Black-Box Attacks against Signed Graph Analysis via Balance Poisoning

Jialong Zhou†, Yuni Lai†, Jian Ren*, Kai Zhou†
† Department of Computing, The Hong Kong Polytechnic University, HKSAR
* Dept of ECE, Michigan State University, Michigan, USA
{jialong.zhou, yunie.lai}@connect.polyu.hk, renjian@msu.edu, kaizhou@polyu.edu.hk

Abstract*—Signed graphs are well-suited for modeling social networks as they capture both positive and negative relationships. Signed graph neural networks (SGNNs) are commonly employed to predict link signs (i.e., positive and negative) in such graphs due to their ability to handle the unique structure of signed graphs. However, real-world signed graphs are vulnerable to malicious attacks by manipulating edge relationships, and existing adversarial graph attack methods do not consider the specific structure of signed graphs. SGNNs often incorporate balance theory to effectively model the positive and negative links. Surprisingly, we find that the balance theory that they rely on can ironically be exploited as a black-box attack. In this paper, we propose a novel black-box attack called *balance*-attack that aims to decrease the balance degree of the signed graphs. We present an efficient heuristic algorithm to solve this NP-hard optimization problem. We conduct extensive experiments on five popular SGNN models and four real-world datasets to demonstrate the effectiveness and wide applicability of our proposed attack method. By addressing these challenges, our research contributes to a better understanding of the limitations and resilience of robust models when facing attacks on SGNNs. This work contributes to enhancing the security and reliability of signed graph analysis in social network modeling. Our PyTorch implementation of the attack is publicly available on GitHub: https://github.com/JialongZhou666/Balance-Attack.git.

***Index Terms*—signed graph, signed graph neural networks, adversarial attacks, balance theory**

I. Introduction

In social interactions, relationships can encompass both positive aspects, such as trust, and negative aspects, such as hate. Signed graphs provide a suitable network structure for capturing these diverse relationships. By incorporating positive and negative signs, signed graphs effectively represent both friendly and hostile connections, making them well-suited for modeling various social networks. An example of such a scenario is the Bitcoin-Alpha platform, where users can rate others using positive or negative scores. This natural setting can be effectively modeled using a signed graph. Machine learning techniques have played a significant role in analyzing signed graph data, addressing tasks such as link sign prediction [1], [2] and node ranking [3], [4].

This paper primarily focuses on the task of link sign prediction, which involves inferring the signs of edges in the unexplored portion of a signed graph based on a known subgraph with its structure and edge signs. Numerous existing approaches for link sign prediction rely on signed graph neural networks (SGNNs) [5]–[9]. SGNNs are constructed based on graph neural networks (GNNs) [10] and are specifically designed to accommodate the unique graph structure of signed graphs. Since the presence of negative edges invalidates the standard message-passing mechanism, the development of SGNN models becomes necessary to effectively handle both positive and negative edges.

To effectively address the challenges associated with negative edges in the design of SGNNs, a common approach is to incorporate balance theory from social psychology. This theory provides valuable insights into managing and integrating positive and negative links, thereby establishing a cohesive framework for learning node representations. Balance theory suggests the existence of an expected "balanced structure" in which signed triangles, composed of three interconnected nodes, should have an even number of negative edges [11]. Empirical studies have confirmed that most triangles in real-world signed social graph datasets adhere to these conditions [1]. Existing SGNN models, such as SGCN [5] and SNEA [8], have leveraged balance theory in the design of their aggregation strategies.

However, real-world signed graphs are vulnerable to malicious attacks. For instance, in bitcoin trading platforms, users may engage in manipulative behavior by providing false ratings, while in e-commerce sites, attackers can disrupt the integrity of the award system by assigning low scores. These attacks typically involve altering a small portion of the edge relationships within the signed network. Such manipulations can have a significant impact on the results of link sign prediction using SGNNs, potentially leading to the deterioration of social relationships.

To understand the vulnerability of SGNNs, it is necessary to develop attack methods for signed graphs. Existing adversarial graph attack methods like Nettack [12] and Metattack [13] are not suitable, as they primarily require node labels and features for node classification tasks. Therefore, a new attack method tailored for signed graphs is required. Currently, there is a noticeable lack of black-box attacks for signed graphs. The only existing method that somewhat resembles a black-box attack is the random attack, where the signs of some edges are randomly altered. However, this method proves to be ineffective.

Given that most SGNN models rely on balance theory to aggregate information, either directly or implicitly, we propose a novel black-box attack for SGNNs by reducing the balance degree, which we termed as ***balance*-attack**. It has been proved that a SGNN is incapable of learning accurate node representations from unbalanced triangles [14]. By developing an algorithm to manipulate the degree of balance, our proposed ***balance*-attack** shows to be effective. The major contributions of our research are as follows:

- We introduce a novel black-box attack for signed graph neural networks by corrupting the balance degree.
- 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 [15].
- We conduct extensive experiments on four datasets using five popular SGNN models to demonstrate the effectiveness and generality of our proposed attack.

By addressing these issues, our aim is to advance the understanding of the limitations and resilience of robust models when faced with attacks on signed graph neural networks.

II. 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. [12] 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. [13], 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 highlight that aforementioned studies primarily focus on unsigned graphs. When it comes to signed graphs, there is limited research in the context of adversarial attacks. Godziszewski et al. [16] introduce the concept of attacking sign prediction, where an attacker aims to conceal the signs of a specific set of target links from a network analyst by eliminating the signs of non-target links. However, this method is not specifically designed for SGNN models, and it does not function as a black-box attack. To the best of our knowledge, there have been no reported instances of adversarial attacks specifically tailored to signed graphs in a black-box manner thus far.

III. Preliminaries

A. Balance Theory

In balance theory, balanced triads in a graph are defined as triads that contain an even number of negative edges. For instance, in Fig. 1, where positive and negative edges are represented by red solid and green dashed lines, respectively, the first two triads, where all three users are friends or only one pair of them are friends, are considered balanced. Based on previous works, this theory posits that individuals within a social network have a propensity to form structures that adhere to balance [1]. To quantify the degree of balance in a signed graph, a measurement called balance degree ($D_3(G)$) was introduced [17]. It calculates the fraction of balanced triads in the graph using the following formula:

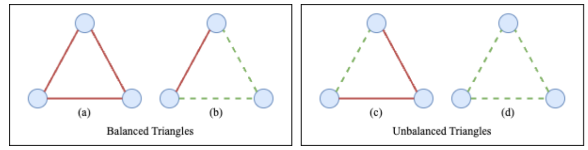

Fig. 1. Four types of triangles.

$$D_3(G) = \frac{Tr(A^3) + Tr(|A|^3)}{2Tr(|A|^3)}, \tag{1}$$

where $Tr(\cdot)$ represents the trace of a matrix, A is the signed adjacency matrix of the signed graph G. The elements in matrix A can take values of $\{-1, 1, 0\}$ to represent negative edges, positive edges, or the absence of an edge in the signed graphs.

B. Signed Graph Analysis

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.

SGCN [5], 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:

$$h_i^{(l)} = [h_i^{pos(l)}, h_i^{neg(l)}], \tag{2}$$

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:

$$\begin{aligned}
t_i^{pos(l)} &= AGG^{(l)}(h_j^{pos(l-1)} : v_j \in \mathcal{N}_i^+, h_j^{neg(l-1)} : v_j \in \mathcal{N}_i^-) \\
h_i^{pos(l)} &= COM^{(l)}(h_i^{pos(l-1)}, t_i^{pos(l)}) \\
t_i^{neg(l)} &= AGG^{(l)}(h_j^{neg(l-1)} : v_j \in \mathcal{N}_i^+, h_j^{pos(l-1)} : v_j \in \mathcal{N}_i^-) \\
h_i^{neg(l)} &= COM^{(l)}(h_i^{neg(l-1)}, t_i^{neg(l)}),
\end{aligned} \tag{3}$$

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 [9] and UGCL [18] utilize graph contrastive learning for signed graphs.. SGDNN [7] combines balance theory and status theory along with the introduction of four weight matrices. RSGNN [14] incorporates structure-based regularizers to enhance performance.

IV. Problem Statements

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 accordingly.

A. 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.

B. Threat Model

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 [13], 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.

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.

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 $\Phi(G; \Delta)$.

C. 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:

$$\min_{\hat{G}\in\Phi(G;\Delta)} \mathcal{L}_{atk} = \sum_{e\in\mathcal{D}_{test}} \mathbb{I}\{f_{\theta^*}(\hat{G})_e = sign(e)\}, \quad (4)$$
$$s.t.\ \theta^* = arg\min_{\theta} \mathcal{L}_{train}(f_\theta(\hat{G})),$$

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.

V. Proposed Black-Box Attack

A. Formulation of black-box attack

Since the model structure and labels of the testing data are always unavailable, directly optimizing (4) 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 [14], 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 (4) with the optimization problem as follows:

$$\min_{\hat{G}\in\Phi(G;\Delta)} D_3(\hat{G}). \quad (5)$$

B. 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 [15], 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

TABLE I
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%

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:

$$\begin{aligned} i^*, j^* &= \underset{\substack{\{i,j|a_{ij}\neq 0 \wedge sign(a_{ij})= \\ sign(\nabla_{ij}D_3(\hat{G}))\}}}{arg\max} |\nabla_{ij}D_3(\hat{G})|, \\ a_{ij} &= -a_{i^*j^*}, \end{aligned} \tag{6}$$

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. V-B.

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 ← A.
while Number of changed edges ≤ Δ do
    Calculate D_3(S).
    Calculate gradient matrix ∇(D_3(S)).
    Filter candidate edges C_e = {i, j | s_ij ≠ 0 ∧ sign(s_ij) = sign(∇_ij D_3(S))}.
    if i*, j* = arg max_{i,j∈C_e} |∇_ij(D_3(S))| then
        Update s_{i*j*} = −s_{i*j*}.
        Number of changed edges += 1.
    end if
end while
Return S.
```

VI. EXPERIMENTS

In this section, we perform experiments on *4* 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 *5* 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?

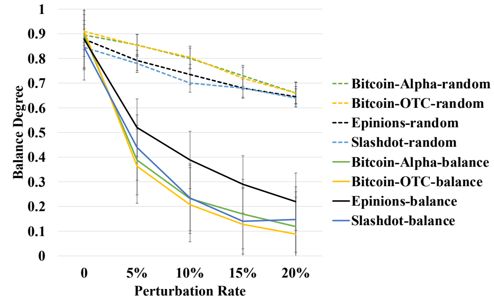

Fig. 2. Balance degree of 4 datasets under 2 attacks.

- **Q3**: How applicable is ***balance*-attack** on various SGNN models?

A. Baseline

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 [13], [19] 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.

B. Setup

We conduct experiments on four public real-world datasets: Bitcoin-Alpha, Bitcoin-OTC [20], Epinions [21], and Slashdot [22]. 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 Table I.

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

- **SGCN** [5] 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** [9] 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** [7] combines both balance theory and status theory, and introduces four weight matrices to aggregate neighbor features based on edge types.
- **RSGNN** [14] 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** [18] 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.

TABLE II
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
	5%	random	0.8299	0.9018	0.6317
		balance	**0.7563**	**0.8534**	**0.5648**
	10%	ranodm	0.7726	0.8642	0.5831
		balance	**0.6802**	**0.7984**	**0.5123**
	15%	random	0.7360	0.8393	0.5499
		balance	**0.6605**	**0.7862**	**0.4814**
	20%	random	0.6839	0.8010	0.5165
		balance	**0.6308**	**0.7631**	**0.4635**
Bitcoin-OTC	0	-	0.8919	0.9382	0.7553
	5%	random	0.8654	0.9216	0.7227
		balance	**0.7970**	**0.8761**	**0.6574**
	10%	random	0.8242	0.8950	0.6782
		balance	**0.7134**	**0.8158**	**0.5849**
	15%	random	0.8180	0.8911	0.6694
		balance	**0.6787**	**0.7909**	**0.5486**
	20%	random	0.7828	0.8673	0.6341
		balance	**0.6424**	**0.7625**	**0.5194**
Slashdot	0	-	0.7823	0.8574	0.6988
	5%	random	0.7384	0.8321	0.6629
		balance	**0.7344**	**0.8225**	**0.6484**
	10%	random	0.7092	0.7982	0.639
		balance	**0.6719**	**0.7761**	**0.5813**
	15%	random	0.6707	0.7637	0.6104
		balance	**0.6378**	**0.7466**	**0.5557**
	20%	random	0.6637	0.7576	0.6044
		balance	**0.6009**	**0.7165**	**0.5215**
Epinions	0	-	0.8280	0.8932	0.7261
	5%	random	0.8155	0.8841	0.7160
		balance	**0.7736**	**0.8542**	**0.6739**
	10%	random	0.7711	0.8516	0.6754
		balance	**0.7342**	**0.8234**	**0.6432**
	15%	random	0.7376	0.8257	0.6475
		balance	**0.7068**	**0.8016**	**0.6201**
	20%	random	0.7409	0.8285	0.6492
		balance	**0.6832**	**0.7836**	**0.5962**

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

TABLE III
LINK SIGN PREDICTION PERFORMANCE OF SGNNS 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
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**

C. Balance Degree of Signed Graphs after Attack (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 compared to the balance degree under the random attack. We present the comparison results of the ***balance*-attack** and random attack in Fig. 2. 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.

D. 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 Table II, 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 Table III.

E. 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.

VII. Conclusion

In this paper, we introduce ***balance*-attack**, a novel black-box attack for signed graphs, which reduces the balance degree. We propose an efficient heuristic algorithm to solve this NP-hard problem. Extensive experiments are conducted using popular SGNN models to validate the attack's effectiveness and generality. Our research aims to enhance the understanding of the limitations and resilience of robust models when faced with attacks on SGNNs.

Acknowledgement

This research was partly supported by the National Science Foundation of China (No. 62106210) and the Hong Kong Research Grant Council (No. PolyU25210821).

References

[1] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th international conference on World wide web*, pp. 641–650, 2010.

[2] D. Song and D. A. Meyer, "Link sign prediction and ranking in signed directed social networks," *Social network analysis and mining*, vol. 5, pp. 1–14, 2015.

[3] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–37, 2016.

[4] Y. Chen, T. Qian, H. Liu, and K. Sun, """ bridge" enhanced signed directed network embedding," in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 773–782, 2018.

[5] T. Derr, Y. Ma, and J. Tang, "Signed graph convolutional networks," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 929–934, IEEE, 2018.

[6] J. Huang, H. Shen, Q. Cao, S. Tao, and X. Cheng, "Signed bipartite graph neural networks," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 740–749, 2021.

[7] J. Huang, H. Shen, L. Hou, and X. Cheng, "Sdgnn: Learning node representation for signed directed networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 196–203, 2021.

[8] Y. Li, Y. Tian, J. Zhang, and Y. Chang, "Learning signed network embedding via graph attention," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 4772–4779, 2020.

[9] L. Shu, E. Du, Y. Chang, C. Chen, Z. Zheng, X. Xing, and S. Shen, "Sgcl: Contrastive representation learning for signed graphs," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1671–1680, 2021.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[11] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory.," *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[12] A. Bojchevski and S. Günnemann, "Adversarial attacks on node embeddings via graph poisoning," in *International Conference on Machine Learning*, pp. 695–704, PMLR, 2019.

[13] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," in *International Conference on Learning Representations (ICLR)*, 2019.

[14] Z. Zhang, J. Liu, X. Zheng, Y. Wang, P. Han, Y. Wang, K. Zhao, and Z. Zhang, "Rsgnn: A model-agnostic approach for enhancing the robustness of signed graph neural networks," in *Proceedings of the ACM Web Conference 2023*, pp. 60–70, 2023.

[15] Z. Diao and Z. Tang, "Approximation algorithms for balancing signed graphs," in *International Conference on Algorithmic Applications in Management*, pp. 399–410, Springer, 2020.

[16] M. T. Godziszewski, T. P. Michalak, M. Waniek, T. Rahwan, K. Zhou, and Y. Zhu, "Attacking similarity-based sign prediction," in *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1072–1077, IEEE, 2021.

[17] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 891–900, 2015.

[18] T. Ko, Y. Choi, and C.-K. Kim, "Universal graph contrastive learning with a novel laplacian perturbation," in *Uncertainty in Artificial Intelligence*, pp. 1098–1108, PMLR, 2023.

[19] S. Zhang, H. Chen, X. Sun, Y. Li, and G. Xu, "Unsupervised graph poisoning attack via contrastive loss back-propagation," in *Proceedings of the ACM Web Conference 2022*, pp. 1322–1330, 2022.

[20] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos, "Edge weight prediction in weighted signed networks," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 221–230, IEEE, 2016.

[21] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th international conference on World Wide Web*, pp. 403–412, 2004.

[22] K. Guo and B. Mohar, "Hermitian adjacency matrix of digraphs and mixed graphs," *Journal of Graph Theory*, vol. 85, no. 1, pp. 217–248, 2017.