CoraML	73.1 \pm 0.6	75.4 \pm 0.4	77.1 \pm 1.1	71.3 \pm 1.0	76.5 \pm 0.7	76.9 \pm 1.3	79.2 \pm 0.6
PGD	Cora	76.7 \pm 0.9	78.3 \pm 1.1	72.0 \pm 0.3	71.6 \pm 0.4	75.0 \pm 2.0	78.4 \pm 1.2	79.9 \pm 0.4
	CiteSeer	67.8 \pm 0.8	70.9 \pm 1.0	62.2 \pm 1.8	60.3 \pm 2.4	68.9 \pm 2.2	70.6 \pm 1.0	73.1 \pm 0.5
	PubMed	75.3 \pm 1.6	73.8 \pm 1.3	78.6 \pm 0.4	81.9 \pm 0.4	84.3 \pm 0.4	77.3 \pm 0.7	77.4 \pm 0.4
	CoraML	76.9 \pm 1.2	75.0 \pm 2.4	77.5 \pm 0.3	73.1 \pm 0.5	75.5 \pm 0.8	81.3 \pm 0.4	84.1 \pm 0.2

in classification accuracy when facing adversarial perturbations. Remarkably, in specific configurations, GCORN manages to recover the clean accuracy of the original GCN, effectively neutralizing the impact of the attack. Moreover, GCORN exhibits strong performance under structural attacks, outperforming robust architectures such as GCN-Guard and RGCN, both of which are considered state-of-the-art for this threat model. These findings highlight the general aspect and effectiveness of the proposed defense across different perturbation types. Additional empirical results can be found in the main paper.

Table 3.3: Performance of GCN and our proposed GCORN model, for different used approximation orders, on the Cora dataset.

	GCN	GCORN(1 ORD)	GCORN(2 ORD)	GCORN(3 ORD)
TRAINING TIME (IN S)	2.8 \pm 0.01	4.8 \pm 0.07	8.7 \pm 0.07	10.9 \pm 0.08
ACCURACY W/O ATTACK	79.2 \pm 1.6	78.8 \pm 1.3	79.8 \pm 0.9	80.8 \pm 1.1
ACCURACY W. ATTACK	68.4 \pm 1.9	77.1 \pm 2.1	78.3 \pm 1.1	78.6 \pm 0.4

Computational Considerations. Alongside its robustness improvements, GCORN introduces an additional computational cost due to its iterative orthonormalization scheme. As described in Equation 3.4, higher-order approximations (i. e., increasing p) yield more accurate projections at the cost of increased runtime. This introduces a trade-off between projection fidelity and computational efficiency, as illustrated in Table 3.3. The primary source of complexity arises from matrix multiplications, which scale cubically with the embedding size i. e., $\mathcal{O}(e^3)$ where e is the embedding dimension. Importantly, this complexity remains unaffected by the size of the input graph, unlike other defenses such as GNNGuard, whose complexity scales with the number of edges ($\mathcal{O}(e \times |E|)$), or GNN-SVD, which incurs $\mathcal{O}(n^3)$ due to its low-rank approximation. To quantify this in practice, we conduct a comprehensive runtime comparison. As shown in Table 3.4, GCORN achieves competitive training time across datasets, particu-

3.3. PAPER B - INJECTING NOISE AS A DEFENSE

larly on larger graphs, where its decoupling from graph size proves beneficial.

Table 3.4: Mean training time analysis (in s) of a our GCORN in comparison to the other benchmarks.

DATASET	GCN	GCN-K	AIRGNN	RGCN	GCORN
CORA	2.8	1.8	2.6	3.2	4.8
CITeseer	2.4	5.8	2.9	2.4	4.6
PUBMED	5.9	8.9	7.4	14.5	7.3
CS	6.1	12.1	12.4	13.8	15.5
OGBN-ARXIV	77.8	185.8	68.1	161.6	78.4

3.3 Paper B - Injecting Noise as a Defense

While the GCORN approach from the previous section provides strong robustness guarantees and empirical performance, its iterative orthonormalization process introduces additional computational overhead. This challenge is common among defense strategies, which often trade increased robustness for added complexity. To address this, and in line with our second research question, we propose a more lightweight defense mechanism that maintains competitive robustness with theoretical guarantees while significantly reducing computational cost.

Inspired by insights from adversarial training in computer vision, we investigate the effect of introducing randomness during training to enhance robustness. In image domains, adversarial training exposes the model to perturbed samples during learning, improving its resilience at test time. However, directly applying this strategy to graph data is non-trivial due to the discrete and combinatorial nature of graphs, which makes attack generation computationally expensive. To overcome this limitation, we propose a method, denoted as NoisyGNN, that injects stochastic noise into the hidden representations of the model during training, thereby simulating the effect of adversarial variability. The core idea is illustrated in Figure 3.2: by exposing the model to random noise in intermediate layers, we encourage it to learn smoother decision boundaries and reduce sensitivity to small perturbations during inference.

We start by analyzing the theoretical implications of noise injection. To do so, we model the GNN as a probabilistic mapping and treat the output space as a probability distribution. The robustness is quantified using the Kullback–Leibler

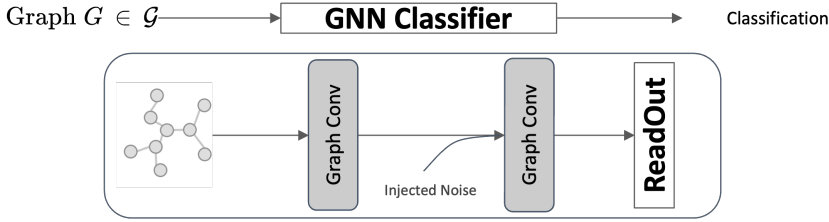


Figure 3.2: Illustration of the proposed NoisyGCN, which injects noise into hidden representations during training.

(KL) divergence, which serves as the output distance metric d_y in our general robustness definition (Equation 3.2).

In practice, various distributions can be used to generate the injected noise. However, for analytical tractability, we focus on the case of zero-mean Gaussian noise $\mathcal{N}(0, \beta I)$, where the variance β controls the strength of the noise. Under this formulation, we express the output of the model as:

$$f(\cdot) = \Phi^\ell \circ \dots \circ \Phi^{i+1} (\Phi^i \circ \dots \circ \Phi^1(\cdot) + T),$$

where $T \sim \mathcal{N}(0, \beta I)$ is a Gaussian random variable added to the hidden representation.

Theorem 6. *Let f be a two-layer graph classifier with 1-Lipschitz activation functions, and assume Gaussian noise with variance β is injected into the hidden layers. Then, under node feature perturbations with budget ϵ , f is (ϵ, γ) -robust with:*

- If f is GCN-based, then:

$$\gamma = \frac{(\|W^{(2)}\| \|W^{(1)}\| \epsilon)^2}{2\beta};$$

- If f is GIN-based, then:

$$\gamma = \frac{(\|A\| \|W^{(2)}\| \|W^{(1)}\| \epsilon)^2}{2\beta}.$$

The full proofs for both theorems are provided in Appendices B and C of Paper B. These theoretical results reinforce the hypothesis that noise injection, when appropriately calibrated, acts as an effective regularizer against adversarial perturbations in both feature and structure spaces.

3.3. PAPER B - INJECTING NOISE AS A DEFENSE

Table 3.5: Classification accuracy (\pm standard deviation) of the models when subject to structural perturbations through Mettack on different benchmark node classification datasets for different perturbation rates ϵ . The best accuracy in each setting, each dataset, and each model is typeset in bold.

Dataset	ϵ	GCN-Guard	GCN-Jaccard	GCN-SVD	RGCN	NoisyGCN
Cora	0%	77.5 \pm 0.7	80.9 \pm 0.7	80.6 \pm 0.4	83.5 \pm 0.3	83.2 \pm 0.4
	5%	75.8 \pm 0.6	78.9 \pm 0.8	78.4 \pm 0.6	78.3 \pm 0.6	81.2 \pm 0.7
	10%	74.7 \pm 0.4	76.7 \pm 0.7	71.5 \pm 0.8	70.7 \pm 0.8	74.5 \pm 0.6
CiteSeer	0%	70.1 \pm 1.5	71.2 \pm 0.7	70.7 \pm 0.4	72.3 \pm 0.5	71.9 \pm 0.4
	5%	69.9 \pm 1.1	70.3 \pm 2.3	68.9 \pm 0.7	70.6 \pm 0.7	72.3 \pm 0.6
	10%	70.0 \pm 1.5	67.5 \pm 2.1	68.8 \pm 0.6	68.7 \pm 1.2	70.4 \pm 0.8
PubMed	0%	84.5 \pm 0.6	85.0 \pm 0.5	82.7 \pm 0.3	85.1 \pm 0.8	85.0 \pm 0.6
	5%	84.3 \pm 0.9	79.6 \pm 0.3	81.3 \pm 0.6	81.1 \pm 0.7	81.8 \pm 0.4
	10%	84.1 \pm 0.3	67.4 \pm 1.1	81.1 \pm 0.7	65.2 \pm 0.4	73.3 \pm 0.6

Empirical Results. As shown in Table 3.5, the proposed NoisyGCN performs competitively in both clean and adversarial settings. Unlike many benchmark defense methods, NoisyGCN does not significantly compromise performance on clean data; in fact, it sometimes improves generalization. Under attack, NoisyGCN consistently outperforms defenses such as GNN-SVD, GNN-Jaccard, and RGNN, and shows robustness comparable to the more complex GNNGuard method. These results confirm that noise injection achieves a favorable trade-off between robustness and accuracy across multiple datasets and attack types. Additional experiments, including results on other attacks, are provided in the main paper.

Computational Efficiency. A major advantage of NoisyGCN is its minimal computational cost. The defense involves a simple forward-pass operation with added Gaussian noise, requiring only sampling from a standard distribution, an operation with negligible runtime overhead. Unlike methods such as GCN-SVD or GNNGuard, NoisyGCN does not require access to the full adjacency matrix or complex matrix factorization. As seen in Table 3.6, NoisyGCN scales efficiently with graph size and offers one of the most competitive time profiles among all tested defenses. In particular, the method shows significantly lower runtime than GNNGuard while maintaining comparable robustness.

Table 3.6: Mean training time analysis (in s) of the NoisyGCN in comparison to other baselines.

DATASET	GCN-GUARD	GCN-JACCARD	RGCN	GCN-SVD	NoisyGCN
CORA	28.52	1.93	1.16	1.39	1.29
CITeseer	36.04	1.58	1.23	1.12	1.24
PUBMED	731.26	12.27	34.19	4.60	2.41
POLBLOGS	18.17	5.17	0.96	0.80	0.65

3.4 Paper C – On the Effect of Training Dynamics

While developing the GCORN framework in Paper A, we observed an intriguing empirical phenomenon: changing the initialization of the model’s weights, while keeping all other factors constant, resulted in noticeable differences in adversarial robustness. Specifically, the accuracy of the model under attack would vary across different hyperparameters, despite similar clean performance. This observation led us to question how the training dynamics of a GNN, specifically, weight initialization and the number of training epochs, influence its robustness to adversarial perturbations.

Upon reviewing the literature, we found that this question has received limited attention, especially in the context of graph-structured data. Most existing studies focus on image domains and provide only empirical evidence while lacking strong theoretical explanations. In this paper, we aim to close this gap by offering both theoretical analysis and empirical validation of how training dynamics affect adversarial robustness in GNNs. Our goal is to understand whether factors such as the initial distribution of weights and the duration of training can increase or mitigate the vulnerability of GNNs to adversarial attacks.

We start by using our earlier robustness formulation (Definition 1) to explicitly account for training-dependent factors. The following theorem characterizes the adversarial risk upper bound for node feature-based attacks as a function of the initial and final model weights, as well as the number of training epochs.

Theorem 7. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ be a graph-based function composed of T GCN layers, where $W_0^{(i)}$ denotes the initial weight matrix of the i -th layer. For node feature-based adversarial attacks with budget ϵ , the function f is*

(ϵ, γ) -robust with:

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

where t is the number of training epochs, $W_*^{(i)}$ is the trained weight matrix of the i -th layer, and \hat{w}_u is the sum of normalized walks of length $T - 1$ originating from node u .

The computed bound reveals two important insights. First, the norm of the initial weights directly impacts the tightness of the robustness bound, with lower norms resulting in better robustness. Second, the number of training epochs t plays a significant role: increasing t can cause exponential growth in the bound, highlighting a trade-off between clean accuracy (which often improves with more training) and robustness (which may degrade). While one might consider initializing all weights to zero to minimize the bound, such a strategy is known to harm learning. As shown in prior work (e. g., see Page 301 in [41]), zero or constant initialization often prevents the model from learning meaningful representations, leading to suboptimal local minima due to poor gradient flow. We note that similar theoretical insights can be seen in the case of structural perturbations, where the adversarial attack modifies the graph's topology:

Theorem 8. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ be a GCN with T layers, initialized with weights $W_0^{(i)}$ and trained for t epochs to reach final weights $W_*^{(i)}$. For structural attacks with budget ϵ , f is (ϵ, γ) -robust with:*

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

As seen in paper A, in the case of structural perturbations, the effect of an attack is seen in each propagation step by altering the aggregation process via the perturbed adjacency matrix. Consequently, the effect of training dynamics is even more amplified than that of node feature attacks, reinforcing the importance of understanding initialization and training schedules.

To further quantify the impact of weight initialization, and to illustrate a practical setting for our theoretical analysis, we examine the case where initial weights are drawn from a Gaussian distribution. The following lemma provides an expectation bound over the distribution of initial weights:

Lemma 9. *Let f be a GCN initialized with weights drawn from a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. Then, under node feature perturbations, the expected adversarial risk is upper-bounded by:*

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

CHAPTER 3. CONTRIBUTIONS - ADVERSARIAL DEFENSES

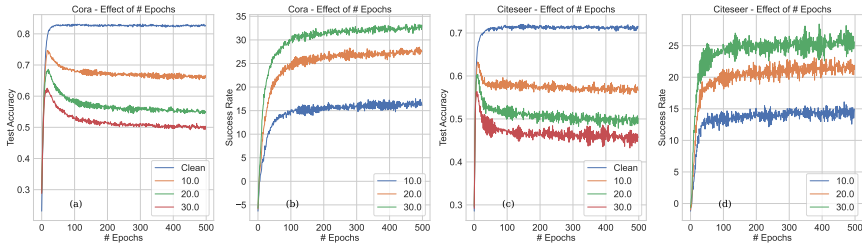


Figure 3.3: Effect of training epochs on the model's robustness on Cora (a,b) and CiteSeer (c,d). Clean accuracy increases and then plateaus, while attacked accuracy peaks early and then degrades.

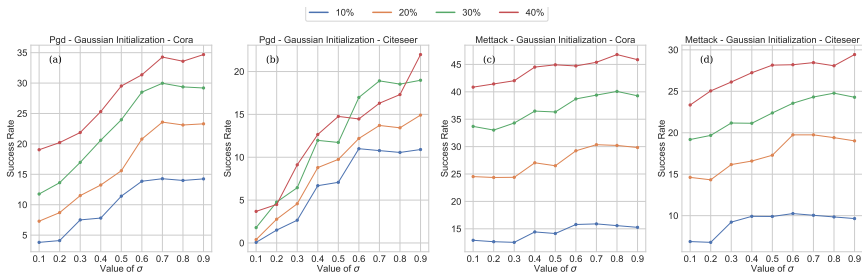


Figure 3.4: Effect of Gaussian initialization variance on adversarial robustness. Higher variance leads to higher success rates for PGD (a,b) and Metattack (c,d) on Cora and CiteSeer.

This result shows that increasing either the mean or the variance of the initialization distribution leads to looser robustness bounds. While reducing both could enhance robustness, it may also degrade clean performance, emphasizing the need for careful balancing as previously discussed.

We empirically validate our theoretical insights by analyzing the impact of training epochs on robustness. We train a 2-layer GCN on the Cora and CiteSeer datasets and assess its clean and attacked accuracy at each epoch. The attacks used include PGD and Metattack.

As shown in Figure 3.3, clean accuracy steadily improves and eventually saturates. However, the model's robustness, measured via the resulting attacked accuracy, initially improves then declines as training continues. This supports the theoretical result from Theorem 7, highlighting a trade-off: prolonged training can lead to better fitting but also increased adversarial vulnerability.

We afterwards analyzed the influence of the initialization distribution. We ini-

3.4. PAPER C – ON THE EFFECT OF TRAINING DYNAMICS

tialize the weights using a Gaussian distribution with fixed mean and varying variance σ^2 , and evaluate the success rate under this setup. As illustrated in Figure 3.4, increasing the variance of the initialization results in higher attack success rates, confirming the insights of Lemma 9. These experiments underscore the practical relevance of initialization: seemingly innocuous changes in variance can significantly impact the robustness of the final model. We note that further experiments exploring different initialization distributions and configurations are available in the paper.

Chapter 4

Other Contributions

In the following chapter, we summarize a contribution related to studying the adversarial robustness of GNNs, but rather from an attack perspective. We additionally provide some additional contributions that were investigated during the thesis and do not necessarily directly connect to any of the research questions.

4.1 Paper D - Adversarial Unbounded Attacks

As we have previously discussed, traditional adversarial attacks on graph neural networks (GNNs) are primarily constrained by small perturbations of existing graph inputs, modifying node features or graph structures with a predefined budget. While these bounded attacks have exposed significant vulnerabilities in GNNs, these methods often rely on hand-crafted perturbations (e.g., edge flips) that may produce unrealistic or invalid graphs in domains like chemistry or social networks. Furthermore, these attack strategies require costly optimization for each input graph and can be easily detected by the different defenses by heuristics or smoothing methods such as edge pruning or structure regularization. In this perspective, and in line with our last research question (Q4), we wanted to explore a fundamentally different and more general threat model: generating adversarial examples entirely from scratch.

In this context, we introduce a novel framework, referred to as UnboundAttack, which departs from the conventional assumption of starting from a real input graph. Instead, UnboundAttack generates new, synthetic graphs that share semantic characteristics with the data distribution, yet successfully mislead the target classifier. These examples are termed unbounded adversarial attacks, referring to the fact that they are not constrained to the neighborhood of a specific graph instance. UnboundAttack leverages recent advances

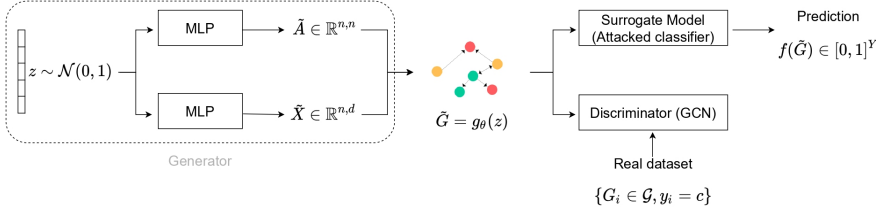


Figure 4.1: Illustration of the proposed framework UnboundAttack for generating unbounded adversarial attacks. The framework consists of three main components : (1) A targeted GNN model; (2) A generator consisting of two MLPs taking a sampled vector as input. (3) A classifier distinguishing between generated and real graphs.

in generative modeling, specifically Generative Adversarial Networks (GANs), to synthesize entirely new graphs that exhibit the statistical properties of the dataset (i.e., degree distribution, motifs, diameter), but are adversarial to the classifier.

The UnboundAttack architecture (illustrated in Figure 4.1) consists of three main components:

1. A **victim classifier** f (e.g., GCN or GIN), which is assumed to be pre-trained and fixed. The model is treated as a gray-box, allowing for white-box or surrogate-based black-box attacks.
2. A **generator** g_θ that maps a noise vector $z \sim \mathcal{N}(0, I)$ to a generated graph (\hat{A}, \hat{X}) , using two MLPs for adjacency and feature generation followed by discretization (via Gumbel-Softmax).
3. A **discriminator** d_ϕ , implemented using a GCN, trained to distinguish real graphs from generated ones, ensuring semantic realism.

The training objective of the generator is two-fold: (1) generate realistic graphs that match the training distribution (WGAN-GP loss), and (2) fool the classifier into misclassifying the graph (adversarial loss). The overall generator loss can be formulated as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{WGAN}} + \beta \cdot \mathcal{L}_{\text{Adv}},$$

where β controls the trade-off between realism and adversarial success. The adversarial loss encourages misclassification by maximizing the probability of predicting another class than the original prediction:

$$\mathcal{L}_{\text{Adv}}(z) = \text{ReLU} \left(0.5 - \max_{i \neq c} f(g_\theta(z))_i \right).$$

4.1. PAPER D - ADVERSARIAL UNBOUNDED ATTACKS

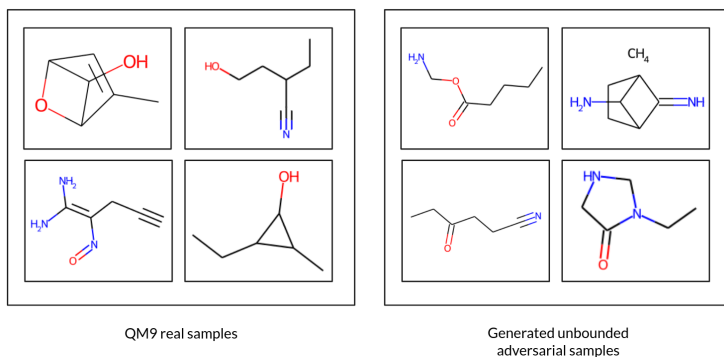


Figure 4.2: Examples of molecular graphs from the QM9 dataset (*left*). Examples of generated attacks (*right*). These examples have succeeded in misleading the classifier (i. e., $o(G) \neq f(G)$)

After training converges, the generator can produce an arbitrary number of unbounded adversarial graphs without querying the victim model, offering both scalability and generalization.

Empirical validation through experiments on the QM9 dataset, a benchmark of small organic molecules, where molecules are represented as graphs with atom and bond types, using different chemical properties, shows the validity of our proposed framework. Specifically, the generated attacks not only have the capacity to perturb the classification but are also realistic. Figure 4.2 illustrates real molecules (*left*) and generated adversarial examples (*right*). The generated graphs are chemically realistic yet successfully fool the classifier (i. e., $f(G) \neq o(G)$).

Overall, the proposed UnboundAttack framework opens a new line of inquiry for adversarial robustness in GNNs. Specifically, we have observed that UnboundAttack is not tied to a test input and can generalize to unseen targets. Furthermore, once trained, the generator can efficiently sample adversarial graphs in constant time, making it ideal for large-scale attack generation. We finally note that while the approach introduces some training overhead, this is amortized over the unlimited number of attacks that can be generated post-training.

4.2 Paper E - Conformalized Adversarial Detection

While most prior sections of this thesis focused on enhancing adversarial robustness through model-centric defenses, such as orthonormal weight constraints (GCORN - Paper A), noise injection (NoisyGCN - Paper B), these approaches generally aim to improve the resilience of GNNs under specific attack models. In contrast, rather than defending against attack, we wanted to investigate the ability to detect the attack and provide the user with a final confidence rate on the validity of the input. Specifically, our work introduces a model-agnostic detection framework that operates independently of the training pipeline and can be integrated with existing GNNs at inference time. We proceed by leveraging the conformal prediction (CP) paradigm to propose a general mechanism for identifying adversarially manipulated graphs by quantifying their conformity to the distribution of clean data.

Unlike defense strategies that operate by modifying the input graph or message-passing process, the proposed method, termed *Conformalized Adversarial Detection*, operates in a black-box setting, making no assumptions about the victim model's internal architecture. It is thus compatible with both white-box and black-box deployments and complements methods such as GCORN and NoisyGCN by adding an inference-layer security mechanism that does not interfere with training. The framework consists of two main components: a **(i)** victim GNN classifier and a **(ii)** non-conformity scoring model. The latter is trained on an enriched dataset combining the clean training set with synthetically generated adversarial examples (created using random Gaussian perturbations). This scoring model assigns a confidence score to new inputs, which are then calibrated via conformal prediction to produce a p-value indicating their similarity to the distribution of clean graphs. If the p-value falls below a user-defined threshold ϵ , the graph is flagged as an attack.

We empirically validated this framework on several benchmark datasets (MUTAG, PROTEINS, NCI1, D&D), demonstrating that it can accurately detect adversarial attacks generated via structural perturbations. Specifically, results show that detection accuracy ranges from 85% to 96% depending on the dataset and perturbation strength, with GCN and GIN both used as victim models. These results are competitive with, and in many cases exceed, the performance of other model-free defense techniques while requiring significantly less architectural modification. Moreover, the approach is adaptable to various types of perturbations, including node features and edge attributes, making it a strong candidate for real-world deployment where the attack type may not be known in advance. From another perspective, detection accuracy remains high even

4.2. PAPER E - CONFORMALIZED ADVERSARIAL DETECTION

for small perturbation budgets (e.g., $\sigma = 0.5$), which are traditionally the hardest to identify. Moreover, the method satisfies the statistical guarantees of conformal prediction, empirically verifying that the false positive rate remains bounded by the chosen confidence level.

To summarize, we see that Conformalized Adversarial Detection provides a complementary dimension to model robustness by introducing probabilistic inference-layer detection. Unlike GCORN or NoisyGCN, which embed defenses into the training dynamics, this method enables lightweight, model-agnostic defense integration and bridges robustness and uncertainty quantification. Together, these approaches offer a multi-layered security perspective for deploying GNNs in adversarial environments.

Chapter 5

Conclusions and Future Work

In this thesis, we investigated the adversarial robustness of Graph Neural Networks (GNNs) through both theoretical and empirical lenses, with the ultimate goal of advancing the design of reliable and secure graph-based learning systems. Grounded in the context of safety-critical domains, where the cost of erroneous predictions may be significant, we addressed four fundamental research questions that guided the structure and contributions of this work.

Q1: Topology and Propagation. Our first research question centered on understanding how the topology of a graph affects the propagation of adversarial perturbations. Through a formal adversarial robustness framework, we derived upper bounds that quantify the vulnerability of GNNs based on topological properties such as node degree and the density of connections. These bounds, expressed in terms of model weights and normalized graph walks, established a direct and interpretable relationship between a graph’s structure and the degree to which adversarial noise can propagate through the network. These findings demonstrated that denser graphs tend to amplify the effect of adversarial noise, while sparse or localized structures exhibit more robustness, a result with direct implications for both model design and graph preprocessing.

Q2: Theoretical and Practical Defenses. Building on these insights, we addressed our second question by developing defense mechanisms that not only perform well in practice but also come with theoretical guarantees. In particular, we introduced GCORN (Paper A), a defense framework based on enforcing orthonormality constraints on the weight matrices of GNNs. This approach explicitly targets the derived robustness bounds, reducing model sensitivity to perturbations while preserving clean performance. Complementing this, we proposed NoisyGCN, a defense method based on injecting Gaussian noise during training to regularize internal representations. Theoretical analysis showed that this noise injection yields provable robustness improvements, while empirical results highlighted its efficiency and adaptability. Importantly, both methods are complementary since the first one provides state-of-the-art results in terms

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

of defense while having an important complexity price, while the second provides good defense and also maintains low computational overhead.

Q3: Training Dynamics and Robustness. In exploring the role of training dynamics, we examined how initialization schemes and training duration influence a model's susceptibility to attacks. Theoretical derivations revealed that both the initial norm of the weights and the number of training epochs significantly impact the tightness of robustness guarantees. Empirical studies further validated that longer training may lead to better clean accuracy but reduced robustness, while higher initialization variance correlates with increased adversarial vulnerability. These results underscore the critical role of early training phases and call for robust-aware training strategies, such as constrained initialization or controlled early stopping, when deploying GNNs in adversarial settings.

Q4: Adaptive Attack Strategies. To thoroughly assess the robustness of defense methods and explore the full threat landscape, we introduced novel adaptive attack mechanisms. In contrast to traditional budget-constrained attacks, our UnboundAttack framework generates adversarial graphs from scratch using a generative modeling approach. By decoupling the attack from any particular input, this method exposes previously unexplored weaknesses in current defenses and reveals the limitations of perturbation-based robustness assumptions. Furthermore, it provides a valuable benchmark for stress-testing GNNs in open-world scenarios where adversaries are not constrained by the training distribution.

Beyond model-centric approaches, we also proposed a complementary framework for detection. Using conformal prediction, we developed a black-box framework to detect adversarial examples during inference by quantifying how "nonconforming" a graph is relative to clean data. This approach introduces a probabilistic layer of defense that is model-agnostic and calibrated, offering robust guarantees on false alarm rates. Unlike other defenses that modify the model or input, this method provides detection without interfering with the learning process, highlighting a new frontier in adversarial defense.

Taken together, the contributions of this thesis span the spectrum from theoretical guarantees to practical implementation, from defense to attack, and from robustness enhancement to adversarial detection. Each chapter builds on the foundational question of how graph structure, learning dynamics, and attack strategies intersect to shape the robustness of GNNs. The overall contributions in terms of defense methodologies are complementary, as illustrated in the diagram provided in Figure 5.1.

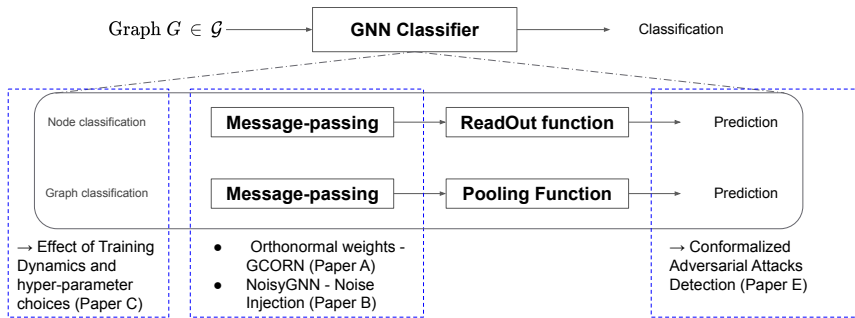


Figure 5.1: Diagram illustrating the contributions of the thesis in terms of adversarial defense on GNNs.

Future Work

As the adoption of GNNs continues to expand across a broad spectrum of applications, ensuring their reliability and trustworthiness under adversarial conditions will remain a central concern. The insights, methodologies, and theoretical tools developed in this thesis contribute to this broader goal by improving our understanding of adversarial vulnerabilities and offering scalable and principled defenses. However, numerous open questions and unexplored pathways remain. We hope that this thesis will not only serve as a solid foundation for future inquiry but also inspire the next generation of work on robust graph learning.

One of the most practically relevant and theoretically rich areas for future research is the trade-off between clean and adversarial performance. As explored in Paper C, training dynamics, including weight initialization and training epochs, have a profound influence on model robustness. However, quantifying this trade-off remains a largely unresolved challenge. In particular, the development of principled early stopping criteria or adaptive training schedules that optimize robustness without degrading clean performance is an area of high potential.

Another promising direction involves extending adversarial robustness considerations to all components of the GNN pipeline. While this thesis has rigorously addressed message-passing mechanisms, aggregation functions, and weight parameterizations, we haven't explored the role of readout and pooling functions. These functions play a critical role in summarizing local node-level embeddings into a global graph-level representation, especially in graph classification tasks. They represent a natural bottleneck where adversarial effects may either be am-

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

plified or suppressed. Therefore, designing pooling operations that are provably robust, or that can detect and limit the adversarial influence, may significantly enhance the security and interpretability of graph models.

On the attack side, while our proposed UnboundAttack framework revealed the risks posed by unrestricted adversaries, future extensions could aim to generalize these attacks to node classification and link prediction. Furthermore, incorporating generative models that are guided by physical or domain-specific constraints (such as diffusion or auto-regressive approaches) may enable the generation of more plausible yet still adversarial examples, especially important in domains like chemistry or social networks.

In summary, while this thesis offers foundational contributions in theoretical robustness, empirical defenses, and attack innovation, it also opens the door to numerous future research opportunities. Continued exploration of these directions will be essential for building the next generation of graph-based machine learning systems that are not only powerful but also safe, fair, and trustworthy.

Bibliography

- [1] OpenAI et al., *Gpt-4 technical report*, 2024. arXiv: 2303.08774. [Online]. Available: <https://arxiv.org/abs/2303.08774>.
- [2] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale”, in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=YicbFdNTTy>.
- [3] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks”, in *International Conference on Learning Representations (ICLR)*, 2017.
- [4] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?”, in *7th International Conference on Learning Representations*, 2019.
- [5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks”, in *ICLR*, 2018.
- [6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry”, in *International Conference on Machine Learning*, PMLR, 2017, pp. 1263–1272.
- [7] A. Chaudhary, H. Mittal, and A. Arora, “Anomaly detection using graph neural networks”, in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, IEEE, 2019, pp. 346–350.
- [8] A. Qabel et al., “Advancing antibiotic resistance classification with deep learning using protein sequence and structure”, *bioRxiv*, 2023. DOI: 10.1101/2022.10.06.511103. eprint: <https://www.biorxiv.org/content/early/2023/04/06/2022.10.06.511103.full.pdf>. [Online]. Available: <https://www.biorxiv.org/content/early/2023/04/06/2022.10.06.511103>.
- [9] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, “Molecular graph convolutions: Moving beyond fingerprints”, *Journal of Computer-Aided Molecular Design*, vol. 30, no. 8, pp. 595–608, 2016.

BIBLIOGRAPHY

- [10] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based Recommendation with Graph Neural Networks", in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 346–353.
- [11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples", in *International Conference on Learning Representations (ICLR)*, 2015.
- [12] S. Günnemann, "Graph neural networks: Adversarial robustness", in *Graph Neural Networks: Foundations, Frontiers, and Applications*, Springer, 2022, pp. 149–176.
- [13] W. Jin et al., "Adversarial attacks and defenses on graphs", *SIGKDD Explor. Newsl.*, vol. 22, no. 2, pp. 19–34, Jan. 2021, ISSN: 1931-0145. DOI: 10.1145/3447556.3447566. [Online]. Available: <https://doi.org/10.1145/3447556.3447566>.
- [14] 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 and Data Mining*, ACM, 2018.
- [15] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning", in *7th International Conference on Learning Representations*, 2019.
- [16] H. Zhan and X. Pei, "Black-box Gradient Attack on Graph Neural Networks: Deeper Insights in Graph-based Attack and Defense", *arXiv preprint arXiv:2104.15061*, 2021.
- [17] H. Dai et al., "Adversarial Attack on Graph Structured Data", in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1115–1124.
- [18] X. Zou et al., "Tdgia: Effective injection attacks on graph neural networks", in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD '21, 2021.
- [19] S. Tao, Q. Cao, H. Shen, J. Huang, Y. Wu, and X. Cheng, "Single node injection attack against graph neural networks", in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ACM, 2021.
- [20] Y. Chen et al., "Understanding and improving graph injection attack by promoting unnoticeability", *International Conference of Learning Representations*, 2022.

- [21] M. Ju, Y. Fan, C. Zhang, and Y. Ye, "Let graph be the go board: Gradient-free node injection attack for graph neural networks via reinforcement learning", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 4383–4390.
- [22] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. 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*, 2020, pp. 169–177.
- [23] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. 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*, 2019, pp. 4816–4823.
- [24] X. Zhang and M. Zitnik, "Gnnguard: Defending graph neural networks against adversarial attacks", *Advances in Neural Information Processing Systems*, vol. 33, pp. 9263–9275, 2020.
- [25] D. Zhu, Z. Zhang, P. Cui, and W. Zhu, "Robust graph convolutional networks against adversarial attacks", in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1399–1407.
- [26] Y. ABBAHADDOU, S. ENNADIR, J. F. Lutzeyer, M. Vazirgiannis, and H. Boström, "Bounding the expected robustness of graph neural networks subject to node feature attacks", in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=DfPtC8uSot>.
- [27] S. Ennadir, Y. Abbahaddou, J. F. Lutzeyer, M. Vazirgiannis, and H. Boström, "A simple and yet fairly effective defense for graph neural networks", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 21 063–21 071.
- [28] S. Ennadir, J. Lutzeyer, M. Vazirgiannis, and E. H. Bergou, "If you want to be robust, be wary of initialization", *Advances in Neural Information Processing Systems*, vol. 37, pp. 23 796–23 823, 2024.
- [29] S. Ennadir, A. Alkhatib, G. Nikolentzos, M. Vazirgiannis, and H. Boström, "Unboundattack: Generating unbounded adversarial attacks to graph neural networks", in *International Conference on Complex Networks and Their Applications*, Springer, 2023, pp. 100–111.
- [30] N. Carlini et al., "Poisoning web-scale training datasets is practical", in *2024 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2024, pp. 407–425.

BIBLIOGRAPHY

- [31] K. Xu et al., “Topology attack and defense for graph neural networks: An optimization perspective”, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.
- [32] W. Jin et al., “Adversarial attacks and defenses on graphs”, *SIGKDD Explor. Newsl.*, vol. 22, no. 2, pp. 19–34, Jan. 2021, ISSN: 1931-0145. DOI: 10.1145/3447556.3447566. [Online]. Available: <https://doi.org/10.1145/3447556.3447566>.
- [33] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data”, *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.
- [34] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, “Pitfalls of graph neural network evaluation”, *Relational Representation Learning Workshop (R2L 2018), NeurIPS*, 2018.
- [35] W. Hu et al., “Open graph benchmark: Datasets for machine learning on graphs”, *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 118–22 133, 2020.
- [36] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, “TUDataset: A collection of benchmark datasets for learning with graphs”, *Graph Representation Learning and Beyond (GRL+), ICML Workshop*, 2020.
- [37] Z. Yang, W. Cohen, and R. Salakhudinov, “Revisiting semi-supervised learning with graph embeddings”, in *International Conference on Machine Learning*, PMLR, 2016, pp. 40–48.
- [38] F. Errica, M. Podda, D. Bacciu, and A. Micheli, “A Fair Comparison of Graph Neural Networks for Graph Classification”, in *8th International Conference on Learning Representations*, 2020.
- [39] Å. Björck and C. Bowie, “An iterative algorithm for computing the best estimate of an orthogonal matrix”, *SIAM Journal on Numerical Analysis*, pp. 358–364, 1971.
- [40] M. E. A. Seddik, C. Wu, J. F. Lutzeyer, and M. Vazirgiannis, “Node feature kernels increase graph convolutional network robustness”, in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2022, pp. 6225–6241.
- [41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.