ADVERSARIAL ANALYSIS OF SIGNED GRAPHS WITH BALANCE THEORY

JIALONG ZHOU

MPhil

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Computing

Adversarial Analysis of Signed Graphs with Balance Theory

Jialong Zhou

A thesis submitted in partial fulfillment of the requirements for
the degree of Master of Philosophy
August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Jialong Zhou

Abstract

Signed graphs have emerged as effective models for capturing positive and negative relationships in social networks. To analyze such graphs, signed graph neural networks (SGNNs) have been widely employed, leveraging the unique structural characteristics of signed graphs. However, it is surprising to discover that the balance theory, which is commonly integrated into SGNNs to effectively model positive and negative links, can unintentionally serve as a vulnerability, susceptible to exploitation as a black-box attack. In this study, we introduce a novel black-box attack termed balance-attack, specifically designed to diminish the balance degree of signed graphs. To address the associated NP-hard optimization problem, we propose an efficient heuristic algorithm.

Furthermore, combating various adversarial attacks on signed graphs has become an urgent concern. We observe that these attacks often result in a reduction of the balance degree in signed graphs. Similar to the restoration of unsigned graphs through structural learning, we propose balance learning techniques to improve the balance degree of compromised graphs. However, we encounter the challenge of "Irreversibility of Balance-related Information", wherein the restored edges may not align with the original targets of the attacks, leading to suboptimal defense effectiveness. To overcome this challenge, we present a robust SGNN framework called Balance Augmented-Signed Graph Contrastive Learning (BA-SGCL), which integrates Graph Contrastive Learning principles with balance augmentation techniques. This approach facilitates the attainment of a high balance degree in the latent space, indirectly addressing the

challenge of "Irreversibility of Balance-related Information". We extensively evaluate our proposed balance-attack and robust BA-SGCL on multiple popular SGNN models and real-world datasets. The experimental results validate the effectiveness of balance-attack and the resilience of BA-SGCL. This research significantly contributes to enhancing the security and reliability of signed graph analysis within the context of social network modeling.

Publications Arising from the Thesis

1. Jialong Zhou, Xing Ai, Yuni Lai, Tomasz Michalak, Gaolei Li, Jianhua Li and Kai Zhou, "Adversarial Robustness of Link Sign Prediction in Signed Graphs", manuscript under submition, 2024.

2. Jialong Zhou, Yuni Lai, Jian Ren and Kai Zhou, "Black-Box Attacks against Signed Graph Analysis via Balance Poisoning", in *11th International Conference on Computing, Networking and Communications (ICNC)*, pp. 530-535, 2024.

Other Publications during My MPhil Study

1. Xing Ai, Jialong Zhou, Yulin Zhu, Gaolei Li, Tomasz P Michalak, Xiapu Luo and Kai Zhou, "Graph Anomaly Detection at Group Level: A Topology Pattern Enhanced Unsupervised Approach", in *40th IEEE International Conference on Data Engineer (ICDE)*, 2024.

2. Yuni Lai, Jialong Zhou, Xiaoge Zhang and Kai Zhou, "Towards Certified Robustness of Graph Neural Networks in Adversarial AIoT Environments", in *IEEE Internet of Things Journal* (2023).

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Kai Zhou, for his invaluable guidance and support throughout my MPhil research journey. His expertise, patience, and unwavering commitment to academic excellence have been instrumental in shaping this thesis. I am truly grateful for his mentorship and the countless hours he dedicated to reviewing my work, providing insightful feedback, and pushing me to reach my full potential.

I would also like to extend my appreciation to the other faculty members of our university who have contributed to my academic development. Their knowledge, encouragement, and constructive criticism have played a crucial role in shaping my research and enriching my understanding of the subject matter. I am grateful for their willingness to share their expertise and for the stimulating discussions that have expanded my horizons.

I am deeply indebted to my fellow colleagues and friends in the laboratory. Their collaboration, camaraderie, and intellectual exchange have created a supportive and inspiring environment for research. I am thankful for their friendship and encouragement to lend a helping hand whenever needed. Together, we have overcome challenges, celebrated successes, and made lasting memories that will forever be cherished.

Last but not least, I would like to express my heartfelt gratitude to my family and friends. Their unwavering love and encouragement in my abilities have been the driving force behind my pursuit of the MPhil program. Their constant support and

understanding have been invaluable throughout this journey. I am truly blessed to have them by my side, and I dedicate this achievement to them.

Finally, I want to take a moment to acknowledge and thank myself. Throughout this MPhil journey, I encountered numerous obstacles and faced moments of doubt. However, I demonstrated resilience and perseverance to navigate through those difficulties. It is through my own unwavering commitment and self-belief that I was able to overcome challenges and accomplish the goals I set for myself.

In conclusion, the completion of this thesis and my MPhil journey would not have been possible without the guidance, support, and encouragement of my supervisor, faculty members, laboratory colleagues, family, and friends. Their contributions have left an indelible mark on my academic and personal growth, and I am forever grateful for their presence in my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human relationships encompass a wide range of connections, including both positive interactions such as liking and trust, as well as negative associations such as distrust and dislike. In order to represent these relationships, social networks are often abstracted as graphs. However, conventional graph structures are unable to simultaneously capture positive and negative edge relationships. To address this limitation, signed graphs have gained popularity by assigning corresponding signs (+/-) to the edges. Various online platforms, such as Slashdot, Bitcoin Alpha, Bitcoin OTC, and e-commerce sites, generate signed graphs through user tagging, rating, and reviewing systems. The analysis of signed networks, including tasks such as link sign prediction and node ranking, has been greatly influenced by machine learning techniques. In recent years, researchers have focused on network representation learning for signed graphs, aiming to learn low-dimensional representations of nodes for downstream network analysis tasks. Graph neural networks (GNNs), as deep learning-based methods operating on graphs, have gained attention due to their promising performance. However, the presence of negative edges in signed graphs introduces challenges to the standard message-passing mechanism, necessitating the development of new GNN models specifically designed for signed graphs, known as signed graph neural net-

works (SGNNs).

Despite the success of SGNNs, there has been limited research focusing on adversarial attacks specifically targeting signed graphs or SGNNs. Adversarial attacks refer to malicious attempts to manipulate a system, leading to misidentification or misclassification. In the context of signed graphs, attackers can disrupt relationships by manipulating a subset of the edge connections. These attacks can significantly impact the performance of SGNNs, potentially deteriorating social relationships. The implications of such attacks are particularly significant in various real-world scenarios. In e-commerce platforms like Taobao and Amazon, signed networks naturally form between users and merchants through rating systems. Malicious users might deliberately give false negative ratings, damaging merchant reputations and business relationships. This represents a form of adversarial attack that can significantly impact business operations and trust within the platform. Similarly, in cryptocurrency trading platforms such as BitcoinAlpha and BitcoinOTC, user trust networks are crucial for secure transactions, where manipulated relationship signals could lead to significant financial risks. Social media platforms like WhatsApp, Instagram, and WeChat present another critical domain where signed networks play a vital role. In these platforms, user relationships form complex signed networks based on positive and negative interactions. Malicious actors might spread false information about relationships or misrepresent their connections with others (e.g., pretending to have positive relationships while harboring negative intentions), which can destabilize social structures and compromise platform security. These attacks, when manifested in signed graphs, can have far-reaching consequences for social cohesion and user trust. Existing adversarial attack approaches for normal graphs are not suitable for signed graphs, necessitating the development of new attack methods. Most SGNN models rely on balance theory, which suggests that signed triangles should have an even number of negative edges. Balance theory suggests that a balanced state occurs when two individuals either like or dislike each other, while an imbalance arises when there are

mixed sentiment relations. In triadic relations, balance is achieved when the algebraic multiplication of signs in the triad is positive. Empirical studies have confirmed that real-world signed graph datasets adhere to these conditions. Existing models incorporate balance theory in their loss functions or aggregation strategies to learn a new signed graph with a high balance degree. By integrating balance theory into SGNNs, models like SGCN and SNEA adopt a two-part representation and a more involved aggregation scheme. For instance, when considering a node, the positive part of its representation can aggregate information from the positive representations of its positive neighbors and the negative representations of its negative neighbors. However, the reliance on balance theory also opens up opportunities for attacks. Through pilot experiments, we found that no matter what the goal of the existing limited number of attacks on signed graphs is (maybe to reduce the test results of the training set), their common impact is to reduce the balance degree of signed graphs.

Based on this finding, we propose a novel black-box attack called "balance-attack" that targets the vulnerability of SGNNs by reducing the balance degree. By developing an approximative algorithm to manipulate balance, our proposed attack proves to be effective in compromising the robustness of SGNNs.

Furthermore, the adversarial robustness of SGNNs has received limited research attention. Adversarial robustness refers to a network's ability to resist small perturbations that can lead to misclassification or incorrect results. To ensure the reliability of SGNNs, it is crucial to develop models that can defend against adversarial attacks. Existing approaches, such as RSGNN, focus on incorporating structure-based regularizers to reduce vulnerability to input noise but do not specifically address adversarial robustness. Based on previous experiments, we propose the utilization of balance learning as an effective approach to enhance adversarial robustness in SGNNs. This approach draws inspiration from the widely used method of structural learning in normal graphs. When normal graphs are subjected to poisoning attacks, they often experience a significant decrease in homophily. By applying structural learning tech-

niques, the original balance of the graph can be successfully restored. In the context of signed graphs, we introduce a novel concept called balance learning, which follows a similar rationale. Adversarial attacks on signed graphs typically result in a low balance degree, indicating an imbalance in positive and negative edges. Our aim is to restore the balance of the graph by leveraging balance theory principles. By increasing the balance degree, we strive to recover a more balanced and representative graph that accurately captures social relationships.

However, we found that balance learning alone does not inherently confer robustness to the model. Our analysis reveals that while balance learning effectively enhances the balance degree of the resulting graph, it does not restore the original signs of the edges. This presents a significant challenge, termed the "Irreversibility of Balance-related Information." In other words, the original sign information of the edges is lost during the balance restoration process, making it difficult to fully recover the structural and sign information of the clean graph through balance learning alone.

To defend against adversarial attacks and tackle the aforementioned challenge, we propose a novel robust SGNN model named *Balance Augmented-Signed Graph Contrastive Learning* (BA-SGCL), which builds upon the Graph Contrastive Learning (GCL) framework to *indirectly* tackle the challenge of Irreversibility of Balance-related Information. Specifically, we consider the graph obtained by enhancing the balance degree as the positive view, while the negative view corresponds to the original input graph. To perturb the positive view, we utilize the balance degree as a guiding factor to shape the Bernoulli probability matrix within a learnable augmenter. Due to the Irreversibility of Balance-related Information challenge, it is difficult to recover the structural and sign information of the clean graph through balance learning. Our method makes the final node embeddings in latent space characterized with a high balance degree by contrasting positive and negative views. By maximizing the mutual information between the embeddings of two views and that between the embeddings and labels, our approach can achieve defend against attacks and improve prediction

accuracy.

The major contributions of our paper can be summarised are as follows:

- We introduce a novel black-box attack for signed graph neural networks by corrupting the balance degree. Also, we propose an effective and efficient algorithm to reduce the balance degree of signed graphs, a problem that has been proven to be NP-hard [14].
- We conduct a comprehensive theoretical analysis of attacks targeting signed graph analysis, shedding light on the fundamental nature of these attacks from an information theoretical perspective.
- We propose a novel robust model BA-SGCL based on the graph contrastive learning framework. We also present the theoretical reasoning of why our model can effectively combat attacks.
- Our extensive experiments provide compelling evidence for the effectiveness and generality of our proposed balance-attack. Furthermore, the experiments conducted on our BA-SGCL model consistently outperform other baseline methods when subjected to different types of adversarial attacks on signed graphs.

Chapter 2

Related Work

Extensive research has been conducted in the machine learning and security communities to explore adversarial attacks across different types of models. While naturally occurring outliers in graphs present certain challenges, adversarial examples are intentionally crafted to deceive machine learning models with unnoticeable perturbations. GNNs are particularly susceptible to these small adversarial perturbations in the data. As a result, numerous studies have focused on investigating adversarial attacks specifically targeted at graph learning tasks.

Bojchevski et al. [4] propose poisoning attacks on unsupervised node representation learning or node embedding, leveraging perturbation theory to maximize the loss incurred after training DeepWalk. Zugner et al. [104], on the other hand, tackle the inherent bi-level problem in training-time attacks by employing meta-gradients, effectively treating the graph as a hyper-parameter to optimize.

However, it is important to note that previous studies have predominantly focused on unsigned graphs, with limited research addressing adversarial attacks on signed graphs. While Godziszewski et al. [23] introduced an approach for attacking sign prediction, where attackers aim to conceal target link signs by manipulating non-target link signs, their method was not specifically designed for SGNN models. The

Fairness and Goodness Algorithm (FGA), despite being a widely-adopted trust system in signed social networks, has been shown vulnerable to the vicinage-attack method [5], which formulates the attack as a combination optimization problem. This method mines candidate attacking edges through perturbation space construction and polymorphic strong tie inference, effectively reducing target nodes' trust scores and outperforming baseline approaches. However, existing attack methods, including vicinage-attack, primarily operate in white-box settings. To our knowledge, no research has yet explored black-box adversarial attacks specifically designed for signed graphs.

Extensive research has been devoted to studying the robustness of graph learning models [45] [68] [100] by exploring various attack and defense methods. Recent studies [23] [97] [101] have also explored the vulnerabilities of signed graph analysis models. For instance, balance-attack [97] can effectively attack SGNNs in a black-box manner by decreasing the balance degree of signed graphs. Unfortunately, SGNNs currently lack strong defense mechanisms to effectively counter such attacks. RSGNN [91] is considered as a leading robust model that enhances robustness by integrating structure-based regularizers. However, while RSGNN excels at handling random noise, its ability to defend against adversarial attacks produces only average results.

Our approach in this paper builds upon Graph Contrastive Learning (GCL) [62] [67], which aims to maximize correspondence between related objects in a graph while capturing their invariant properties, enabling models to learn more invariant and generalized node representations. One key step in GCL is to define positive and negative views for contrastive pairs. One common approach uses graph augmentation to generate multiple views for flexible contrastive pairs [3] [31] [71]. The augmented views can provide different perspectives of the original graph, enhancing the model's ability to capture important graph properties. Specifically for signed graphs, SGCL [65] applies graph contrastive learning to signed graphs, combining augmented graph

and signed structure contrasts. UGCL [42] improves stability with Laplacian perturbation, making it applicable to various graph types.

Below, we provide a detailed explanation of the most commonly used Graph Neural Networks (GNNs) and specifically focus on Signed Graph Neural Networks (SGNNs), which are applicable to our study. We also discuss attacks and defenses in the context of graph representation learning. Finally, we introduce the graph contrastive learning technique adopted in our method, as well as the graph augmentation involved.

2.1 Graph Neural Networks

2.1.1 Introduction to Graphs in Machine Learning

Graphs are data structures used to model a set of objects, consisting of nodes and their relationships represented by edges. In the realm of machine learning, graphs provide a rich representation for data with complex relationships, enabling a deeper understanding of interconnected systems. The nodes in a graph can represent entities such as users in a social network, molecules in a chemical compound, or words in a document, while the edges capture the connections or interactions between these entities.

The utilization of machine learning techniques for graph analysis has gained significant attention in recent years. The inherent flexibility and expressive power of graphs make them well-suited for capturing intricate patterns and dependencies in diverse datasets. By leveraging machine learning algorithms, researchers and practitioners can extract valuable insights from graph-structured data, leading to advancements in various fields.

Graphs find applications across a wide range of domains, illustrating their versatility and utility [76] [98]. In social sciences, graphs can model social relationships, influ-

ence networks, and information diffusion processes. In natural sciences, graphs are used to represent biological networks, chemical compounds, and ecological systems. Moreover, in bioinformatics, graphs play a crucial role in modeling protein-protein interaction networks, genetic interactions, and disease pathways.

2.1.2 Graph Analysis Tasks

Graph analysis, a field focused on the study of non-Euclidean data [66] structures, encompasses a diverse set of tasks crucial for unraveling complex relationships and patterns within interconnected datasets. Among the fundamental tasks in graph analysis are node classification [77], link prediction [86], and community detection [17].

Node classification [77] involves the assignment of labels or categories to nodes within a graph based on their attributes and connectivity patterns. By leveraging information from neighboring nodes and the overall graph structure, machine learning algorithms can accurately classify nodes, aiding in tasks such as identifying community influencers in social networks or predicting protein functions in biological networks.

Link prediction [86] is another essential task in graph analysis, aiming to forecast the likelihood of connections between nodes that are not currently linked. This task is particularly valuable for recommendation systems [81], social network analysis, and predicting potential interactions in various domains. By analyzing the network topology and node features, predictive models can uncover latent relationships and missing links within the graph.

Community detection [17] focuses on identifying cohesive subgroups or communities within a graph based on the density of connections between nodes. This task is instrumental in understanding the modular structure and functional units within complex networks [6], leading to insights into network dynamics, group behavior, and information diffusion processes [16].

Beyond node classification [77], link prediction [86], and community detection [17], several other essential tasks play a pivotal role in extracting insights from graph-structured data.

Graph clustering [99], a fundamental task in graph analysis, involves partitioning the nodes of a graph into clusters or groups based on their structural similarities or connectivity patterns. By identifying densely connected regions within the graph, clustering algorithms reveal underlying structures and help in understanding the organization of complex networks.

Anomaly detection [1] [56] in graphs is another critical task that focuses on identifying outliers or anomalies within the network. These anomalies could represent unusual patterns, deviations from the norm, or potentially fraudulent activities. Detecting such anomalies is essential for maintaining the integrity and security of networked systems.

Graph summarization [54] is a task that aims to condense large and complex graphs into more manageable and insightful representations. By capturing the essential characteristics and key properties of the original graph, summarization techniques facilitate a more concise and interpretable view of the underlying data, enabling efficient analysis and visualization.

Additionally, graph embedding [7] or graph representation learning [9] [83] has gained prominence as a task that involves mapping nodes or entire graphs into low-dimensional vector spaces while preserving their structural information and semantic relationships. These learned embeddings serve as powerful features for downstream machine learning tasks, enabling efficient information retrieval, similarity computation, and graph visualization.

GNNs have emerged as a powerful paradigm for analyzing graph-structured data, offering a scalable and expressive framework for learning representations from graph topology and node attributes. GNNs have garnered significant attention in recent

years for their exceptional performance across a wide range of graph analysis tasks. Their ability to capture intricate relationships and dependencies within graphs has led to breakthroughs in fields like social network analysis, bioinformatics, and recommendation systems [18] [32].

2.1.3 Types of GNNs

Graph Neural Networks (GNNs) have emerged as a powerful class of deep learning models tailored for processing and analyzing graph-structured data, exhibiting remarkable capabilities across various domains. Within the realm of GNNs, a spectrum of architectures has been devised to cater to different aspects of graph learning and representation. Among these architectures, Graph Convolutional Networks (GCNs) [87] stand out as a prevalent and effective framework for capturing local structural information [37] between nodes in a graph. GCNs excel in tasks such as node classification and link prediction, where understanding the local neighborhood relationships is crucial for accurate predictions and classifications. GCNs operate based on the principle of message passing between nodes in a graph. Initially proposed by Kipf and Welling, GCNs iteratively aggregate information from neighboring nodes, allowing each node to update its representation based on the information received from its local neighborhood. This process mimics the convolutional operation in traditional convolutional neural networks but is adapted to the graph domain. By capturing and propagating local structural information through multiple layers, GCNs can effectively model the graph's topology and learn meaningful node representations. This localized information aggregation makes GCNs particularly effective for tasks that rely on understanding the relationships and structures within a node's immediate vicinity.

On the other hand, Graph Attention Networks (GATs) [73] introduce attention mechanisms [61] into graph learning, enabling nodes to selectively attend to informative

neighbors during message passing. At the core of GATs is the attention mechanism, which allows each node to assign different importance weights to its neighbors based on learned attention coefficients. By dynamically focusing on relevant nodes and aggregating their representations with adaptive weights, GATs can effectively capture essential information within the graph structure. This attention-driven mechanism enables GATs to prioritize and incorporate crucial information from neighboring nodes, making them particularly adept at modeling complex relationships and dependencies within graph data.

GraphSAGE [28] represents another significant advancement in GNNs, employing node sampling and aggregation techniques to learn node representations effectively, especially in scenarios involving large-scale graph data [57] [58]. The key principle behind GraphSAGE is the idea of sampling a fixed-size neighborhood around each node and aggregating information from these sampled nodes to update the target node's representation. This approach allows GraphSAGE to scale effectively to massive graphs while preserving the graph's essential structural information. By aggregating information from diverse neighborhood samples, GraphSAGE can capture both local and global graph properties in the learned node representations. The sampling and aggregation process in GraphSAGE enables it to overcome computational challenges associated with processing large graphs, making it a versatile and scalable solution for learning representations in graph-structured data.

Furthermore, Gated Graph Neural Networks (GGNNs) [52] [63] incorporate gate mechanisms to control the flow and update of information throughout the network. These mechanisms enable GGNNs to manage long-range dependencies within the graph structure more efficiently, enhancing the model's ability to capture intricate relationships and patterns that span across distant nodes. GGNNs have demonstrated effectiveness in scenarios where temporal dependencies and long-range interactions play a critical role in the learning process.

In a different vein, Graph Isomorphism Networks (GINs) [70] focus on learning global,

permutation-invariant graph representations while preserving the isomorphic properties of the graph structure. By considering the entire graph as a whole during the representation learning process, GINs excel in scenarios where capturing the overall structural information of the graph is essential for downstream tasks such as graph classification and regression.

These diverse types of GNN architectures collectively provide a rich toolkit for analysts and researchers working with graph data, offering a range of choices to address specific challenges and requirements within graph-based learning tasks. By leveraging the unique strengths and capabilities of each GNN variant, practitioners can tailor their approach to suit the nuances of the dataset and the complexity of the task at hand, ultimately contributing to advancements in graph analysis, machine learning, and related fields.

2.1.4 Applications of GNNs

In the field of social network analysis, Graph Neural Networks (GNNs) are widely applied due to their ability to effectively capture and utilize the complex relationships and interaction patterns between nodes in social networks [60] [90]. Nodes in social networks typically represent individual users, while edges represent relationships or interactions between users. Through node classification, GNNs can effectively distinguish and categorize individuals within the social network, thereby identifying potential user groups or social relationships. This is of significant importance for user recommendation, targeted advertising, and social influence analysis on social media platforms. For instance, by classifying nodes, platforms can identify user groups with similar interests or behavior patterns, enabling precise ad targeting, which increases ad conversion rates and user satisfaction.

Moreover, the application of GNNs in link prediction helps forecast potential new relationships between users, thereby promoting the development and evolution of social

networks. Link prediction can assist platforms in identifying opportunities for friend recommendations and social circle expansions, enhancing user interaction and engagement. For example, by analyzing mutual friends, interests, and other information, GNNs can predict which users are likely to form new connections, thus recommending new friends or social circles to users and improving their social experience.

Through community detection, GNNs can reveal hidden groups and community structures within social networks, providing insights and decision support for social network managers and researchers. Community detection helps platforms identify tightly-knit user groups, enabling more targeted content recommendations and event planning. For instance, by identifying core users within a community, platforms can push relevant activities or content to that community, increasing user engagement and satisfaction.

In the field of recommendation systems [19] [20] [43] [76], the application of GNNs offers a new perspective and technical means for personalized recommendations. Traditional recommendation systems typically rely on users' historical behavior data, such as browsing and purchase records. In contrast, GNNs model the interaction relationships between users and items as a graph structure, allowing for a more accurate capture of users' interests and preferences, thereby improving the accuracy and user satisfaction of recommendation systems. Through the graph structure, GNNs can comprehensively consider various information, such as user similarities and item associations, to generate more precise recommendation results.

Especially when facing the cold start problem, GNNs can utilize existing graph information and node representation learning techniques to provide personalized recommendations for new users and items, thereby improving the performance and coverage of recommendation systems. The cold start problem is a significant challenge in recommendation systems, referring to how to provide effective recommendations for new users or items. By leveraging GNNs, recommendation systems can use information within the graph structure, such as social relationships between users and similarities

between items, to generate initial recommendation lists for new users or items, thus alleviating the cold start issue.

Additionally, recommendation systems that incorporate social network information utilize users' social relationships and interaction patterns to provide more targeted and personalized recommendation services, thereby enhancing user experience and platform stickiness. By analyzing users' social networks, recommendation systems can identify users' social circles, interests, and other information to generate more personalized recommendation results. For example, by analyzing users' friend relationships, recommendation systems can recommend items liked by friends to users, thereby increasing the relevance of recommendations and user satisfaction.

In summary, GNNs exhibit tremendous potential and development space in the applications of social network analysis and recommendation systems. They not only help in deeply understanding the complex structures and relationships within social networks but also provide more intelligent and personalized recommendation algorithms for recommendation systems. By combining the powerful representation learning capabilities of graph neural networks with the rich information in social networks, we can expect to see more innovations and breakthroughs in the fields of social network analysis and recommendation systems, bringing more high-quality and meaningful service experiences to users and platforms.

2.1.5 Training and Optimization of GNNs

Training and optimizing Graph Neural Networks (GNNs) involves critical steps and techniques to ensure their effectiveness in various applications. Prior to training GNNs, data preparation is essential, encompassing the representation of graph structures and node features in formats compatible with GNN models. Selecting a suitable GNN architecture is the second step, based on the specific task and characteristics of the graph data, such as Graph Convolutional Networks (GCNs), Graph Attention

Networks (GATs), and GraphSAGE.

To measure performance and guide the training process, selecting an appropriate loss function, such as node classification, link prediction, or graph regression, is necessary. Utilizing optimization algorithms like stochastic gradient descent (SGD) [2], Adam [39], or RMSprop [102] to update model parameters is crucial, with fine-tuning [59] of hyperparameters like learning rate, weight decay, and momentum for optimal performance.

Within the training loop, iterating through training data, passing it through the GNN model, calculating losses, and updating model parameters via backpropagation is essential. Monitoring the training process ensures convergence and guards against overfitting [64]. Apart from training, optimizing GNNs involves a range of strategies.

To prevent overfitting and enhance the model's generalization capabilities, regularization techniques like L_2 regularization or dropout can be applied. Optimizing hyperparameters such as learning rate, batch size, number of layers, and hidden units improves the model's performance on validation sets.

Gradient clipping [10] prevents exploding gradients during training by capping gradient values at predefined thresholds. Enriching input features with node attributes, edge features, or graph structure information enhances the learned representations of the GNN model.

Effective parameter initialization is critical. Utilizing appropriate initialization techniques ensures stable training and faster convergence. Additionally, implementing early stopping based on validation losses prevents overfitting on training data and enhances generalization to unseen data.

By adhering to these training and optimization strategies, researchers and practitioners can effectively train and fine-tune GNN models for tasks in social network analysis, recommendation systems, and other graph-related applications, improving performance and generalization to unseen data.

2.1.6 Cross-Domain Applications of GNNs

Graph Neural Networks (GNNs) have shown remarkable potential in various domains, including healthcare [49], finance [12], and transportation [82] [95], by effectively modeling complex relational data and addressing real-world challenges in innovative ways.

In healthcare [49], GNNs have revolutionized patient care and medical research by leveraging patient records, biological interactions, and medical knowledge graphs. GNNs can predict disease progression, recommend personalized treatments, and assist in drug discovery by analyzing molecular structures and interactions. For instance, GNNs can identify potential drug targets, predict patient outcomes, and optimize clinical workflows, leading to more efficient healthcare delivery and improved patient outcomes.

GNNs have transformed the financial sector by enhancing risk management, fraud detection, and investment strategies [12]. In finance, GNNs can analyze transaction networks to detect suspicious activities, predict market trends, and optimize portfolio management. By capturing intricate dependencies in financial data, GNNs enable more accurate risk assessment, fraud prevention, and investment decision-making, ultimately increasing efficiency and reducing financial risks.

In transportation [82] [95], GNNs are being used to optimize traffic flow, enhance route planning, and improve public transportation systems. By modeling transportation networks and analyzing traffic patterns, GNNs can predict congestion, recommend optimal routes, and facilitate real-time decision-making for traffic management. GNNs can also assist in predicting demand, optimizing logistics, and enhancing overall transportation efficiency, leading to reduced commute times and improved urban mobility.

GNN technology addresses complex real-world challenges by effectively capturing the underlying structures and relationships in diverse datasets. By leveraging graph rep-

resentations and learning from interconnected data points, GNNs excel at tasks requiring relational reasoning, pattern recognition, and predictive analytics. For example, in healthcare, GNNs can predict patient outcomes based on medical records and genetic data. In finance, GNNs can detect fraudulent transactions by analyzing transaction networks. In transportation, GNNs can optimize traffic flow by considering road connectivity and traffic patterns.

Overall, GNN technology offers a powerful framework for addressing complex real-world challenges across diverse domains, empowering industries to make data-driven decisions, improve operational efficiency, and drive innovation in healthcare, finance, transportation, and beyond.

2.1.7 Advantages of GNNs

There are several reasons for the adoption of Graph Neural Networks (GNNs). Firstly, traditional deep learning models like Convolutional Neural Networks (CNNs) [53] are not well-suited for non-Euclidean data structures, necessitating the need for GNNs [50]. GNNs excel at capturing and leveraging the complex relationships and interaction patterns between nodes in graph-structured data, such as social networks and recommendation systems. Secondly, graph representation learning aims to learn meaningful representations of graph nodes, edges, or subgraphs in the form of low-dimensional vectors [29]. This capability allows GNNs to integrate both node features and graph topology, enabling more accurate and context-aware predictions. The increasing interest in machine learning for graph analysis can be attributed to the expressive power of graphs as versatile representations for various systems and the unique challenges they pose, which necessitate specialized methods such as GNNs. As a result, GNNs significantly enhance performance in tasks like node classification, link prediction, and community detection by providing deeper insights into the underlying structure of the data. In recommendation systems [76], GNNs can better

understand user preferences and item similarities, leading to more personalized and precise recommendations. This ultimately enhances user experience and engagement by delivering content that is more relevant and tailored to individual needs. Additionally, GNNs are particularly effective in addressing challenges such as the cold start problem, where they can utilize existing graph information to make informed predictions even for new users or items. Overall, the ability of GNNs to harness the full potential of graph-structured data makes them a powerful tool in various applications, driving innovation and improving outcomes across multiple domains.

2.2 Signed Graph Neural Networks

The widespread influence of online platforms such as social media, business transactions, and cryptocurrency exchanges has led to a significant increase in graph datasets. These datasets possess complex and interconnected structures, posing significant challenges for analysis. Over the past decade, graph machine learning methods, particularly Graph Neural Networks (GNNs) [28] [40] [73], have garnered attention from academia and industry, demonstrating significant progress in various applications such as link prediction, node classification, and graph classification.

Even as GNNs have advanced considerably, many current GNN approaches are tailored for unsigned graphs that predominantly contain positive edges. In reality, node connections extend beyond positive ties like friendship and trust to encompass negative relationships such as enmity and distrust. These negative links disrupt the flow of information propagation, prompting the need for novel models like Signed Graph Neural Networks (SGNNs) to effectively accommodate both positive and negative connections.

Within the realm of signed graph analysis, there exist important surveys [69] [92] that delve into the properties and analysis tasks associated with signed graphs. These

surveys shed light on social balance [30], a critical collective property of signed graphs, covering fundamental measures and detection algorithms. However, the exploration of graph representation learning methods within these surveys is limited, a domain that has only recently garnered attention.

Currently, the analysis methods for signed graphs primarily utilize two approaches: signed graph embedding and, more predominantly, SGNNs. Among these methods, SiNE [74] represents a signed graph embedding approach that leverages deep neural networks and employs a loss function based on extended structural balance theory. The field has seen significant advancement with the introduction of various SGNN models. SGCN [13] pioneers a novel information aggregator grounded in balance theory, successfully extending GCN's application to signed graphs. Building upon this, SNEA [51] adapts the graph attention network (GAT) for signed graphs, maintaining its foundation in balance theory. BESIDE [11] takes a comprehensive approach by integrating both balance and status theories, specifically utilizing status theory to learn "bridge" edge information and combining it with triangle information. A notable breakthrough comes with SGCL [65], which is the first to extend Graph Contrastive Learning (GCL) to signed graphs. SDGNN [33] advances the field by combining balance and status theories while introducing four weight matrices for neighbor feature aggregation based on edge types. The development continues with RSGNN [91], which enhances SGNN performance through structure-based regularizers, effectively highlighting signed graphs' intrinsic properties while reducing vulnerability to input graph noise. Additional innovations include SDGCN [41], which introduces a spectral graph convolution encoder with a magnetic Laplacian, and UGCL [42], which presents a GCL framework incorporating Laplacian perturbation. While these diverse approaches demonstrate the rapid development of signed graph analysis methods, there is still considerable room for improvement in terms of model accuracy, and importantly, the security aspects of these methods remain largely unexplored in current research.

2.3 Attacks and Defenses for Graph Learning

In recent years, there has been a surge of interest in studying the robustness of graph learning models [80], leading to the exploration of various attack and defense methods [68]. This increased focus stems from the growing realization of the vulnerabilities present in these models when faced with adversarial manipulation [103]. Researchers have been actively investigating how these vulnerabilities can be exploited and what measures can be taken to mitigate these risks effectively.

Significantly, an in-depth exploration of adversarial attacks and defenses in images, graphs, and text has offered invaluable insights into the complex realm of security challenges present in various fields [79]. This exploration acts as a cornerstone resource, illuminating the varied threats that confront these models, spanning from image recognition systems to applications in natural language processing. Grasping the extensive scope and intricacies of these vulnerabilities is essential for crafting resilient defense strategies capable of standing up against advanced adversarial tactics.

Furthermore, researchers have delved into the specific challenges and techniques for defending graph convolutional networks against adversarial attacks [24] [48] [89]. These investigations have revealed the intricacies involved in safeguarding these networks, particularly when considering the complex interplay between the network structure and the potential attack vectors. By dissecting these challenges and developing tailored defense strategies, researchers aim to bolster the resilience of graph convolutional networks in the face of evolving adversarial threats.

Centered on signed graph analysis models, existing work has delved deeply into adversarial attacks on graph neural networks through the utilization of meta-learning techniques. These investigations have exposed the inherent vulnerabilities present in signed GNNs and showcased the feasibility of crafting targeted attacks that exploit these weaknesses. By unraveling the nuances of these attacks, avenues are being paved for the development of proactive defense mechanisms that can effectively counter such

threats.

Moreover, attention has been given to understanding the impact of adversarial attacks on the integrity and accuracy of signed graph analysis within the existing literature [84] [97]. These explorations have shed light on the disruptive nature of adversarial attacks on the fundamental operations of signed graph analysis models, emphasizing the importance of fortifying these models against malicious interventions [5] [22] [55] [91] [96]. By identifying the implications of such attacks, previous work provides insights for devising robust defense strategies that can uphold the reliability and accuracy of signed graph analysis in the face of adversarial challenges.

2.4 Graph Contrastive Learning

Graph Contrastive Learning (GCL) [27] [75] [93] is a cutting-edge technique in the field of graph representation learning. It focuses on enhancing node representations within graph structures by leveraging contrastive learning methods. The core concept of GCL involves training models to differentiate between positive and negative node pairs effectively. By maximizing the similarity between positive pairs and minimizing the similarity between negative pairs, GCL aims to learn informative node representations that capture the underlying structure of the graph.

One key technical aspect of GCL is the selection of an appropriate contrastive loss function, such as InfoNCE [72] [78] or NT-Xent [35] [38]. These loss functions guide the model in embedding similar nodes closer together while pushing dissimilar nodes apart in the embedding space. Additionally, the construction of positive and negative node pairs plays a crucial role in the training process. Positive pairs typically consist of nodes from the same graph structure, while negative pairs can be generated through random sampling [36] or other strategies to ensure a diverse training signal for the model.

GCL offers several advantages that make it a compelling approach in graph representation learning. Firstly, it enhances the quality of representation learning by encouraging the model to learn more discriminative and informative node embeddings. Secondly, GCL is adept at capturing both local and global structural information within graphs, leading to a deeper understanding and better representation of graph data. Furthermore, GCL boosts the model's robustness against noise and adversarial perturbations, making the learned representations more resilient [15] [21]. Lastly, GCL's simplicity and effectiveness make it a versatile tool applicable across various graph-related tasks, showcasing significant performance improvements in diverse applications.

2.5 Graph Augmentation

In Graph Contrastive Learning (GCL), graph augmentation plays a pivotal role in enriching the training dataset by applying various transformations and operations to the original graph data [85]. The primary goal is to generate new data samples to enhance the model's generalization capabilities and performance. Graph augmentation is a crucial strategy in GCL, introducing diverse data augmentation techniques to provide the model with richer training signals, enabling it to better capture the underlying features and structural information present in the graph data.

Within GCL, graph augmentation can be implemented in various ways, including random node additions or deletions, edge permutations or removals, subgraph sampling [34], among others. These operations aim to introduce varying degrees of noise and perturbations, encouraging the model to learn more robust and generalizable representations. By incorporating these diverse data augmentation strategies, GCL can train more powerful models with enhanced robustness and generalization capabilities, leading to superior performance in real-world applications.

Through graph augmentation, models not only receive diverse inputs during training but also adapt better to different data distributions and graph structures. This targeted data augmentation approach helps improve the model's generalization abilities, enabling it to handle unseen data samples more effectively and exhibit better performance in complex real-world environments.

Chapter 3

Preliminaries

3.1 Balance Theory

The social balance theory [94] stipulates that people tend to maintain relative symmetry in their social relations, especially triadic ones. It encapsulates the notion that "the friend of my friend is my friend" and "the enemy of my enemy is my friend". Following this, triangles in signed networks are classified as either balanced or unbalanced based on whether they consist of either even or odd number of negative links, respectively [46] [47]. For instance, the first two triads in Fig. 3.1, where all three users are friends or only one pair of them are friends, are considered balanced. To quantitatively assess the balance-related information, a metric known as the balance degree $D_3(G)$ was introduced [8]. This measure computes the proportion of balanced triads within the graph using the following formula:

$$D_3(G) = \frac{\mathsf{Tr}(A^3) + \mathsf{Tr}(|A|^3)}{2\mathsf{Tr}(|A|^3)}, \tag{3.1}$$

where $\mathsf{Tr}(\cdot)$ represents the trace of a matrix, A is the signed adjacency matrix of the signed graph $\mathcal{G}$. The balance degree D_3 of signed graph datasets commonly exhibits

a range of 0.85 to 0.95.

Balance theory plays an essential role in signed graph representation learning. In particular, SGCN [13] incorporates it into the aggregation process. Other SGNN models, such as SGCL [65] and SDGNN [33], leverage balance theory in augmentation or the construction of the loss function.

3.2 Signed Graph Analysis

SGCN [13], the pioneering SGNN model, extends GCN to handle signed graphs by incorporating balance theory to determine positive and negative relationships between nodes. To provide further clarity, the representation of a node v_i at a given layer l is defined as:

$$h_i^{(l)} = [h_i^{pos(l)}, h_i^{neg(l)}], \tag{3.2}$$

where $h_i^{pos(l)}$ and $h_i^{neg(l)}$ respectively denote the positive and negative representation vectors of node $v_i \in \mathcal{V}$ at the lth layer, and $[\cdot, \cdot]$ denotes the concatenation operation. The updating process for $l > 1$ layer could be written as:

$$\begin{aligned}
t_i^{pos(l)} &= AGG^{(l)}(h_j^{pos(l-1)} : v_j \in \mathcal{N}_i^+, h_j^{neg(l-1)} : v_j \in \mathcal{N}_i^-) \\
h_i^{pos(l)} &= COM^{(l)}(h_i^{pos(l-1)}, t_i^{pos(l)}) \\
t_i^{neg(l)} &= AGG^{(l)}(h_j^{neg(l-1)} : v_j \in \mathcal{N}_i^+, h_j^{pos(l-1)} : v_j \in \mathcal{N}_i^-) \\
h_i^{neg(l)} &= COM^{(l)}(h_i^{neg(l-1)}, t_i^{neg(l)}),
\end{aligned} \tag{3.3}$$

where AGG and COM refers to the aggregation and combination processes, respectively. t_i represents the temporary node representation vectors after the aggregation step. The set $\mathcal{N}$ corresponds to the neighbors of node v_i. SGNNs handle positive and negative edges by employing a two-part representation and a unique aggregation scheme. In other SGNN models, they hold similar mechanism. SGCL [65] and UGCL [42] utilize graph contrastive learning for signed graphs.. SGDNN [33] combines balance theory and status theory along with the introduction of four weight matrices.

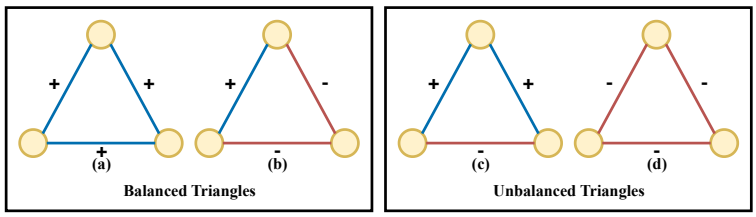

Figure 3.1: Balanced and unbalanced triangles. Positive and negative edges are represented by blue and red lines, respectively.

RSGNN [91] incorporates structure-based regularizers to enhance performance.

3.3 Link Sign Prediction

In this paper, we focus on the adversarial robustness of link sign prediction. Link sign prediction is a crucial task of analyzing signed graphs, as it entails deducing the signs of edges in the uncharted section of the graph. This prediction relies on a known subgraph, encompassing both its structure and edge signs. In the realm of signed graph analysis, link sign prediction takes precedence over other tasks such as node ranking.

Formally, we define a signed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-)$ where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ represents the set of n nodes. The positive edges are denoted by $\mathcal{E}^+ \subseteq \mathcal{V} \times \mathcal{V}$, while the negative edges are $\mathcal{E}^- \subseteq \mathcal{V} \times \mathcal{V}$, and $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$. We denote the sign of edge e_{ij} as $\sigma(e_{ij}) \in \{+, -\}$. The structure of $\mathcal{G}$ is captured by the adjacency matrix $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. Note that for signed graphs, a node is normally not given a feature. $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a node embedding matrix and d is the dimension of nodes.

Chapter 4

Problem Definition and Our Preliminary Analysis

We focus on the task of link sign prediction, which involves predicting the signs of edges in the complementary part of a given subgraph of a signed graph with known structure and edge signs. To begin, we introduce the necessary notations and formulate the attack model and defense model accordingly. We consider an analyst predicting missing signs from a collected signed graph, while an attacker aims to disrupt the prediction task. The attacker can manipulate the data collection process, resulting in a poisoned graph used for training the prediction model. We assume a strong attacker with full access to the training data, including the graph structure and link signs, and the ability to change signs within a specified budget.

4.1 Notations

Formally, let $G = (V, E^+, E^-)$ be a signed-directed graph where $V = \{v_1, v_2, \cdots, v_n\}$ represents the set of n nodes. The positive edges are denoted by $E^+ \subseteq V \times V$, while the negative edges are $E^- \subseteq V \times V$, and $E^+ \cap E^- = \emptyset$. Let $\mathbb{I}\{\cdot\}$ be the

indicator function, and $sign(\cdot)$ be the sign function. We denote the sign of edge e_{ij} as $sign(e_{ij}) \in \{+, -\}$. The structure of G is captured by the adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$, where each entry $A_{ij} \in \{1, -1, 0\}$ represent negative edges, positive edges, or the absence of an edge in the signed graphs. We denote the training edges and testing edges by $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$, respectively, and each edge $e \in \mathcal{D}_{train} \cup \mathcal{D}_{test}$ has its sign label $sign(e)$. Let $\mathcal{L}_{train}$ be the training loss of the target model based on $\mathcal{D}_{train}$, and θ denote the model parameter. The model predictions for the sign of edges are denoted as $f_{\theta^*}(G)$, and $f_{\theta^*}(G)_e \in \{+, -\}$ is the prediction for the given edge $e \in E^+ \cap E^-$. $\mathcal{L}_{train}$ is the training loss of the target model and $\mathcal{L}_{atk}$ represents the objective that the attacker seeks to optimize.

4.2 Threat Model

4.2.1 Attacker's goal

Our study aims to investigate the vulnerability of link sign prediction models by developing a black-box attack that aims to assess the extent to which the predictions of the algorithm can be disturbed. Following [104], we focus on global attacks, aiming to decrease the overall prediction performance of the model. We leverage an attack method to manipulate the graph effectively. The modified graph is then utilized to train SGNNs, intentionally aiming to degrade their performance.

4.2.2 Attacker's knowledge

We assume that the attackers have access to the training data, enabling them to observe both the graph structure and edge signs, but they do not know the model structure and parameters.

4.2.3 Attacker's capability

To ensure effective and inconspicuous adversarial attacks, we impose a budget constraint denoted as Δ, limiting the number of changes made to the graph. Specifically, the constraint restricts the number of altered edges $\|A - \hat{A}\|_0$ to stay within Δ. In our case, we disregard changes in edge signs and assume graph symmetry, resulting in a budget constraint of 2Δ. We also take precautions to prevent node disconnection during the attack process. Unnoticeability of changes is maintained by imposing a constraint on the degree distribution. Although our current focus is altering edge signs, our algorithm can be easily adapted to modify the overall graph structure. These constraints are consolidated as the set of permissible perturbations on the given graph G, denoted as $\Phi(G; \Delta)$.

4.3 Problem of Attack

In the case of global and unspecific attacks, the primary aim of the attacker is to reduce the model's generalization performance on the testing nodes. Poisoning attacks can be mathematically formulated as a bi-level optimization problem:

$$\min_{\hat{G} \in \Phi(G;\Delta)} \mathcal{L}_{atk} = \sum_{e \in \mathcal{D}_{test}} \mathbb{I}\{f_{\theta^*}(\hat{G})_e = sign(e)\}, \tag{4.1}$$
$$s.t.\ \theta^* = arg\min_{\theta} \mathcal{L}_{train}(f_\theta(\hat{G})),$$

where the attacker aims to reduce the number of testing edges to be correctly classified by manipulating the graph, and the model itself is trained on the manipulated graph.

We consider an analyst predicting missing signs from a collected signed graph, while an attacker aims to disrupt the prediction task. The attacker can manipulate the data collection process, resulting in a poisoned graph used for training the prediction model. We assume a strong attacker with full access to the training data, including the graph structure and link signs, and the ability to change signs within a specified

budget.

Table 4.1: Comparison of SGCN without/with *Balance Learning* under balance-attack (Ratio: Overlapping Ratio of Graphs; D_3: Balance Degree)

Dataset	Ptb(%)	SGCN			SGCN+*balance learning*		
		AUC	ratio(%)	D_3	AUC	ratio(%)	D_3
BitcoinAlpha	0	**0.7992**	**100.00**	0.9232	0.7981	97.75	**0.9976**
	10	0.6913	**89.98**	0.2006	**0.6962**	84.77	**0.9856**
	20	**0.6535**	**79.94**	0.1054	0.6153	63.82	**0.9616**
BitconOTC	0	**0.8253**	**100.00**	0.9267	0.8113	96.62	**0.9978**
	10	**0.7504**	**89.99**	0.2072	0.7324	79.32	**0.9598**
	20	**0.6985**	**79.98**	0.0881	0.6687	64.92	**0.9335**
Slashdot	0	**0.8153**	**100.00**	0.9331	0.7957	98.57	**0.9981**
	10	**0.6892**	**89.78**	0.2345	0.6668	84.13	**0.9436**
	20	**0.6348**	**79.96**	0.1472	0.6092	68.23	**0.9031**
Epinions	0	0.7763	**100.00**	0.8099	**0.7814**	97.47	**0.9889**
	10	**0.7383**	**89.97**	0.3889	0.7253	88.58	**0.9384**
	20	**0.6881**	**79.83**	0.2197	0.6824	75.92	**0.9081**

4.4 Problem of Defense

Given a poisoned graph, the defender (i.e., the analyst) aims to train a *robust* SGNN model to mitigate the impact of the attack. The primary objective for the defender is to restore the prediction accuracy to a level comparable to that of an unpolluted graph. It is important to note that the defender is only aware of the poisoned graph and does not have access to a clean version. We emphasize that the defender does not know the attack algorithm. That is, the defender's goal is to develop a model that would stay robust possibly against different types of attacks.

4.5 Our Preliminary Analysis

The attacks proposed in the literature, balance-attack [97] and FlipAttack [101] in particular, have been shown to considerably reduce the balance degree of the graph. In response, a natural defense strategy is to restore the balance of the poisoned graph. To this end, we adapt the widely used *structural learning* technique for unsigned graphs and employ it to train a robust SGNN model. Specifically, this method regards the signs as variables and employs the balance degree as a regularizer to iteratively update the graphs, aiming to maximize the balance degree by updating signs without altering the graph structure. We term this method as *balance learning*.

Table 4.2: Comparison of SGCN without/with *Balance Learning* under FlipAttack (Ratio: Overlapping Ratio of Graphs; D_3: Balance Dsegree)

Dataset	Ptb(%)	SGCN			SGCN+*balance learning*		
		AUC	ratio(%)	D_3	AUC	ratio(%)	D_3
BitcoinAlpha	0	**0.7992**	**100.00**	0.9232	0.7981	97.75	**0.9976**
	10	**0.6746**	**89.98**	0.6942	0.6634	86.24	**0.9883**
	20	**0.5601**	**79.94**	0.6806	0.5429	74.79	**0.9628**
BitconOTC	0	**0.8253**	**100.00**	0.9267	0.8113	96.62	**0.9978**
	10	**0.6844**	**89.99**	0.6988	0.6792	85.23	**0.9782**
	20	**0.6315**	**79.98**	0.6557	0.6159	73.04	**0.9553**
Slashdot	0	**0.8153**	**100.00**	0.9331	0.7957	98.57	**0.9981**
	10	**0.6412**	**89.78**	0.6423	0.6382	87.53	**0.9723**
	20	**0.6041**	**79.96**	0.6153	0.5923	73.24	**0.9358**
Epinions	0	0.7763	**100.00**	0.8099	**0.7814**	97.47	**0.9889**
	10	**0.6898**	**89.97**	0.6336	0.6792	87.82	**0.9624**
	20	0.6149	**79.83**	0.5961	**0.6234**	75.46	**0.9314**

We measure the performance of *balance learning* under different perturbation ratios of balance-attack [97] and FlipAttack [101] and show AUC in Tab. 4.1 and Tab.

Table 4.3: Overlapping Ratios of RSGNN under Adversarial Attacks

Dataset	Ptb(%)	balance-attack(%)	FlipAttack(%)
BitcoinAlpha	10	74.87	86.22
	20	56.71	74.77
BitcoinOTC	10	75.33	85.05
	20	61.19	72.83
Slashdot	10	83.72	87.12
	20	66.06	73.89
Epinions	10	88.04	86.67
	20	74.57	75.78

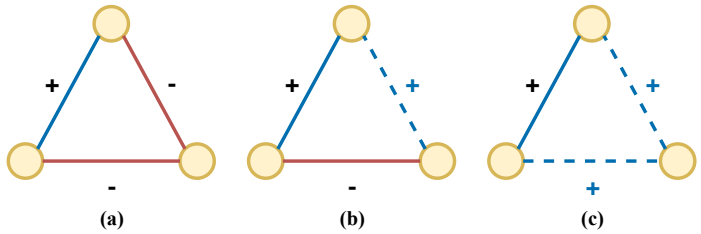

Figure 4.1: An example of the Irreversibility of Balance-related Information challenge. (a) The initial balanced graph; (b) The unbalanced graph after attack; (c) A recovered graph, which is balanced but has a different sign distribution from the graph in (a).

4.2, respectively. We observe that while *balance learning* can significantly recover the degree of balance, it can not improve the model performance as measured by AUC. In addition, *balance learning* will also result in a lower overlapping ratio between the poisoned graphs and the clean graphs, as observed when comparing the overlapping ratio before and after applying *balance learning*, thereby indicating its incapacity to recover the signs. We evaluate the comparable metrics for methods such as RSGNN [91], which incorporates a high balance degree as part of their regularizers, thereby leveraging balance learning techniques. On RSGNN, we observe the same phenomenon as in the above test, as illustrated in Tab. 4.3.

The primary reason for the ineffectiveness of *balance learning* is that different distributions of the signs could result in the same degree of balance (a toy example is shown in Fig. 4.1). Consequently, it is difficult to *reversely* restore the sign distribution of the clean graph using the balance degree as the single guidance. Indeed as shown in Tab. 4.1, while the balance degree is restored, the distribution of the signs is still quite different from that of the clean graph. We term this observation as *Irreversibility of Balance-related Information*.

Chapter 5

Methodologies

In this section, we divide our discussion into three parts: 1) Our black-box attacks, 2) Our robust SGNN models, and 3) The theoretical foundations of the aforementioned methods.

5.1 Proposed Black-Box Attack

5.1.1 Formulation of black-box attack

Since the model structure and labels of the testing data are always unavailable, directly optimizing Eq. (4.1) becomes infeasible. To address this challenge, we adopt an alternative approach by minimizing the balance degree of the graph. According to the analysis conducted in a previous study [91], it has been determined that SGNNs lack the ability to effectively learn precise node representations from unbalanced triangles. From this finding, we can infer that targeting the balance attribute of graphs has the potential to degrade the performance of SGNNs. Consequently, if the target model θ is trained on a poisoned graph that has a low balance degree, it is expected to exhibit an also low $\mathcal{L}_{atk}$ value. Therefore, we replace the optimization problem Eq.

(4.1) with the optimization problem as follows:

$$\min_{\hat{G}\in\Phi(G;\Delta)} D_3(\hat{G}). \tag{5.1}$$

5.1.2 Attack Method

In the training phase, our objective is to minimize the balance degree of the subgraph $\hat{G}$ within a specified budget Δ. This problem, however, is challenging due to the discrete nature of the signs. As mentioned in [14], optimizing this problem is known to be NP-hard. To approximate the optimization problem, we propose an algorithm based on gradient descent and greedy edge selection.

Our solution revolves around the core concept of computing the gradient of the objective function $D_3(\hat{G})$ with respect to the adjacency matrix A. The primary approach employed is an iterative and greedy strategy that involves flipping the sign of an existing edge with the highest absolute gradient value and the correct sign, while ensuring compliance with the budget constraint. In the given scenario, if the candidate edge possesses a positive sign and its gradient value is negative, updating the adjacency value through gradient descent would result in a value exceeding 1, thereby violating the constraints inherent in an adjacency matrix. The modification options available encompass the selection of positive edges with the maximum positive gradient values or negative edges with the maximum negative gradient values. Consequently, during each epoch, one edge is chosen from these options for updating. This iterative process continues until the budget is exhausted. During each iteration, we update an element in the adjacency matrix using the following procedure:

$$\begin{aligned} i^*, j^* &= \underset{\substack{\{i,j|a_{ij}\neq 0 \wedge sign(a_{ij})= \\ sign(\nabla_{ij} D_3(\hat{G}))\}}}{arg\max} |\nabla_{ij} D_3(\hat{G})|, \\ a_{ij} &= -a_{i^*j^*}, \end{aligned} \tag{5.2}$$

where a_{ij} represents an element located at row i and column j of the adjacency matrix. The variable $\nabla_{ij} D_3(\hat{G})$ denotes the gradient of each edge computed through

back-propagation.

To provide a clearer understanding of our approach, we outline the steps of our greedy flips method in Alg. 1.

Algorithm 1 Algorithm of ***balance*-attack** via Greedy Flips

Input: Adjacency matrix A of G, perturbation budget Δ.

Output: Attacked adjacency matrix S (s is the element in S).

Initialize $S \leftarrow A$.
while Number of changed edges $\leq \Delta$ **do**
    Calculate $D_3(S)$.
    Calculate gradient matrix $\nabla(D_3(S))$.
    Filter candidate edges $C_e = \{i, j | s_{ij} \neq 0 \wedge \text{sign}(s_{ij}) = \text{sign}(\nabla_{ij} D_3(S))\}$.
    if $i^*, j^* = \arg\max_{\{i,j \in C_e\}} |\nabla_{ij}(D_3(S))|$ **then**
        Update $s_{i^*j^*} = -s_{i^*j^*}$.
        Number of changed edges $+ = 1$.
    end if
end while
Return S.

5.2 Balance Augmented-Signed Graph Contrastive Learning

5.2.1 Overview

Directly addressing the Irreversibility of Balance-related Information is a daunting task as the clean graph is not accessible. To address this issue, we introduce Balance Augmented-Signed Graph Contrastive Learning (**BA-SGCL**), which utilizes Graph Contrastive Learning (GCL) to generate robust embeddings from the attacked graph

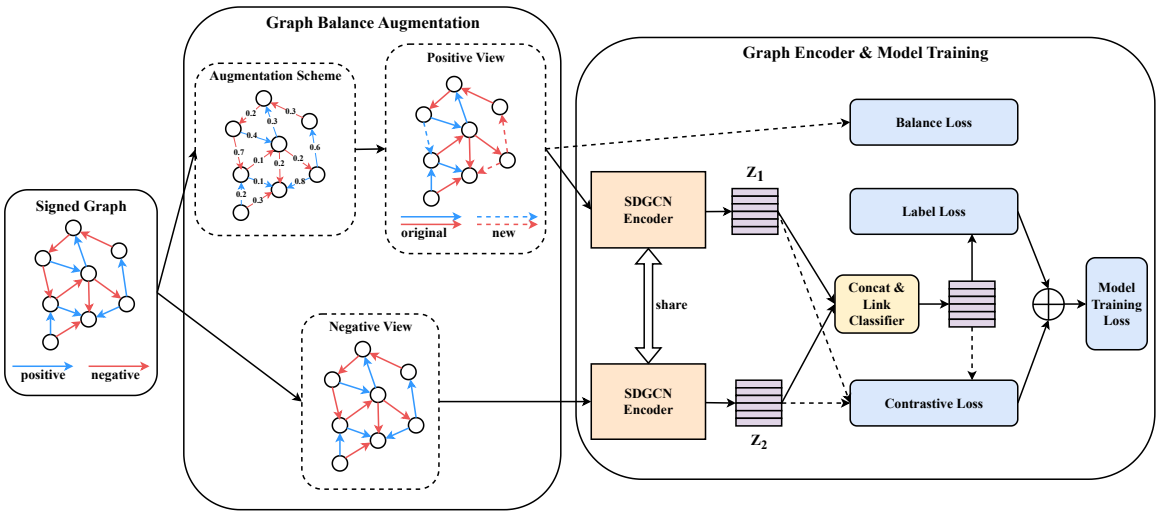

Figure 5.1: The Overview of BA-SGCL.

instead of increasing the balance degree directly. BA-SGCL integrates Graph Contrastive Learning (GCL) with a novel learnable balance augmentation to improve the robustness of embeddings. Specifically, we design the positive and negative views augmentation in GCL such that one view has an improved balanced degree while the other view is the poisoned graph tends to have a smaller balanced degree. Although it is difficult to directly recover the structural and sign information of the clean graph through balance learning, our method makes *the final node embeddings in latent space characterized with a high balance degree* by using GCL. Our proposed solution maximizes the mutual information between the embedding of the poisoned graph and the joint distribution of the positive view's embedding and labels in order to effectively achieve the defense objective. The framework of BA-SGCL is illustrated in Fig. 5.1. In the following, we first describe the model details and then delve into the theoretical underpinnings of the attacks and also elucidate how our model can enhance representation learning from an information theoretical perspective.

5.2.2 Learnable Balance Augmentation

Despite the difficulty in restoring the clean graph with a highly balanced degree accurately, we mitigate the issue via balance augmentation leveraging a GCL framework. GCL mainly relies on generating pairs of positive and negative views to conduct self-supervised learning. In the context of defending against poisoning attacks, we only have knowledge of the poisoned graphs where the attacker has hindered the balanced degrees. These poisoned graphs can serve as negative views. To generate a positive view with increased balance, we introduce a novel balance augmentation technique involving flipping the signs on the poisoned graph.

Specifically, we learn a Bernoulli distribution to determine the flipping of the signs that increase the balanced degree. Let $\Delta = [\Delta_{ij}]_{n\times n} \in [0,1]^{n\times n}$ denote the probability of flipping. The key to our balance augmentation is to learn the optimal Δ. We further represent the edge flipping Bernoulli distribution as $\mathcal{B}(\Delta_{ij})$. Then, we can sample a sign perturbation matrix denoted as $E \in \{0,1\}^{n\times n}$, where $E_{ij} \sim \mathcal{B}(\Delta_{ij})$ indicates whether to flip the sign of edge (i,j). If $E_{ij} = 1$, we flip the sign; otherwise not. The sampled augmented positive graph can be represented as A_p as follows:

$$A_p = A + C \circ E, \tag{5.3}$$

where $C = -2 \times A$ denotes legitimate edge sign flipping for each node pair. Specifically, changing the sign of positive edge (i,j) to negative is allowed if $C_{ij} = -2$, while changing the negative edge sign to positive is allowed if $C_{ij} = 2$. By taking the Hadamard product $C \circ E$, we obtain valid sign perturbations for the graph.

Since E is a matrix of random variables following Bernoulli distributions, we can easily obtain the expectation of sampled augmented graphs as $\mathbb{E}[A_p] = A + C \circ \Delta$. Therefore, the probability matrix Δ controls the balance augmentation scheme. To learn the parameter Δ, we next define the augmentation learning as the following problem:

$$\begin{aligned} &\min_{A_p \in \Phi(A)} \mathcal{L}_{ptb} = -\frac{\mathsf{Tr}(A_p^3) + \mathsf{Tr}(|A_p|^3)}{2\mathsf{Tr}(|A_p|^3)}, \\ &s.t.\ A_p = A + C \circ E, C = -2 \times A, E_{ij} \sim \mathcal{B}(\Delta_{ij}), \end{aligned} \tag{5.4}$$

where $\mathcal{L}_{ptb}$ is the negative balance degree. By minimizing it, we aim to generate positive view with balance degree as large as possible. To avoid deformation of original adjacency matrix A, we add a constraint denoted by $\Phi(A)$ to limit the maximum number of edge flipping from A. In practice, we choose the top $n\%$ of Δ_{ij} to sample E_{ij}, where $n\%$ is the perturbation budget. With the positive view generated, we will next demonstrate the graph encoder and the contrastive loss that are used to learn the robust embedding.

We adopt SDGCN [41] as our encoder, which is currently the state-of-the-art SGNN encoder. SDGCN overcomes the limitations of the graph Laplacian and utilizes complex numbers to represent both the sign and direction information of edges in signed graphs.

5.2.3 Design of Loss Function

The losses include the contrastive loss $\mathcal{L}_{con}$, label loss $\mathcal{L}_{label}$, and balance loss $\mathcal{L}_{balance}$, which correspond to contrastive learning, the link sign prediction task, and balance augmentation, respectively. We utilize the combination of contrastive loss $\mathcal{L}_{con}$ and label loss $\mathcal{L}_{label}$ to train the encoder's parameters, while the balance loss $\mathcal{L}_{balance}$ is employed to train the probability matrix Δ within the augmentation scheme.

Contrastive loss

The contrastive objective focuses on aligning the latent representations of the same node while distinguishing them from other nodes. Two identical nodes from different graph views are considered as an inter-positive pair, while other node pairs are

considered inter-negative pairs. For example, a node u from $\mathcal{G}_1$ and the same node u from $\mathcal{G}_2$ form an inter-positive pair. Conversely, any other node $v \in \mathcal{V}; v \neq u$ from $\mathcal{G}_2$ and node u of $\mathcal{G}_1$ form an inter-negative pair. Even though the nodes in the inter-positive pair come from different graph views, they are the same nodes. The goal of the inter-view objective is to maximize the similarity of positive pairs and minimize the similarity of negative pairs. The inter-view loss function is defined as:

$$\mathcal{L}_{inter} = \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} log \frac{exp((z_1^u \cdot z_2^u)/\tau)}{\sum_{v \in \mathcal{V}} exp((z_1^u \cdot z_2^v)/\tau)}. \tag{5.5}$$

where z_1^u and z_2^u represent the low-dimensional embedding vectors of node u from view 1 and view 2, respectively.

The intra-view loss serves the purpose of calculating the discriminative loss within a single graph view, in contrast to the inter-view loss which compares the latent representations of nodes between two distinct graph views. It plays a critical role in ensuring that the latent representations of all nodes are distinct from one another, taking into account their individual and unique characteristics. The primary objective is to promote distinctiveness among the latent representations of all nodes. Mathematically, the intra-view loss can be defined as follows:

$$\mathcal{L}_{intra} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} log \frac{1}{\sum_{v \in \mathcal{V}, u \neq v} exp((z_k^u \cdot z_k^v)/\tau)}, \tag{5.6}$$

where k indicates the graph view index.

The contrastive loss is the sum of the inter-view and intra-view loss functions:

$$\mathcal{L}_{con} = \mathcal{L}_{inter} + \mathcal{L}_{intra}. \tag{5.7}$$

Label Loss

The augmented views serve as input to the graph encoders, generating node representations $\mathbf{Z}_1$ and $\mathbf{Z}_2$. These representations are concatenated and pass through the

output layer to produce the final node embedding as:

$$\mathbf{R} = \sigma([\mathbf{Z}_1||\mathbf{Z}_2]\mathbf{W}^{out} + \mathbf{B}^{out}). \tag{5.8}$$

Specifically, after generating the final representations for all nodes by Eq. (5.8), we utilize a 2-layer MLP to estimate the sign scores from i to j:

$$\hat{y_{i,j}} = \sigma([\mathbf{r}_i||\mathbf{r}_j]\mathbf{W}^{pred} + \mathbf{B}^{pred}). \tag{5.9}$$

The loss function of the link sign prediction is formulated based on the cross entropy:

$$\begin{aligned} \mathcal{L}_{label} = &- \sum_{(i,j)\in\Omega^+} y_{i,j} log\sigma(\hat{y_{i,j}}) \\ &- \sum_{(i',j')\in\Omega^-} (1 - y_{i',j'}) log(1 - \sigma(\hat{y_{i',j'}})), \end{aligned} \tag{5.10}$$

where Ω^+ and Ω^- denote the training positive node pairs and negative node pairs respectively, $\sigma(\dot{})$ is the sigmoid function, and $y_{i,j}$ represents the sign ground truth.

Balance Loss

Balance loss $\mathcal{L}_{balance}$ is the same as Eq. (5.4). To enhance the positive views, the balance degree serves as a guiding factor. By enhancing the balance degree of positive views to above approximately 0.8, we simulate the high balance degree characteristic of clean signed graphs.

5.2.4 Model Training

Contrastive learning can be viewed as the regularization of the target task, thus we update model encoder's parameters using the combination, as follow:

$$\mathcal{L} = \alpha \times \mathcal{L}_{con} + \mathcal{L}_{label}. \tag{5.11}$$

To augment the positive view, the probability matrix Δ in the augmentation scheme is updated using the balance loss $\mathcal{L}_{balance}$. This updated Δ is then used to sample the

Algorithm 2 BA-SGCL Training Algorithm

for *epoch* $= 0, 1, \ldots$ **do**
    // Learnable Graph Augmentations
    (Generate a random augmentation scheme initially)
    Sample a positive view via Eq. (5.3)
    // Graph Encoders
    Obtain two representations of two views $\mathbf{Z}_1$ and $\mathbf{Z}_2$
    // Contrastive Learning
    Compute inter-view contrastive loss $\mathcal{L}_{inter}$, intra-view contrastive loss $\mathcal{L}_{intra}$, combined contrastive loss $\mathcal{L}_{con}$ via Eq. (5.5), (5.6), and (5.7)
    // Model Training
    Compute the loss of sign link prediction task $\mathcal{L}_{label}$ via Eq. (5.10) and combination loss $\mathcal{L}$ via Eq. (5.11)
    Compute balance loss via Eq. (5.4)
    Update model parameter θ by $\frac{\partial \mathcal{L}}{\partial \theta}$ and augmentation scheme Δ by $\frac{\partial \mathcal{L}_{balance}}{\partial \Delta}$
end for
return node representations $\mathbf{Z}$

positive view. The introduction of balance loss in every training iteration, followed by the training of contrastive loss and label loss, would result in substantial time overhead. Therefore, we propose to train the above losses at one stage and arrange them uniformly in the model training stage. The learning algorithm is outlined in Alg. 2.

5.3 Theoretical Analysis

In this section, we undertake an analysis of adversarial attacks through the lens of mutual information and then present the theoretical foundations of our defense framework.

Theorem 1. *The essence of adversarial attacks specified for signed graphs is to decrease the mutual information between balanced information and labels Y by the perturbed balanced information $\hat{B}$, thus reducing model performance:*

$$\arg\min_{\hat{B}} I(\hat{B}; Y), \tag{5.12}$$

where $I(\dot{;})$ is mutual information.

Proof. Suppose the SGNN model f_θ accepts a signed graph $\mathcal{G}$ as the input and output the embedding $\mathbf{Z}$: $\mathbf{Z} = f_\theta(\mathcal{G})$. Due to there being no node attributes in real-world signed graph datasets [25] [26] [44], existing SGNN aims at capturing structural information and balance-related information to predict labels Y, we can formulate the embedding of SGNN models as follows:

$$\arg\max_{\theta} I(f_\theta(A, B); Y), \tag{5.13}$$

where A and B are structural information and balance-related information respectively. The goal of SGNN is to maximize the mutual information between embeddings and labels.

Theorem. 1 provides the insight that offers guidance for defending against attacks: a robust model should conversely increase the mutual information between balance-related information and labels. Next, we introduce the theoretical foundation of our robust model BA-SGCL.

Since A and B are independent, we can rewrite Eq. (5.13) as:

$$\arg\max_{g_1, g_2} I((g_1(A), g_2(B)); Y), \tag{5.14}$$

where g_1 and g_2 are two functions that capture structure information and balance-related information, respectively. Suppose $h_A = g_1(A)$ and $h_B = g_2(B)$, according to properties of the mutual information, we have:

$$\begin{aligned} I((g_1(A), g_2(B)); Y) &= I((h_A, h_B); Y) \\ &= I(h_A; Y) + I(h_B; Y | h_A). \end{aligned} \tag{5.15}$$

Conversed with the goal of SGNN models, the goal of attackers is minimizing Eq. (5.13) via certain perturbations to induce the model to give wrong predictions.

Specifically, the balance-related attack perturbs the balance degree of the input signed graph while maintaining its structure. Therefore, A remains unchanged while B is converted to $\hat{B}$. The target of the attack can be formulated as:

$$\begin{aligned} \arg\min_{\hat{B}} I(f_\theta(A, \hat{B}); Y) &= \arg\min_{\hat{B}} I((h_A, h_{\hat{B}}); Y) \\ &= I(h_A; Y) + I(h_{\hat{B}}; Y|h_A). \end{aligned} \tag{5.16}$$

Both A and Y remain unchanged, thus the target of the balance-related attack is minimizing $I(h_{\hat{B}}; Y|h_A)$, which means the attack aims at minimizing the mutual information between balance-related information and labels that are not related to structural information.

□

Theorem 2. *BA-SGCL generates robust embeddings by maximizing the mutual information between the perturbed graph's embeddings* $\mathbf{Z}_2$ *and the joint distribution of the embeddings of positive sample* $\mathbf{Z}_1$ *and labels* Y*:* $\max I((\mathbf{Z}_1, Y); \mathbf{Z}_2)$.

Proof. According to principles of mutual information, we can rewrite $I((\mathbf{Z}_1, Y); \mathbf{Z}_2)$ as follows:

$$\max I((\mathbf{Z}_1, Y); \mathbf{Z}_2) = \max(I(\mathbf{Z}_1, \mathbf{Z}_2) + I(\mathbf{Z}_2; Y|\mathbf{Z}_1)).$$

The first term on the right-hand side of the equation implies that the embedding of the positive view should share as much mutual information as possible with the embedding of the original graph, or in other words, the robust embedding should capture high balance characteristics. The second term indicates that the defense model should maximize the mutual information between the embedding and the labels, thereby enabling accurate label predictions. □

In summary, increasing the balance degree of the positive view requires meeting dual requirements: accurately representing the original graph and accurately predicting the labels. The first requirement is achieved through the GCL framework and the contrastive loss proposed by us. The second requirement is fulfilled by the label loss.

Chapter 6

Experiments

This section includes separate evaluations of balance-attack and BA-SGCL.

We perform experiments on four real-world datasets to showcase the efficacy of the proposed balance-attack in diminishing the performance of SGNNs compared to random attacks in link sign prediction. Additionally, we apply balance-attack to five state-of-the-art methods in signed graph representation. We will answer the following questions:

- **Q1**: Can balance-attack decrease the balance degree of signed graphs significantly?
- **Q2**: How does balance-attack perform on existing SGNN models compared with random attack?
- **Q3**: How applicable is balance-attack on various SGNN models?

We test the performance of link sign prediction with nine state-of-the-art SGNNs and our robust model BA-SGCL under different adversarial attacks specifically designed for signed graphs. We aim to provide insights for the following two major questions:

- **Q4**: How does BA-SGCL perform compared with other SGNN methods under different signed graph adversarial attacks?

- **Q5**: How effective is balance augmentation in BA-SGCL compared to random augmentation in original SGCL?

6.1 Datasets

We conduct experiments on four public real-world datasets: Bitcoin-Alpha, Bitcoin-OTC [44], Epinions [25], and Slashdot [26]. The Bitcoin-Alpha and Bitcoin-OTC datasets are publicly available and collected from Bitcoin trading platforms. These datasets are obtained from platforms where users have the ability to label other users as either trust (positive) or distrust (negative) users. This labeling system serves as a means to prevent transactions with fraudulent and risky users from trading or perform transactions, given the anonymity of these trading platforms. Slashdot is a renowned technology-related news website that boasts a distinctive user community. Within this community, users have the option to tag each other as friends or foes based on their interactions and relationships. Similarly, Epinions represents an online social network centered around a general consumer review site called Epinions.com. The users of this site have the autonomy to decide whether they trust other members or not, forming a network based on mutual trust relationships. In the experiments, we randomly select 80% links as training set and the remaining 20% as testing set. Since these datasets have no attributes, we randomly generate a 64-dimensional vector for each node as the initial node attribute. More detailed dataset statistics are shown in Tab. 6.1.

Table 6.1: Dataset Statistics

Dataset	#Nodes	#Pos-Edges	#Neg-Edges	%Pos-Ratio	%Density
Bitcoin-Alpha	3,784	22,650	1,536	93.65	0.3379%
Bitcoin-OTC	5,901	32,029	3,563	89.99	0.2045%
Slashdot	33,586	295,201	100,802	74.55	0.0702%
Epinions	16,992	276,309	50,918	84.43	0.2266%

6.2 Setup

To conduct our evaluation, we divide the available datasets randomly, allocating 80% of the links for training purposes and reserving the remaining 20% for testing. As the datasets lack attributes, we generate a random 64-dimensional vector as the initial node attribute.

6.2.1 Attack Setup

We follow the hyper-parameter setting suggestions by those papers and set the embedding dimension to 64 for all SGNN models to achieve a fair comparison. To speed up the attack process, we opt to modify 10 edges per epoch. Specifically, we target the 10 elements in the adjacency matrix that possess the highest absolute gradient values and the correct signs, when doing back-propagation. In the experiment, the perturbation rate varies from 5% to 20% of total edges. To evaluate our method, we employ three metrics: micro-average F1 score (Micro-F1), binary-average F1 score (Binary-F1), and macro-average F1 score (Macro-F1). These metrics have been widely used in previous studies and provide valuable insights into the performance of SGNN models. Lower values of these metrics indicate poorer model performance and greater effectiveness of attack methods. However, we find that the area under the curve (AUC) metric may not be suitable for assessing the performance of models on signed graph datasets. AUC tends to yield misleading results on imbalanced datasets, which is the

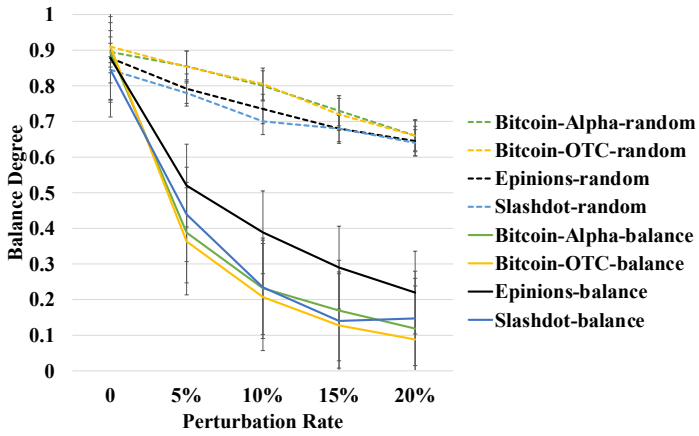

Figure 6.1: Balance Degree of 4 Datasets under 2 Attacks.

case for signed graph datasets that predominantly contain positive edges. Therefore, we exclude the AUC metric from our evaluation.

6.2.2 Defense Setup

To assess the effectiveness of our approach, we employ four commonly used metrics: AUC, Micro-F1, Binary-F1, and Macro-F1. These metrics have been widely utilized in previous studies. Higher values represent better performance. By convention, the best comparison results are denoted in bold, while the second-best results are underlined, unless otherwise specified.

For adversarial attacks and SGNN models, we utilize the parameters specified in their respective papers. Our proposed model, BA-SGCL, is implemented using PyTorch, with a learning rate set to 0.001. To ensure an adequate balance degree for the positive view, we impose a lower limit of 0.9 (within the range of 0 to 1).

6.3 Baselines

6.3.1 Attack Baselines

With the above benchmark datasets, we evaluate balance-attack on five popular SGNN models, as follows:

- **SGCN** [13] aims to bridge the gap between unsigned GCN and the analysis of signed graphs. It strives to develop a novel information aggregator by leveraging balance theory, thereby extending the applicability of GCN to signed graphs.
- **SGCL** [65] is the first work to generalize graph contrastive learning to signed graphs, which employs graph augmentations to reduce the harm of noisy interactions and enhances the model robustness.
- **SDGNN** [33] combines both balance theory and status theory, and introduces four weight matrices to aggregate neighbor features based on edge types.
- **RSGNN** [91] incorporates structure-based regularizers to enhance the performance of SGNNs by emphasizing the intrinsic properties of a signed graph and mitigating their vulnerability to potential edge noise in the input graph.
- **UGCL** [42] introduces a novel contrastive learning framework that incorporates Laplacian perturbation, offering a unique advantage through the utilization of an indirect perturbation method that ensures stability and maintains effective perturbation effects.

6.3.2 Defense Baselines

To establish a baseline for comparison, we employ a random attack strategy since there is currently no established black-box attack model specifically designed for signed

graphs. The applicability of unsigned graph methods [88] [104] to signed graphs is limited due to their strong dependence on node labels and node features, rendering them unsuitable for the present scenario. In the case of the random attack, we randomly select a set of edges from the input signed graph and flip their signs.

We evaluate attacks on nine popular SGNN models, which are categorized as with / without attack-tolerant signed graph representation learning (attack-tolerant SNE/SNE): SiNE [74], SGCN [13], SNEA [51], BESIDE [11], SDGNN [33] and SDGCN [41] are methods without attack-tolerant proporties. RSGNN [91], SGCL [65] and UGCL [42] are attack-tolerant methods.

- **SiNE** [74] is a signed graph embedding method that uses deep neural networks and extended structural balance theory-based loss function.
- **SGCN** [13] introduces a novel information aggregator based on balance theory, expanding the application of GCN to signed graphs.
- **SNEA** [51] generalizes Graph Attention Network (GAT) [73] to signed graphs and is also based on the balance theory.
- **BESIDE** [11] combines balance and status theory. It utilizes status theory to learn "bridge" edge information and combines it with triangle information.
- **SGCL** [65] is the first work to generalize GCL to signed graphs.
- **SDGNN** [33] combines both balance and status theory, and introduces four weight matrices to aggregate neighbor features based on edge types.
- **RSGNN** [91] improves SGNN performance by using structure-based regularizers to highlight the intrinsic properties of signed graphs and reduce vulnerability to input graph noise.
- **SDGCN** [41] defines a spectral graph convolution encoder with a magnetic Laplacian.

- **UGCL** [42] presents a GCL framework that incorporates Laplacian perturbation.

We utilize two adversarial attacks on signed graphs:

- **balance-attack** [97]: A type of black-box attack that effectively reduces the degree of balance in signed graphs. The author's rationale stems from the observation that signed graphs commonly exhibit a high balance degree, leading them to propose reducing the balance degree as a means to achieve the desired effect of adversarial attacks.
- **FlipAttack** [101]: An adversarial attack method against trust prediction in signed graphs, which can effectively downgrade the classification performances for typical machine learning models. The attack was first designed to target representative trust prediction models, formulating it as a hard bi-level optimization problem. The attack was further refined by integrating conflicting metrics as penalty terms into the objective function, resulting in secrecy-awareness.

Both methods manipulate signed graphs by modifying the sign information while preserving the topological information. Furthermore, due to the constraints imposed by the attack methods, excessively large datasets (such as Slashdot and Epinions) necessitate testing using subsets. In our experimental setup, we iteratively extract a subgraph comprising 2000 nodes as a representative dataset, which is subsequently partitioned into training and testing sets in a proportional manner.

6.4 Balance Degree of Signed Graphs after Attacks (Q1)

To validate the effectiveness of our method, we first apply our approach and obtain conclusive results: the balance degree of signed graphs is significantly reduced com-

pared to the balance degree under the random attack. We present the comparison results of the balance-attack and random attack in Fig. 6.1. Initially, in each dataset, the balance degree ranges from 0.85 to 0.9. When subjected to random attacks with a perturbation rate of 20%, the minimum balance degree drops to approximately 0.65. However, by utilizing our designed balance-attack method with a perturbation rate of 5%, the balance degree becomes more lower, ranging between 0.35 and 0.55. Furthermore, at a perturbation rate of 20%, the balance degree can be further reduced to about 0.1, which is significantly lower than what is achieved through random attacks. These results unequivocally demonstrate the effectiveness of our proposed method in significantly reducing the balance degree of the graph.

6.5 Attack Performance of balance-attack (Q2)

We conduct a comparative analysis between random attack and balance-attack on five existing SGNN models. To evaluate their performance, we tested the models at perturbation rates from 0% to 20% based on the three metrics mentioned before to evaluate the attack performance. RSGNN is a model known for its resilience against random attacks. While its original design may not have explicitly focused on adversarial attacks, we can infer that it possesses greater robustness against various attack scenarios compared to other SGNN models. Based on the results in Tab. 6.2, it is evident that RSGNN can maintain satisfactory performance even when subjected to random attacks. However, when exposed to our balance-attack, the performance of RSGNN experiences a significant decline. Similar results are observed in the other four SGNN models as presented in Tab. 6.3 and Tab. 6.4.

6.6 Applicability of balance-attack on various SGNNs (Q3)

In addition to assessing balance-theory-based models, we also evaluate the performance of our proposed attack on non-balance-based models (i.e. UGCL). Even though our attack method is designed based on the intuition that many SGNNs rely on balance theory, we surprisingly find that it also proves to be effective against non-balance-based SGNNs. This showcases the versatility and efficacy of balance-attack across different SGNNs.

6.7 Defense Performance against Attacks (Q4)

To answer **Q4**, we first use the attack methods to poison the graph, varying the perturbation ratio within the range of 0 to 20%. These two attacks are global attacks and can flip edge signs, without any capability to add or delete edges. The perturbation ratio represents the proportion of edges that can change their signs within the entire set of edges. We then train models on the poisoned graph and evaluate the link sign prediction performance achieved by these methods.

Link sign prediction results with AUC and Macro-F1 under adversarial attacks for BA-SGCL and other baselines are shown in Tab. 6.5 and Tab. 6.6. Micro-F1 and Binary-F1 are shown in Tab, 6.7 and Tab. 6.8. Notably, both AUC and Macro-F1 exhibit significant improvements in comparison to other baselines. Regarding Micro-F1 and Binary-F1, their performance closely aligns with that of the current state-of-the-art models.

We make the following observations from the results. Firstly, existing SGNNs degrade significantly under various attacks, while our model maintains high performance with minimal degradation, indicating its robustness. Secondly, RSGNN performs well

against random attacks but exhibits weaker defense against adversarial attacks, likely due to the direct enhancement of the balance degree, which encounters the Irreversibility of Balance-related Information challenge. Thirdly, even with an attack rate of 0, BA-SGCL outperforms other GCL models, attributed to guided balance augmentation for capturing more graph invariance and obtaining higher-quality embeddings.

6.8 Analysis of Balance Augmentation (Q5)

To evaluate the effectiveness of balance augmentation, we compare BA-SGCL with a control model called random-SGCL. In random-SGCL, sign perturbation is applied to an augmented view, randomly changing the sign of links. The another view remains unchanged. For the remaining parts, we adopt the same settings as BA-SGCL. The remaining components of random-SGCL are the same as those in BA-SGCL. Detailed results of the effectiveness of balance augmentation under balance-attack and FlipAttack are shown in Tab. 6.9 and Tab. 6.10, respectively. The superior performance of BA-SGCL in comparison to random-SGCL is evident. These results clearly demonstrate the effectiveness of our balance augmentation.

6.9 Ablation Study

To confirm that the improved robust performance of our model is a result of the combination of the GCL framework and balance augmentation instead of the SDGCN encoder, we conduct experiments where we replaced the SDGCN encoder with other encoders, such as SGCN [13], while keeping the remaining modules unchanged. The performance comparison between BA-SGCL using SGCN encoder with the original SGCN model under balance-attack are shown in Tab. 6.11 and Tab. 6.12, respectively. More comprehensive results are shown in Appendix C. It is evident that the performance of BA-SGCL (SGCN encoder) significantly outperforms that of the

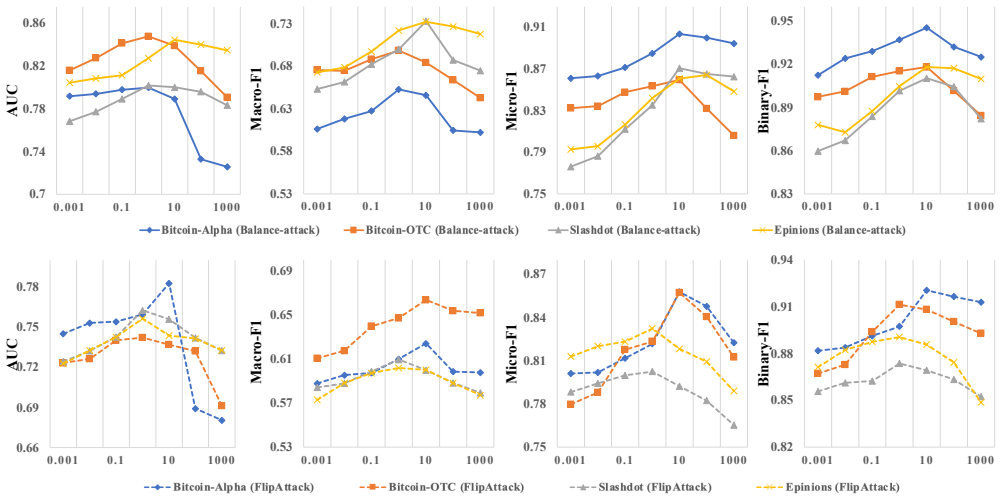

Figure 6.2: Parameter Analysis with Perturbation Rate = 10%. The first line is under balance-attack, the second line is under FlipAttack.

SGCN model, thus affirming the efficacy of our proposed GCL framework and the balance augmentation technique.

6.10 Parameter Analysis

We explore the sensitivity of hyper-parameters α in the loss function. Our objective is to examine how varying the value of α can impact the performance of BA-SGCL. Specifically, we experiment with different hyper-parameter values ranging from 1e-3 to 1e3. The performance of BA-SGCL under attacks with a perturbation rate of 10% is depicted in Fig. 6.2. The parameter analysis test with a perturbation rate equals to 20% is shown in Fig. 6.3. Considering four metrics, we conclude that proper settings of the parameter α can enhance the performance of BA-SGCL. However, if the value of α is too small or too large, it can significantly impair the model's performance.

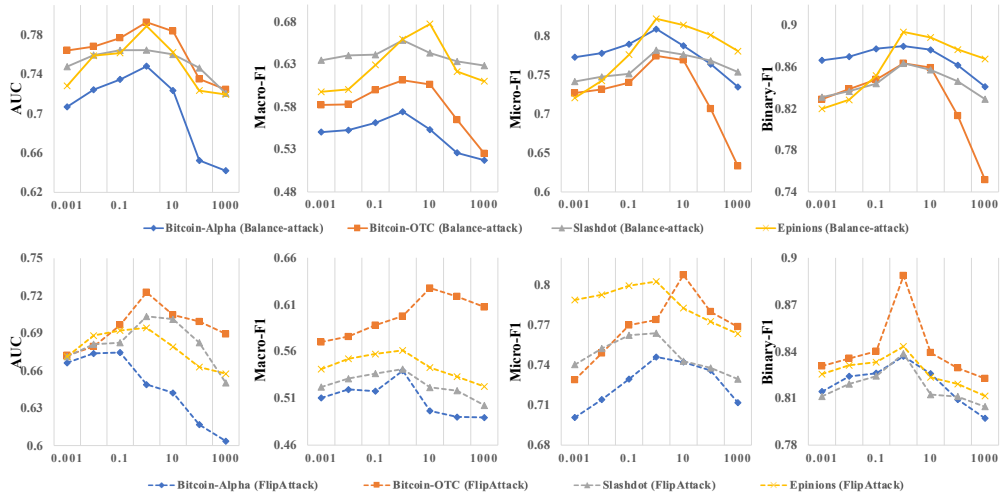

Figure 6.3: Parameter Analysis with Perturbation Rate = 20%. The first line is under balance-attack, the second line is under FlipAttack.

Table 6.2: Link Sign Prediction Performance of RSGNN under Random Attack and balance-attack

Dataset	Ptb	Attack	Micro_f1	Binary_f1	Macro_f1
Bitcoin-Alpha	0	-	0.8820	0.9341	0.6841
	10%	ranodm	0.7726	0.8642	0.5831
		balance	**0.6802**	**0.7984**	**0.5123**
	20%	random	0.6839	0.8010	0.5165
		balance	**0.6308**	**0.7631**	**0.4635**
Bitcoin-OTC	0	-	0.8919	0.9382	0.7553
	10%	random	0.8242	0.8950	0.6782
		balance	**0.7134**	**0.8158**	**0.5849**
	20%	random	0.7828	0.8673	0.6341
		balance	**0.6424**	**0.7625**	**0.5194**
Slashdot	0	-	0.7823	0.8574	0.6988
	10%	random	0.7092	0.7982	0.6390
		balance	**0.6719**	**0.7761**	**0.5813**
	20%	random	0.6637	0.7576	0.6044
		balance	**0.6009**	**0.7165**	**0.5215**
Epinions	0	-	0.8280	0.8932	0.7261
	10%	random	0.7711	0.8516	0.6754
		balance	**0.7342**	**0.8234**	**0.6432**
	20%	random	0.7409	0.8285	0.6492
		balance	**0.6832**	**0.7836**	**0.5962**

Table 6.3: Link Sign Prediction Performance of UGCL and SGCL under Random Attack and balance-attack with Perturbation Rate = 20%

Model	Dataset	Attack	Micro_F1	Binary_F1	Macro_F1
UGCL	Bitcoin-Alpha	random	0.9199	0.9576	0.6192
		balance	**0.8044**	**0.8883**	**0.5526**
	Bitcoin-OTC	random	0.8988	0.9442	0.6983
		balance	**0.7752**	**0.8643**	**0.6044**
	Slashdot	random	0.8538	0.9173	0.6318
		balance	**0.7826**	**0.8704**	**0.5971**
	Epinions	random	0.8635	0.9237	**0.6390**
		balance	**0.8328**	**0.9018**	0.6665
SGCL	Bitcoin-Alpha	random	0.9305	0.9636	0.6007
		balance	**0.8108**	**0.8931**	**0.5312**
	Bitcoin-OTC	random	0.9026	0.9480	0.6131
		balance	**0.7931**	**0.8785**	**0.5919**
	Slashdot	random	0.8338	0.9072	0.5578
		balance	**0.7002**	**0.8163**	**0.5001**
	Epinions	random	0.8482	0.9160	**0.5673**
		balance	**0.7385**	**0.8371**	0.5872

Table 6.4: Link Sign Prediction Performance of SDGNN and SGCN under Random Attack and balance-attack with Perturbation Rate = 20%

Model	Dataset	Attack	Micro_F1	Binary_F1	Macro_F1
SDGNN	Bitcoin-Alpha	random	0.8616	0.9234	0.6062
		balance	**0.7775**	**0.8698**	**0.5528**
	Bitcoin-OTC	random	0.8333	0.9028	0.6593
		balance	**0.7388**	**0.8371**	**0.5893**
	Slashdot	random	0.8405	0.8981	0.6966
		balance	**0.7326**	**0.8286**	**0.6106**
	Epinions	random	0.8336	0.9023	0.6714
		balance	**0.7696**	**0.8550**	**0.6467**
SGCN	Bitcoin-Alpha	random	0.6614	0.7842	0.4991
		balance	**0.6022**	**0.7346**	**0.4704**
	Bitcoin-OTC	random	0.6833	0.7925	0.5620
		balance	**0.6265**	**0.7434**	**0.5288**
	Slashdot	random	0.6835	0.7752	0.6204
		balance	**0.5939**	**0.7029**	**0.5307**
	Epinions	random	0.6725	0.7712	0.5977
		balance	**0.6453**	**0.7497**	**0.5706**

Table 6.5: AUC and Macro-F1 of SGNNs on Link Sign Prediction under balance-attack

	Ptb(%)	SNE												Attack-tolerant SNE						Proposed	
		SiNE		SGCN		SNEA		BESIDE		SDGNN		SDGCN		RSGNN		SGCL		UGCL		BA-SGCL	
Metrics		AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1
BitcoinAlpha	0	0.8109	0.6712	0.7992	0.6656	0.8013	0.6724	0.8635	0.7106	0.8552	0.7145	0.8598	0.7201	0.8034	0.6841	0.8491	0.7121	0.8645	0.7351	**0.8948**	**0.7772**
	5	0.7412	0.5811	0.7354	0.5648	0.7421	0.5794	0.7972	0.6332	0.8038	0.6481	0.8002	0.6425	0.7454	0.5848	0.8074	0.6546	0.8266	0.6729	**0.8461**	**0.6908**
	10	0.6879	0.5212	0.6913	0.5125	0.6994	0.5231	0.7569	0.6047	0.7791	0.5744	0.7782	0.5689	0.7063	0.5323	0.7614	0.6056	0.7843	0.6225	**0.7997**	**0.6527**
	15	0.6715	0.4911	0.6855	0.4836	0.6912	0.4986	0.7328	0.5495	0.7466	0.5478	0.7431	0.5397	0.6199	0.5014	0.7256	0.5681	0.7485	0.5777	**0.7716**	**0.5994**
	20	0.6512	0.4755	0.6535	0.4704	0.6693	0.4881	0.6977	0.5139	0.7186	0.5128	0.6925	0.5078	0.6025	0.4835	0.6919	0.5312	0.7244	0.5426	**0.7479**	**0.5742**
BitcoinOTC	0	0.8211	0.7612	0.8253	0.7501	0.8304	0.7613	0.8858	0.7607	0.8962	0.7511	0.8846	0.7638	0.8171	0.7553	0.8931	0.7711	0.8942	0.7801	**0.9108**	**0.8079**
	5	0.7613	0.6612	0.7753	0.6471	0.7814	0.6514	0.8373	0.7236	0.8565	0.6928	0.8412	0.6847	0.7954	0.6574	0.8458	0.7218	0.8602	0.7559	**0.8774**	**0.7814**
	10	0.7494	0.6232	0.7504	0.6142	0.7511	0.6287	0.8039	0.6455	0.8266	0.6327	0.8104	0.6385	0.7452	0.5849	0.8091	0.6611	0.8329	0.6863	**0.8476**	**0.6984**
	15	0.7151	0.5798	0.7271	0.5814	0.7351	0.5934	0.7713	0.5914	0.7969	0.5824	0.7859	0.5747	0.6923	0.5486	0.7827	0.6091	0.7951	0.6402	**0.8134**	**0.6456**
	20	0.6812	0.5589	0.6985	0.5621	0.7045	0.5698	0.7411	0.5565	0.7561	0.5393	0.7305	0.5238	0.6603	0.5194	0.7407	0.5715	0.7654	0.5944	**0.7926**	**0.6111**
Slashdot	0	0.8214	0.6812	0.8153	0.6834	0.8296	0.6945	0.8389	0.7092	0.8909	0.7203	0.8933	0.7298	0.7829	0.6988	0.8848	0.6874	0.8885	0.7375	**0.8956**	**0.7549**
	5	0.7322	0.6328	0.7432	0.6332	0.7524	0.6473	0.7832	0.6873	0.8285	0.6867	0.8015	0.6774	0.7184	0.6484	0.8155	0.6506	0.8479	0.6948	**0.8561**	**0.7485**
	10	0.6912	0.5811	0.6892	0.5719	0.6998	0.5824	0.7622	0.6743	0.7691	0.6334	0.7406	0.6257	0.6564	0.5813	0.7465	0.5611	0.7771	0.6658	**0.8017**	**0.7329**
	15	0.6412	0.5412	0.6497	0.5404	0.6591	0.5563	0.7393	0.6439	0.7391	0.5989	0.7098	0.5825	0.6377	0.5557	0.6918	0.5009	0.7365	0.6397	**0.7701**	**0.6842**
	20	0.6211	0.5164	0.6348	0.5207	0.6415	0.5258	0.7152	0.6057	0.6972	0.5706	0.6706	0.5669	0.5976	0.5215	0.6589	0.5001	0.6911	0.5871	**0.7643**	**0.6581**
Epinions	0	0.7911	0.6847	0.7763	0.6957	0.7912	0.6998	0.8575	0.7104	0.8591	0.7141	0.8613	0.6784	0.7821	0.7161	0.8512	0.7155	0.8723	0.6861	**0.8731**	**0.7301**
	5	0.7815	0.6501	0.7711	0.6603	0.7833	0.6724	0.8074	0.6958	0.8261	0.7032	0.8052	0.6583	0.7535	0.6739	0.8034	0.6661	0.8352	0.6846	**0.8523**	**0.7203**
	10	0.7421	0.6114	0.7383	0.6125	0.7421	0.6231	0.7478	0.6605	0.7981	0.6898	0.7823	0.6366	0.7419	0.6432	0.7881	0.6536	0.8127	0.6789	**0.8444**	**0.7321**
	15	0.7273	0.5889	0.7142	0.5843	0.7156	0.5895	0.7203	0.6339	0.7812	0.6598	0.7653	0.6105	0.7257	0.6201	0.7442	0.6165	0.7871	0.6632	**0.8037**	**0.7032**
	20	0.6912	0.5712	0.6881	0.5606	0.6891	0.5679	0.6997	0.6075	0.7583	0.6367	0.7424	0.6075	0.6981	0.5962	0.7136	0.5872	0.7711	0.6465	**0.7887**	**0.6772**

Table 6.6: AUC and Macro-F1 of SGNNs on Link Sign Prediction under FlipAttack

	Ptb(%)	SNE												Attack-tolerant SNE						Proposed	
		SiNE		SGCN		SNEA		BESIDE		SDGNN		SDGCN		RSGNN		SGCL		UGCL		BA-SGCL	
Metrics		AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1	AUC	Macro-F1
BitcoinAlpha	0	0.8109	0.6712	0.7992	0.6656	0.8013	0.6724	0.8635	0.7106	0.8552	0.7145	0.8598	0.7201	0.8034	0.6841	0.8491	0.7121	0.8645	0.7351	**0.8948**	**0.7772**
	5	0.7136	0.5618	0.7114	0.5782	0.7223	0.5892	0.7886	0.6549	0.7485	0.6312	0.7428	0.6314	0.7301	0.5847	0.7749	0.6328	0.7991	0.6597	**0.8204**	**0.6708**
	10	0.6854	0.5764	0.6746	0.5543	0.6885	0.5623	0.7441	0.5987	0.7252	0.5775	0.7395	0.5783	0.6908	0.5431	0.7202	0.5579	0.7565	0.5972	**0.7828**	**0.6241**
	15	0.5913	0.4923	0.5984	0.4987	0.6014	0.5038	0.6651	0.5721	0.6614	0.5632	0.6639	0.5682	0.6261	0.5057	0.6598	0.5139	0.7081	0.5671	**0.7299**	**0.5901**
	20	0.5714	0.4412	0.5601	0.4322	0.5772	0.4423	0.6404	0.5197	0.6325	0.5088	0.6258	0.5082	0.5783	0.4574	0.5888	0.4801	0.6293	0.5047	**0.6745**	**0.5395**
BitcoinOTC	0	0.8211	0.7612	0.8253	0.7501	0.8304	0.7613	0.8858	0.7607	0.8962	0.7511	0.8846	0.7638	0.8171	0.7553	0.8931	0.7711	0.8942	0.7801	**0.9108**	**0.8079**
	5	0.7301	0.6434	0.7397	0.6538	0.7419	0.6634	0.7871	0.6875	0.7746	0.6755	0.7824	0.6792	0.7447	0.6799	0.7721	0.6885	0.7836	0.6849	**0.8005**	**0.7056**
	10	0.6905	0.5996	0.6844	0.6086	0.6924	0.6134	0.7066	0.6117	0.7086	0.5987	0.7123	0.5983	0.7013	0.6278	0.6817	0.6014	0.7233	0.6298	**0.7442**	**0.6636**
	15	0.6598	0.5596	0.6518	0.5639	0.6634	0.5698	0.6953	0.5919	0.6889	0.5848	0.6942	0.5912	0.6741	0.5856	0.6647	0.5895	0.7117	0.5986	**0.7413**	**0.6333**
	20	0.6424	0.5587	0.6315	0.5512	0.6412	0.5634	0.6508	0.5822	0.6714	0.5813	0.6739	0.5821	0.6677	0.5732	0.6438	0.5628	0.7037	0.5762	**0.7227**	**0.6277**
Slashdot	0	0.8214	0.6812	0.8153	0.6834	0.8296	0.6945	0.8389	0.7092	0.8909	0.7203	0.8933	0.7298	0.7829	0.6988	0.8848	0.6874	0.8885	0.7375	**0.8956**	**0.7549**
	5	0.7221	0.6024	0.7193	0.6092	0.7285	0.6124	0.7812	0.6356	0.7695	0.6234	0.7584	0.6173	0.7253	0.6118	0.7735	0.6301	0.7864	0.6421	**0.8014**	**0.6623**
	10	0.6398	0.5279	0.6412	0.5212	0.6593	0.5331	0.7395	0.5689	0.7193	0.5589	0.7158	0.5498	0.6608	0.5323	0.7214	0.5665	0.7455	0.5812	**0.7626**	**0.6094**
	15	0.6195	0.5027	0.6128	0.5058	0.6245	0.5285	0.6914	0.5582	0.6819	0.5452	0.6845	0.5412	0.6274	0.5112	0.6745	0.5412	0.6979	0.5633	**0.7234**	**0.5757**
	20	0.6124	0.4681	0.6041	0.4776	0.6184	0.4872	0.6725	0.5193	0.6519	0.4989	0.6537	0.5028	0.6235	0.4898	0.6596	0.5025	0.6803	0.5234	**0.7035**	**0.5412**
Epinions	0	0.7911	0.6847	0.7763	0.6957	0.7912	0.6998	0.8575	0.7104	0.8591	0.7141	0.8613	0.6784	0.7821	0.7161	0.8512	0.7155	0.8723	0.6861	**0.8731**	**0.7301**
	5	0.7113	0.5789	0.7124	0.5824	0.7296	0.5983	0.7764	0.6387	0.7437	0.6358	0.7363	0.6382	0.7387	0.6121	0.7765	0.6453	0.7889	0.6412	**0.7967**	**0.6554**
	10	0.6814	0.5168	0.6898	0.5215	0.6934	0.5353	0.7214	0.5885	0.7355	0.5712	0.7295	0.5872	0.7146	0.5524	0.7216	0.5812	0.7311	0.5898	**0.7564**	**0.6018**
	15	0.6592	0.4919	0.6698	0.4989	0.6774	0.5034	0.6934	0.5415	0.7044	0.5339	0.6985	0.5389	0.6822	0.5123	0.6977	0.5416	0.7012	0.5543	**0.7219**	**0.5718**
	20	0.6087	0.4996	0.6149	0.4884	0.6255	0.4982	0.6611	0.5268	0.6708	0.5279	0.6684	0.5287	0.6435	0.5051	0.6598	0.5331	0.6784	0.5413	**0.6943**	**0.5612**

Table 6.7: Micro-F1 and Binary-F1 of SGNNs on Link Sign Prediction under balance-attack

	Ptb(%)	SNE												Attack-tolerant SNE						Proposed	
		SiNE		SGCN		SNEA		BESIDE		SDGNN		SDGCN		RSGNN		SGCL		UGCL		BA-SGCL	
	Metrics	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1
BitcoinAlpha	0	0.8495	0.8953	0.8525	0.9041	0.8613	0.9188	0.9291	0.9497	0.9114	0.9443	0.9216	0.9475	0.8821	0.9341	0.9231	0.9591	0.9481	0.9731	**0.9526**	**0.9751**
	5	0.7133	0.8246	0.7263	0.8134	0.7325	0.8276	0.8954	0.9216	0.8754	0.9241	0.8739	0.9275	0.7563	0.8534	0.9165	0.9353	0.9215	0.9579	**0.9297**	**0.9626**
	10	0.6598	0.7574	0.6602	0.7684	0.6737	0.7758	0.8584	0.9144	0.8303	0.8981	0.8219	0.8895	0.6802	0.7984	0.8848	0.9175	0.8838	0.9364	**0.9037**	**0.9454**
	15	0.6154	0.7485	0.6205	0.7562	0.6309	0.7633	0.8082	0.8986	0.7961	0.8712	0.7889	0.8693	0.6605	0.7862	**0.8565**	0.9008	0.8452	0.9135	0.8435	**0.9166**
	20	0.5989	0.7296	0.6022	0.7346	0.6137	0.7487	0.7635	0.8797	0.7375	0.8398	0.7329	0.8275	0.6308	0.7631	0.8051	0.8731	0.7993	0.8751	**0.8088**	**0.8802**
BitcoinOTC	0	0.8681	0.9184	0.8791	0.9296	0.8824	0.9374	0.9129	0.9454	0.9044	0.9439	0.9127	0.9397	0.8919	0.9382	0.9202	0.9564	0.9361	0.9651	**0.9383**	**0.9662**
	5	0.7739	0.8526	0.7829	0.8661	0.7959	0.8775	0.8845	0.9357	0.8677	0.9224	0.8648	0.9276	0.7971	0.8761	0.9074	0.9447	0.9156	0.9531	**0.9224**	**0.9571**
	10	0.7312	0.8301	0.7474	0.8409	0.7532	0.8537	0.8269	0.9046	0.8143	0.8891	0.8048	0.8843	0.7134	0.8158	0.8549	0.9194	**0.8669**	**0.9239**	0.8596	0.9189
	15	0.6946	0.8097	0.7066	0.8104	0.7159	0.8259	0.7878	0.8715	0.7651	0.8569	0.7662	0.8573	0.6787	0.7909	0.8135	0.8946	**0.8234**	**0.8964**	0.8134	0.8896
	20	0.6724	0.7846	0.6833	0.7925	0.6983	0.8094	0.7409	0.8421	0.7088	0.8171	0.7175	0.8275	0.6424	0.7625	0.7731	**0.8665**	**0.7752**	0.8643	0.7743	0.8636
Slashdot	0	0.8012	0.8614	0.8127	0.8788	0.8235	0.8853	0.8549	0.9145	0.8698	0.9295	0.8648	0.9275	0.7823	0.8574	0.8739	0.9291	0.8791	0.9297	**0.8792**	**0.9301**
	5	0.7015	0.7988	0.7199	0.8091	0.7206	0.8137	0.8364	0.8789	0.8578	0.9161	0.8426	0.9086	0.7444	0.8225	0.8281	0.8995	0.8732	**0.9278**	**0.8747**	0.9269
	10	0.6412	0.7461	0.6518	0.7528	0.6648	0.7648	0.8094	0.8512	0.8056	0.8812	0.7984	0.8775	0.6719	0.7761	0.7572	0.8553	0.8533	**0.9156**	**0.8708**	0.9103
	15	0.6012	0.6914	0.6025	0.7093	0.6196	0.7183	0.7839	0.8215	0.7704	0.8568	0.7637	0.8429	0.6378	0.7466	0.7221	0.8331	0.8301	0.9011	**0.8391**	**0.9054**
	20	0.5859	0.6911	0.5939	0.7029	0.6069	0.7138	0.7426	0.8014	0.7326	0.8286	0.7286	0.8119	0.6009	0.7165	0.7002	0.8163	**0.7826**	**0.8704**	0.7817	0.8637
Epinions	0	0.8095	0.8724	0.8181	0.8863	0.8285	0.8964	0.8549	0.9153	0.8727	0.9265	0.8637	0.9258	0.8281	0.8932	0.8673	0.9232	**0.8762**	**0.9304**	0.8742	0.9279
	5	0.7624	0.8448	0.7734	0.8528	0.7849	0.8637	0.8366	0.9005	0.8618	0.9196	0.8533	0.9074	0.7736	0.8542	0.8478	0.9125	**0.8726**	**0.9281**	0.8718	0.9279
	10	0.6914	0.7871	0.7038	0.7976	0.7119	0.8094	0.7939	0.8814	0.8374	0.9032	0.8229	0.9054	0.7342	0.8234	0.8218	0.8951	**0.8679**	**0.9249**	0.8608	0.9181
	15	0.6616	0.7689	0.6711	0.7707	0.6839	0.7843	0.7695	0.8412	0.8097	0.8837	0.7974	0.8854	0.7068	0.8016	0.7758	0.8636	**0.8532**	**0.9152**	0.8451	0.9084
	20	0.6374	0.7443	0.6453	0.7497	0.6576	0.7586	0.7332	0.8141	0.7696	0.8551	0.7527	0.8496	0.6832	0.7836	0.7385	0.8371	**0.8328**	**0.9018**	0.8219	0.8938

Table 6.8: Micro-F1 and Binary-F1 of SGNNs on Link Sign Prediction under FlipAttack

	Ptb(%)	SNE												Attack-tolerant SNE						Proposed	
		SiNE		SGCN		SNEA		BESIDE		SDGNN		SDGCN		RSGNN		SGCL		UGCL		BA-SGCL	
	Metrics	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1	Micro-F1	Binary-F1
BitcoinAlpha	0	0.8495	0.8953	0.8525	0.9041	0.8613	0.9188	0.9291	0.9497	0.9114	0.9443	0.9216	0.9475	0.8821	0.9341	0.9231	0.9591	0.9481	0.9731	**0.9526**	**0.9751**
	5	0.7295	0.8342	0.7259	0.8293	0.7312	0.8388	0.8533	0.9124	0.8331	0.9021	0.8315	0.9123	0.7502	0.8469	0.8675	0.9165	**0.8816**	**0.9265**	0.8741	0.9212
	10	0.6724	0.8023	0.6747	0.7982	0.6884	0.8044	0.8285	0.8792	0.8173	0.8592	0.8042	0.8758	0.6992	0.8102	0.8201	0.9045	**0.8599**	**0.9222**	0.8577	0.9205
	15	0.6399	0.7721	0.6315	0.7632	0.6448	0.7735	0.7835	0.8624	0.7667	0.8425	0.7523	0.8315	0.6645	0.7859	0.7706	0.8733	**0.8166**	**0.8994**	0.8152	0.8943
	20	0.5724	0.7124	0.5798	0.7074	0.5872	0.7263	0.7386	0.8315	0.7123	0.8123	0.7045	0.8093	0.5909	0.7266	0.6913	0.8183	0.7458	0.8346	**0.7459**	**0.8369**
BitcoinOTC	0	0.8681	0.9184	0.8791	0.9296	0.8824	0.9374	0.9129	0.9454	0.9044	0.9439	0.9127	0.9397	0.8919	0.9382	0.9202	0.9564	0.9361	0.9651	**0.9383**	**0.9662**
	5	0.7684	0.8583	0.7786	0.8614	0.7985	0.8776	0.8224	0.9036	0.8124	0.9195	0.8068	0.9112	0.8009	0.8767	0.8416	0.9033	0.8505	0.9113	**0.8601**	**0.9188**
	10	0.7244	0.8278	0.7238	0.8253	0.7474	0.8369	0.8123	0.8787	0.8012	0.8824	0.7998	0.8775	0.7502	0.8413	0.7939	0.8843	0.8484	0.9049	**0.8572**	**0.9113**
	15	0.6682	0.7748	0.6713	0.7764	0.6835	0.7894	0.7785	0.8649	0.7662	0.8535	0.7524	0.8438	0.6963	0.7998	0.7791	0.8724	0.8144	**0.8889**	**0.8181**	0.8864
	20	0.6489	0.7512	0.6425	0.7597	0.6553	0.7627	0.7643	0.8327	0.7464	0.8251	0.7354	0.8048	0.6781	0.7848	0.7348	0.8425	0.8037	0.8812	**0.8072**	**0.8885**
Slashdot	0	0.8012	0.8614	0.8127	0.8788	0.8235	0.8853	0.8549	0.9145	0.8698	0.9295	0.8648	0.9275	0.7823	0.8574	0.8739	0.9291	0.8791	0.9297	**0.8792**	**0.9301**
	5	0.7149	0.7424	0.7086	0.7592	0.7186	0.7626	0.7747	0.8616	0.7798	0.8637	0.7685	0.8548	0.7103	0.7797	0.8034	0.8557	**0.8243**	0.8797	0.8206	**0.8842**
	10	0.6736	0.7419	0.6867	0.7475	0.6943	0.7583	0.7718	0.8512	0.7626	0.8537	0.7535	0.8386	0.7023	0.7635	0.7942	0.8521	**0.8114**	0.8716	0.8024	**0.8736**
	15	0.6695	0.7286	0.6642	0.7329	0.6708	0.7428	0.7538	0.8379	0.7429	0.8398	0.7388	0.8181	0.6887	0.7532	0.7732	0.8328	0.7809	0.8465	**0.7885**	**0.8573**
	20	0.6264	0.7018	0.6286	0.7092	0.6378	0.7212	0.7274	0.8092	0.7219	0.8217	0.7266	0.8007	0.6532	0.7328	0.7443	0.8192	0.7574	0.8321	**0.7637**	**0.8386**
Epinions	0	0.8095	0.8724	0.8181	0.8863	0.8285	0.8964	0.8549	0.9153	0.8727	0.9265	0.8637	0.9258	0.8281	0.8932	0.8673	0.9232	**0.8762**	**0.9304**	0.8742	0.9279
	5	0.7143	0.7715	0.7125	0.7768	0.7244	0.7886	0.8189	0.8637	0.8143	0.8662	0.8023	0.8629	0.7386	0.7958	0.8234	0.8635	0.8319	**0.8986**	**0.8352**	0.8975
	10	0.7017	0.7627	0.7098	0.7635	0.7212	0.7759	0.7982	0.8521	0.8023	0.8511	0.7982	0.8476	0.7328	0.7837	0.8026	0.8529	0.8278	**0.8946**	**0.8323**	0.8905
	15	0.6935	0.7378	0.7023	0.7441	0.7145	0.7528	0.7989	0.8314	0.7993	0.8392	0.7928	0.8242	0.7148	0.7648	0.8015	0.8321	**0.8253**	0.8726	0.8213	**0.8774**
	20	0.6889	0.7219	0.6835	0.7289	0.6982	0.7375	0.7823	0.8291	0.7723	0.8272	0.7635	0.8192	0.6983	0.7456	0.7864	0.8265	0.7992	0.8375	**0.8023**	**0.8432**

Table 6.9: Effectiveness of Balance Augmentation under balance-attack

Dataset	Ptb(%)	random-SGCL				BA-SGCL			
		AUC	Macro-F1	Micro-F1	Binary-F1	AUC	Macro-F1	Micro-F1	Binary-F1
BitcoinAlpha	0	0.8363	0.7258	0.9221	0.9601	**0.8948**	**0.7772**	**0.9526**	**0.9751**
	10	0.7692	0.6123	0.8778	0.9103	**0.7997**	**0.6527**	**0.9037**	**0.9454**
	20	0.7034	0.5401	0.8005	0.8762	**0.7479**	**0.5742**	**0.8088**	**0.8802**
BitconOTC	0	0.8894	0.7723	0.9189	0.9587	**0.9108**	**0.8079**	**0.9383**	**0.9662**
	10	0.8023	0.6632	0.8541	0.9083	**0.8476**	**0.6984**	**0.8596**	**0.9189**
	20	0.7539	0.5885	0.7724	0.8623	**0.7926**	**0.6111**	**0.7743**	**0.8636**
Slashdot	0	0.8814	0.6889	0.8746	0.9223	**0.8956**	**0.7549**	**0.8792**	**0.9301**
	10	0.7498	0.5789	0.7982	0.8779	**0.8017**	**0.7329**	**0.8708**	**0.9103**
	20	0.6798	0.5122	0.7652	0.8523	**0.7643**	**0.6581**	**0.7817**	**0.8637**
Epinions	0	0.8582	0.7123	0.8698	0.9223	**0.8731**	**0.7301**	**0.8742**	**0.9279**
	10	0.7943	0.6579	**0.8625**	0.9123	**0.8444**	**0.7321**	0.8608	**0.9181**
	20	0.7331	0.6293	**0.8241**	**0.8996**	**0.7887**	**0.6772**	0.8219	0.8938

Table 6.10: Effectiveness of Balance Augmentation under FlipAttack

Dataset	Ptb(%)	random-SGCL				BA-SGCL			
		AUC	Macro-F1	Micro-F1	Binary-F1	AUC	Macro-F1	Micro-F1	Binary-F1
BitcoinAlpha	0	0.8523	0.7329	0.9441	0.9712	**0.8948**	**0.7772**	**0.9526**	**0.9751**
	10	0.7423	0.5889	0.8512	0.9198	**0.7828**	**0.6241**	**0.8577**	**0.9205**
	20	0.6182	0.5021	0.7421	0.8327	**0.6745**	**0.5395**	**0.7459**	**0.8369**
BitconOTC	0	0.8829	0.7789	0.9351	0.9612	**0.9108**	**0.8079**	**0.9383**	**0.9662**
	10	0.7179	0.6143	0.8427	0.9024	**0.7442**	**0.6636**	**0.8572**	**0.9113**
	20	0.6988	0.5613	0.7989	0.8769	**0.7227**	**0.6277**	**0.8072**	**0.8885**
Slashdot	0	0.8853	0.7214	0.8679	0.9123	**0.8956**	**0.7549**	**0.8792**	**0.9301**
	10	0.7332	0.5721	**0.8097**	0.8687	**0.7626**	**0.6094**	0.8024	**0.8736**
	20	0.6712	0.5179	0.7527	0.8305	**0.7035**	**0.5412**	**0.7637**	**0.8386**
Epinions	0	0.8647	0.6932	0.8726	0.9228	**0.8731**	**0.7301**	**0.8742**	**0.9279**
	10	0.7267	0.5823	0.8198	**0.8943**	**0.7564**	**0.6018**	**0.8323**	0.8905
	20	0.6665	0.5402	0.7994	0.8279	**0.6943**	**0.5612**	**0.8023**	**0.8432**

Table 6.11: Ablation Study under balance-attack

Dataset	Ptb(%)	SGCN				BA-SGCL (SGCN encoder)			
		AUC	Macro-F1	Micro-F1	Binary-F1	AUC	Macro-F1	Micro-F1	Binary-F1
BitcoinAlpha	0	0.7992	0.6656	0.8525	0.9041	**0.8447**	**0.7226**	**0.9298**	**0.9615**
	10	0.6913	0.5125	0.6602	0.7684	**0.7723**	**0.6113**	**0.8832**	**0.9225**
	20	0.6535	0.4704	0.6022	0.7346	**0.6954**	**0.5383**	**0.7921**	**0.8718**
BitconOTC	0	0.8253	0.7501	0.8791	0.9296	**0.8957**	**0.7821**	**0.9244**	**0.9557**
	10	0.7504	0.6142	0.7474	0.8409	**0.8229**	**0.6743**	**0.8578**	**0.9143**
	20	0.6985	0.5621	0.6833	0.7925	**0.7449**	**0.5823**	**0.7742**	**0.8598**
Slashdot	0	0.8153	0.6834	0.8127	0.8788	**0.8851**	**0.7044**	**0.8724**	**0.9211**
	10	0.6892	0.5719	0.6518	0.7528	**0.7559**	**0.5962**	**0.7973**	**0.8721**
	20	0.6348	0.5207	0.5939	0.7029	**0.6776**	**0.5543**	**0.7685**	**0.8427**
Epinions	0	0.7763	0.6957	0.8181	0.8863	**0.8623**	**0.7083**	**0.8661**	**0.9292**
	10	0.7383	0.6125	0.7038	0.7976	**0.7998**	**0.6737**	**0.8574**	**0.9046**
	20	0.6881	0.5606	0.6453	0.7497	**0.7478**	**0.6122**	**0.7992**	**0.8575**

Table 6.12: Ablation Study under FlipAttack

Dataset	Ptb(%)	SGCN				BA-SGCL (SGCN encoder)			
		AUC	Macro-F1	Micro-F1	Binary-F1	AUC	Macro-F1	Micro-F1	Binary-F1
BitcoinAlpha	0	0.7992	0.6656	0.8525	0.9041	**0.8447**	**0.7226**	**0.9298**	**0.9615**
	10	0.6746	0.5543	0.6747	0.7982	**0.7445**	**0.5689**	**0.8332**	**0.9098**
	20	0.5601	0.4322	0.5798	0.7074	**0.6052**	**0.4925**	**0.7121**	**0.8232**
BitconOTC	0	0.8253	0.7501	0.8791	0.9296	**0.8957**	**0.7821**	**0.9244**	**0.9557**
	10	0.6844	0.6086	0.7238	0.8253	**0.6929**	**0.6077**	**0.8112**	**0.8923**
	20	0.6315	0.5512	0.6425	0.7597	**0.6639**	**0.5723**	**0.7661**	**0.8524**
Slashdot	0	0.8153	0.6834	0.8127	0.8788	**0.8851**	**0.7044**	**0.8724**	**0.9211**
	10	0.6412	0.5212	0.6867	0.7475	**0.7278**	**0.5752**	**0.8003**	**0.8661**
	20	0.6041	0.4776	0.6268	0.7092	**0.6723**	**0.5114**	**0.7476**	**0.8212**
Epinions	0	0.7763	0.6957	0.8181	0.8863	**0.8623**	**0.7083**	**0.8661**	**0.9292**
	10	0.6898	0.5215	0.7098	0.7635	**0.7259**	**0.5778**	**0.8132**	**0.8776**
	20	0.6149	0.4884	0.6835	0.7289	**0.6662**	**0.5387**	**0.7889**	**0.8236**

Chapter 7

Conclusion

In this paper, we present the balance-attack, a novel black-box attack designed to reduce the balance degree in signed graphs. To tackle this NP-hard problem, we propose an efficient heuristic algorithm. We conduct extensive experiments using popular SGNN models to validate the effectiveness and generality of the attack. Through our research, we aim to enhance the understanding of the limitations and resilience of robust models when confronted with attacks on SGNNs.

Furthermore, we shed light on the vulnerability of existing SGNNs to adversarial attacks, which significantly impact the balance of signed graphs. To address this issue, we introduce the *balance learning* method for restoring attacked graphs. However, during the course of our investigation, we encounter the challenge of *Irreversibility of Balance-related Information*. In response, we propose BA-SGCL, a novel robust SGNN model that combines balance augmentation and GCL techniques. This approach aims to defend against adversarial attacks and indirectly tackle the aforementioned challenge. The theoretical foundation of our approach is supported by mutual information theory. Through empirical evaluations on various signed graph benchmarks under attacks, we demonstrate the effectiveness of our model in defense. This work represents a pioneering effort in the field of robust learning to defend against ad-

versarial attacks in signed graph representation learning, holding promising potential for future advancements.

Looking ahead, there are several promising directions for future research in the field of signed graph security. While some studies have explored dynamic signed graph neural networks, the security aspects of these temporal structures remain largely unexplored. Future work could investigate how adversarial attacks evolve and propagate in dynamic signed networks, and develop adaptive defense mechanisms that account for temporal dependencies. The extension to heterogeneous signed graphs presents another crucial direction, where the interplay between different node types and their relationships could introduce new security challenges. Beyond the development of more sophisticated targeted attacks, future research could delve into the interpretability of SGNN models to better understand their decision-making processes and vulnerabilities. Additionally, privacy preservation in signed graphs presents another critical challenge, particularly in scenarios where relationship polarities may contain sensitive information. The examination of fairness in signed graph learning also warrants attention, ensuring that models maintain equitable performance across different subgroups and relationship types. These directions, combined with the ongoing advancement in robust learning techniques, could significantly enhance our understanding and capability to secure signed graph-based systems.

References

[1] Xing Ai, Jialong Zhou, Yulin Zhu, Gaolei Li, Tomasz P Michalak, Xiapu Luo, and Kai Zhou. Graph anomaly detection at group level: A topology pattern enhanced unsupervised approach. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1213–1227. IEEE, 2024.

[2] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[3] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V Chawla. Graph barlow twins: A self-supervised representation learning framework for graphs. *Knowledge-Based Systems*, 256:109631, 2022.

[4] Aleksandar Bojchevski and Stephan Günnemann. Adversarial attacks on node embeddings via graph poisoning. In *International Conference on Machine Learning*, pages 695–704. PMLR, 2019.

[5] Yu Bu, Yulin Zhu, Longling Geng, and Kai Zhou. Uncovering strong ties: A study of indirect sybil attack on signed social network. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4535–4539. IEEE, 2024.

[6] Binh-Minh Bui-Xuan and Nick S Jones. How modular structure can simplify tasks on networks: parameterizing graph optimization by fast local commu-

nity detection. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470(2170):20140224, 2014.

[7] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

[8] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900, 2015.

[9] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9:e15, 2020.

[10] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020.

[11] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. " bridge" enhanced signed directed network embedding. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 773–782, 2018.

[12] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

[13] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018.

[14] Zhuo Diao and Zhongzheng Tang. Approximation algorithms for balancing signed graphs. In *International Conference on Algorithmic Applications in Management*, pages 399–410. Springer, 2020.

[15] Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in neural information processing systems*, 34:21480–21492, 2021.

[16] Bahareh Fatemi, Soheila Molaei, Shirui Pan, and Samira Abbasgholizadeh Rahimi. Gcnfusion: An efficient graph convolutional network based model for information diffusion. *Expert Systems with Applications*, 202:117053, 2022.

[17] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[18] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

[19] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Graph neural networks for recommender system. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1623–1625, 2022.

[20] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

[21] Aritra Ghosh and Andrew Lan. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2703–2708, 2021.

[22] Michał T Godziszewski, Marcin Waniek, Yulin Zhu, Kai Zhou, Talal Rahwan, and Tomasz P Michalak. Adversarial analysis of similarity-based sign prediction. *Artificial Intelligence*, 335:104173, 2024.

[23] Michał Tomasz Godziszewski, Tomasz P Michalak, Marcin Waniek, Talal Rahwan, Kai Zhou, and Yulin Zhu. Attacking similarity-based sign prediction. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1072–1077. IEEE, 2021.

[24] Lukas Gosch, Simon Geisler, Daniel Sturm, Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Adversarial training for graph neural networks: Pitfalls, solutions, and new directions. *Advances in Neural Information Processing Systems*, 36, 2024.

[25] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412, 2004.

[26] Krystal Guo and Bojan Mohar. Hermitian adjacency matrix of digraphs and mixed graphs. *Journal of Graph Theory*, 85(1):217–248, 2017.

[27] Xiaojun Guo, Yifei Wang, Zeming Wei, and Yisen Wang. Architecture matters: Uncovering implicit mechanisms in graph contrastive learning. *Advances in Neural Information Processing Systems*, 36:28585–28610, 2023.

[28] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[29] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[30] Frank Harary and Jerald A Kabell. A simple algorithm to detect balance in signed graphs. *Mathematical Social Sciences*, 1(1):131–136, 1980.

[31] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pages 4116–4126. PMLR, 2020.

[32] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 804–813, 2017.

[33] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Sdgnn: Learning node representation for signed directed networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 196–203, 2021.

[34] Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. Metropolis algorithms for representative subgraph sampling. In *2008 eighth ieee international conference on data mining*, pages 283–292. IEEE, 2008.

[35] Yuxin Jiang, Linhan Zhang, and Wei Wang. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3021–3035, 2022.

[36] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Improving contrastive learning on imbalanced data via open-world sampling. *Advances in neural information processing systems*, 34:5997–6009, 2021.

[37] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[38] Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-guided contrastive learning for bert sentence representations. *arXiv preprint arXiv:2106.07345*, 2021.

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[40] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[41] Taewook Ko, Yoonhyuk Choi, and Chong-Kwon Kim. A spectral graph convolution for signed directed graphs via magnetic laplacian. *Neural Networks*, 164:562–574, 2023.

[42] Taewook Ko, Yoonhyuk Choi, and Chong-Kwon Kim. Universal graph contrastive learning with a novel laplacian perturbation. In *Uncertainty in Artificial Intelligence*, pages 1098–1108. PMLR, 2023.

[43] Ishaan Kumar, Yaochen Hu, and Yingxue Zhang. Eflec: Efficient feature-leakage correction in gnn based recommendation systems. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 1885–1889, 2022.

[44] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 221–230. IEEE, 2016.

[45] Yuni Lai, Jialong Zhou, Xiaoge Zhang, and Kai Zhou. Towards certified robustness of graph neural networks in adversarial aiot environments. *IEEE Internet of Things Journal*, 2023.

[46] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, 2010.

[47] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[48] Jintang Li, Jie Liao, Ruofan Wu, Liang Chen, Zibin Zheng, Jiawang Dan, Changhua Meng, and Weiqiang Wang. Guard: Graph universal adversarial defense. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1198–1207, 2023.

[49] Michelle M Li, Kexin Huang, and Marinka Zitnik. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering*, 6(12):1353–1369, 2022.

[50] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[51] Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. Learning signed network embedding via graph attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4772–4779, 2020.

[52] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[53] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[54] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM computing surveys (CSUR)*, 51(3):1–34, 2018.

[55] Tomasz Lizurej, Tomasz Michalak, and Stefan Dziembowski. On manipulating weight predictions in signed weighted networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5222–5229, 2023.

[56] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection

with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12012–12038, 2021.

[57] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146, 2010.

[58] Vasimuddin Md, Sanchit Misra, Guixiang Ma, Ramanarayan Mohanty, Evangelos Georganas, Alexander Heinecke, Dhiraj Kalamkar, Nesreen K Ahmed, and Sasikanth Avancha. Distgnn: Scalable distributed training for large-scale graph neural networks. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14, 2021.

[59] Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, 55(12):1–37, 2023.

[60] Shengjie Min, Zhan Gao, Jing Peng, Liang Wang, Ke Qin, and Bo Fang. Stgsn—a spatial–temporal graph neural network framework for time-evolving social networks. *Knowledge-Based Systems*, 214:106746, 2021.

[61] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[62] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1150–1160, 2020.

[63] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing*, 68:6303–6318, 2020.

[64] Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (CSUR)*, 54(10s):1–25, 2022.

[65] Lin Shu, Erxin Du, Yaomin Chang, Chuan Chen, Zibin Zheng, Xingxing Xing, and Shaofeng Shen. Sgcl: Contrastive representation learning for signed graphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1671–1680, 2021.

[66] Prem Kumar Singh. Data with non-euclidean geometry and its characterization. *Journal of Artificial Intelligence and Technology*, 2(1):3–8, 2022.

[67] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019.

[68] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, S Yu Philip, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[69] Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3):1–37, 2016.

[70] Tian-li Tao, Liang-hu Guo, Qiang He, Han Zhang, and Lin Xu. Seizure detection by brain-connectivity analysis using dynamic graph isomorphism network. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2302–2305. IEEE, 2022.

[71] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[72] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

[73] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[74] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 327–335. SIAM, 2017.

[75] Cheng Wu, Chaokun Wang, Jingcao Xu, Ziyang Liu, Kai Zheng, Xiaowei Wang, Yang Song, and Kun Gai. Graph contrastive learning with generative adversarial network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2721–2730, 2023.

[76] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

[77] Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. Graph neural networks in node classification: survey and evaluation. *Machine Vision and Applications*, 33(1):4, 2022.

[78] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:30414–30425, 2021.

[79] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International journal of automation and computing*, 17:151–178, 2020.

[80] Jiarong Xu, Junru Chen, Siqi You, Zhiqing Xiao, Yang Yang, and Jiangang Lu. Robustness of deep learning models on graphs: A survey. *AI Open*, 2:69–78, 2021.

[81] Kaige Yang and Laura Toni. Graph-based recommendation system. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 798–802. IEEE, 2018.

[82] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, and Hui Xiong. Coupled layer-wise graph convolution for transportation demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4617–4625, 2021.

[83] Hai-Cheng Yi, Zhu-Hong You, De-Shuang Huang, and Chee Keong Kwoh. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1):bbab340, 2022.

[84] Xiaoyan Yin, Wanyu Lin, Kexin Sun, Chun Wei, and Yanjiao Chen. A 2 s 2-gnn: Rigging gnn-based social status by adversarial attacks in signed social networks. *IEEE Transactions on Information Forensics and Security*, 18:206–220, 2022.

[85] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[86] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[87] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

[88] Sixiao Zhang, Hongxu Chen, Xiangguo Sun, Yicong Li, and Guandong Xu. Unsupervised graph poisoning attack via contrastive loss back-propagation. In *Proceedings of the ACM Web Conference 2022*, pages 1322–1330, 2022.

[89] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

[90] Yanfu Zhang, Shangqian Gao, Jian Pei, and Heng Huang. Improving social network embedding via new second-order continuous graph neural networks. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2515–2523, 2022.

[91] Zeyu Zhang, Jiamou Liu, Xianda Zheng, Yifei Wang, Pengqian Han, Yupan Wang, Kaiqi Zhao, and Zijian Zhang. Rsgnn: A model-agnostic approach for enhancing the robustness of signed graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 60–70, 2023.

[92] Zeyu Zhang, Peiyao Zhao, Xin Li, Jiamou Liu, Xinrui Zhang, Junjie Huang, and Xiaofeng Zhu. Signed graph representation learning: A survey. *arXiv preprint arXiv:2402.15980*, 2024.

[93] Han Zhao, Xu Yang, Cheng Deng, and Dacheng Tao. Unsupervised structure-adaptive graph contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[94] Xiaolong Zheng, Daniel Zeng, and Fei-Yue Wang. Social balance in signed networks. *Information Systems Frontiers*, 17:1077–1095, 2015.

[95] Fan Zhou, Qing Yang, Ting Zhong, Dajiang Chen, and Ning Zhang. Variational graph neural networks for road traffic prediction in intelligent transportation systems. *IEEE Transactions on Industrial Informatics*, 17(4):2802–2812, 2020.

[96] Jialong Zhou, Xing Ai, Yuni Lai, and Kai Zhou. Adversarially robust signed graph contrastive learning from balance augmentation. *arXiv preprint arXiv:2401.10590*, 2024.

[97] Jialong Zhou, Yuni Lai, Jian Ren, and Kai Zhou. Black-box attacks against signed graph analysis via balance poisoning. In *2024 International Conference on Computing, Networking and Communications (ICNC)*, pages 530–535. IEEE, 2024.

[98] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[99] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment*, 2(1):718–729, 2009.

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

[101] Yulin Zhu, Tomasz Michalak, Xiapu Luo, Xiaoge Zhang, and Kai Zhou. Towards secrecy-aware attacks against trust prediction in signed social networks. *IEEE Transactions on Information Forensics and Security*, 2024.

[102] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11127–11135, 2019.

[103] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM*

SIGKDD international conference on knowledge discovery & data mining, pages 2847–2856, 2018.

[104] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations (ICLR)*, 2019.