Hypergraph Attacks via Injecting Homogeneous Nodes into Elite Hyperedges

Meixia He[1], Peican Zhu[1*], Keke Tang[2,3*], Yangming Guo[4*]

[1]School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University
[2]Cyberspace Institute of Advanced Technology, Guangzhou University
[3]Huangpu Research School, Guangzhou University
[4]School of Cybersecurity, Northwestern Polytechnical University
meixia_he@mail.nwpu.edu.cn, ericcan@nwpu.edu.cn, tangbohutbh@gmail.com, yangming_g@nwpu.edu.cn

Abstract

Recent studies have shown that Hypergraph Neural Networks (HGNNs) are vulnerable to adversarial attacks. Existing approaches focus on hypergraph modification attacks guided by gradients, overlooking node spanning in the hypergraph and the group identity of hyperedges, thereby resulting in limited attack performance and detectable attacks. In this manuscript, we present a novel framework, i.e., Hypergraph **Attack**s via **I**njecting Homogeneous Nodes into **E**lite Hyperedges (IE-Attack), to tackle these challenges. Initially, utilizing the node spanning in the hypergraph, we propose the elite hyperedges sampler to identify hyperedges to be injected. Subsequently, a node generator utilizing Kernel Density Estimation (KDE) is proposed to generate the homogeneous node with the group identity of hyperedges. Finally, by injecting the homogeneous node into elite hyperedges, IE-Attack improves the attack performance and enhances the imperceptibility of attacks. Extensive experiments are conducted on five authentic datasets to validate the effectiveness of IE-Attack and the corresponding superiority to state-of-the-art methods.

Introduction

Graph Neural Networks (GNNs) (Kipf and Welling 2017; Veličković et al. 2017; Cheng et al. 2024a; Liu et al. 2023; Dong et al. 2023; Wang et al. 2023; Cheng et al. 2024b) capture intricate relationships and patterns within graph data, facilitating tasks like node classification, recommendation systems and source detection, etc. Nevertheless, with the increasing complexity and diversity of real-world networks, attention has shifted towards higher-order networks like hypergraphs and simplicial complexes (Benson, Gleich, and Leskovec 2016; Battiston et al. 2021; Wang et al. 2024). Subsequently, Hypergraph Neural Networks (HGNNs) (Feng et al. 2019; Bai, Zhang, and Torr 2021) are proposed to extract higher-order features from hypergraphs, significantly improving the efficiency of downstream graph-related tasks. Despite its success, GNNs and HGNNs have been shown to be vulnerable to adversarial attacks (Zügner, Akbarnejad, and Günnemann 2018; Bojchevski and Günnemann 2019), which has attracted increasing research interest.
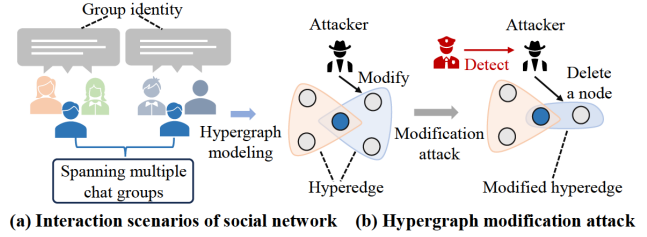

Figure 1: An illustration of the attack scenarios of hypergraph. (a) The same user spans multiple chat groups and chat groups exhibit group identity in social network. (b) Current hypergraph attack methods only involve hypergraph modification attacks and are easily detectable.

Adversarial attacks on GNNs are divided into modification attack and injection attack according to attack techniques (Wei et al. 2020; Sun et al. 2022). In modification attack (Chang et al. 2020; Jin et al. 2023), attackers aim to affect the performance of GNNs by modifying the attributes or connections of nodes and edges. Conversely, due to the high authority required by modification attack, the injection attack has received more attention. For instance, in social networks, injection attacks involve create new fake accounts without the need to compromise existing accounts, thus requiring relatively low permissions. Injection attacks (Sun et al. 2019; Zou et al. 2021; Zhu et al. 2024c) inject new nodes or edges into the graph, introducing malicious information to degrade the performance of GNNs. Nevertheless, researches on adversarial attacks against HGNNs are limited to the hypergraph modification attack, including HyperAttack (Hu et al. 2023) and MGHGA (Chen et al. 2023). These methods do not further consider the phenomenon of node spanning in the hypergraph and the group identity of hyperedges, resulting in poor attack performance and easily detectable attacks, as illustrated in Figure 1.

By modeling higher-order relationships in the network, hypergraphs exhibit richer graph structural features, leading to more limitations on hypergraph attacks. First, modifying hypergraphs requires high authority, limiting the applicability of hypergraph modification attacks. Second, the existence of node spanning in hypergraphs (Battiston et al. 2021) implies that when the same node appears in multiple hy-

peredges, hyperedges with substantial influence can be discerned by the frequency of node occurrences across various hyperedges. For instance, in social networks, when a user presents in multiple chat groups, it suggests that the user can extensively disseminate information through these groups within the network. However, current methods select modified hyperedges by calculating gradients, which cannot maximize the destruction of the feature aggregation of HGNNs, resulting in poor attack performance. Moreover, hyperedges are viewed as groups and exhibit group identity according to social psychology research (Spears 2021). Current hypergraph modification attacks do not consider the group identity of hyperedges when adding or removing known nodes in hyperedges, making the attacks easily detectable.

To address these challenges, we propose the Hypergraph **Attack**s via **I**njecting Homogeneous Nodes into **E**lite Hyperedges (IE-Attack). Firstly, due to the high authority required by hypergraph modification, we present the node injection attacks on hypergraphs. Secondly, inspired by the elite group in social psychology (Shayegh et al. 2022; Howard 2000), we propose an elite hyperedges sampler to identify elite hyperedges with significant influence by utilizing the node spanning phenomenon in the hypergraph. Additionally, we develop a node generator based on Kernel Density Estimation (KDE) to generate the homogeneous node, so that the elite hyperedges injected into the homogeneous node still exhibit group identity. By injecting the homogeneous node into elite hyperedges, IE-Attack maximizes malicious information propagation to obtain excellent attack results and enhances the imperceptibility of attacks. We validate the effectiveness of our approach on five publicly available datasets. Extensive experimental results demonstrate that IE-Attack exhibits excellent attack performance on HGNNs, outperforming state-of-the-art node injection attack methods.

Overall, our contributions are summarized as:

- We are the first to analyze the problem of node injection attack in hypergraphs and to propose a methodology to address this challenge.
- We propose a novel attack method against HGNNs for the Hypergraph Attacks via Injecting Homogeneous Nodes into Elite Hyperedges.
- We demonstrate the effectiveness of our proposed method over baseline approaches through extensive experimental validation.

Related Works

Graph Injection Attack   Deep neural networks (Tang et al. 2022a) are known to be vulnerable to adversarial attacks, and this susceptibility has been extensively studied across various domains, including images (Zhu et al. 2024a; Li et al. 2023), point clouds (Tang et al. 2022b), and graphs (Wei et al. 2020). In this paper, we focus on graph injection attacks. NIPA (Sun et al. 2019) introduced a novel node injection approach to poison the graph structure, which was based on reinforcement learning. Likewise, AFGSM (Wang et al. 2020) addressed a more practical attack scenario, allowing adversaries to inject malicious nodes into

the graph without manipulating the existing graph structure. Conversely, both NIPA and AFGSM were developed in a poisoning environment, requiring retraining of the defense models after each attack. TDGIA (Zou et al. 2021) followed the evasion attack setting of KDDCUP 2020, where different attacks were evaluated based on the same set of models and weights. In contrast to previous attacks in a white-box setting, G^2A2C (Ju et al. 2023) and (Zhu et al. 2024b) proposed node injection attacks in a black-box setting. Subsequently, considering that previous studies involved injecting multiple nodes for attacks, G-NIA (Tao et al. 2021) introduced the extreme scenario of single node injection. G-NIA_CANA (Tao et al. 2023) incorporated a generator-discriminator structure to enhance the imperceptibility of attacks based on G-NIA.

Different from node injection attack in ordinary graphs, hyperedges involve higher-order relationships among multiple nodes in hypergraphs, rendering graph injection attack methods unsuitable for hypergraph structures.

Hypergraph Neural Networks   To encapsulate more intricate higher-order interaction details within graph datasets, advanced structures like hypergraphs, represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, have been identified and developed (Jin et al. 2019; Antelmi et al. 2023). Subsequently, HGNNs (Feng et al. 2019) were introduced, where the feature aggregation process involved node-hyperedge-node interactions. This enabled HGNNs to extract more comprehensive feature information compared to GNNs. Approaches such as hypergraph convolution and hypergraph attention (Bai, Zhang, and Torr 2021; Yadati et al. 2019; Zhang, Zou, and Ma 2019) delved into refining the feature aggregation mechanism within hypergraphs, demonstrating notable effectiveness across graph-related tasks like node classification and link prediction. Additionally, a range of related HGNNs provided varied perspectives and find applications in fields such as recommendation systems and biological information networks (Zeng et al. 2023; Wang et al. 2018; Yu et al. 2023).

While the security of HGNNs warrants significant attention, current attacks on HGNNs are solely focused on hypergraph modifications attacks (Hu et al. 2023; Chen et al. 2023), overlooking the node spanning phenomenon in the hypergraph and the group identity within hyperedge.

Preliminary and Problem Statement

Graph Neural Networks   The graph $G = (V, E)$, where V represents the set of nodes and E represents pairwise edges. The node features are denoted by $X \in \mathbb{R}^{|V| \times |F|}$, where F signifies the feature dimension of X. The adjacency matrix is represented as $A \in \mathbb{R}^{|V| \times |V|}$, with A_{ij} being 1 if there exists a connection between node v_i and v_j, and 0 otherwise. The GNNs involve a graph convolution process, where the feature aggregation is expressed as:

$$X^{l+1} = \sigma(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}X^l W^l), \qquad (1)$$

where D denotes the degree matrix, I stands for the identity matrix, and W^l represents the weight matrix of the l-th layer in GNNs. The symbol σ signifies the activation function.
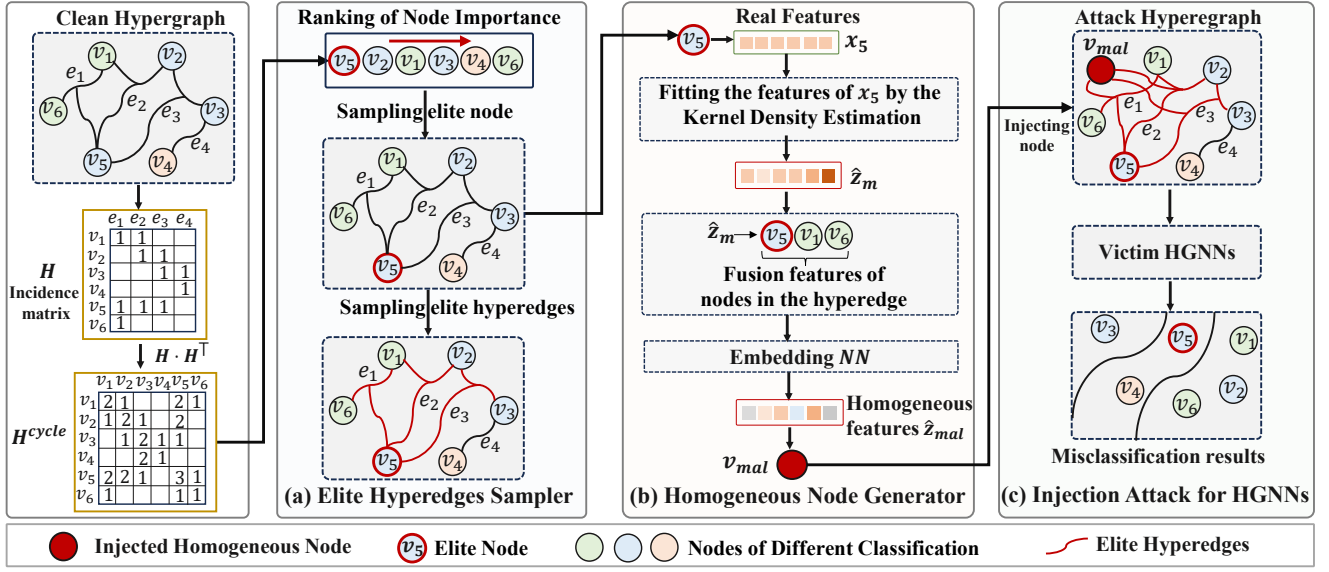
Figure 2: Framework of IE-Attack. (a) Elite Hyperedges Sampler obtains elite hyperedges. (b) Homogeneous Node Generator generates the homogeneous node. (c) IE-Attack attacks HGNNs by injecting the homogeneous node into elite hyperedges.

Hypergraph Neural Networks In contrast to graph $G = (V, E)$, hypergraph defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes and $\mathcal{E}$ denotes hyperedges sets. The feature matrix $\mathcal{X} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{F}|}$ is the node features, with $\mathcal{F}$ denoting the feature dimension of $\mathcal{X}$. Hyperedges consist of sets of nodes comprising two or more nodes, such as the hyperedge $e_1 = \{v_1, v_5, v_6\}$ in Figure 2. The incidence matrix H, which delineates the relationships between nodes and hyperedges, is represented as follows:

$$H = \begin{cases} H_{ij} = 1, & if \quad v_i \in e_j, \\ H_{ij} = 0, & if \quad v_i \notin e_j. \end{cases} \tag{2}$$

Then, the feature aggregation of HGNNs is defined as:

$$\mathcal{X}^{l^h+1} = \sigma(D_{\mathcal{V}}^{-\frac{1}{2}} H \mathcal{W} D_{\mathcal{E}}^{-1} H^\mathsf{T} D_{\mathcal{V}}^{-\frac{1}{2}} \mathcal{X}^{l^h} \Theta^{l^h}), \tag{3}$$

where $D_{\mathcal{V}}$ signifies the degree matrix of nodes, $D_{\mathcal{E}}$ represents the degree matrix of hyperedges, and $\mathcal{W}$ denotes the weight associated with hyperedges. The parameters for training Θ^{l^h} correspond to the l^h-th layer in HGNNs.

Main Challenges of Node Injection Attack in HGNNs In the ordinary graph $G = (V, E)$, the connections between nodes are pairwise. When discussing node injection attacks on GNNs, the focus lies in introducing new nodes into the node set within the graph. The key steps are as follows: (1) the attacker selects target nodes for the injection of new nodes; (2) the attacker generates new nodes and inserts them into the chosen target nodes, thereby establishing new edges.

Conversely, due to the node spanning phenomenon in the hypergraph, the information propagation capacity of nodes across different hyperedges varies. Additionally, hyperedges exhibit group identity, leading to vigilance towards injected nodes and making attacks detectable. Consequently, the attack against HGNNs have two main challenges: (1) proposing a method that leverages the node spanning phenomenon

to identify elite hyperedges for node injection attacks, maximizing the spread of malicious information; (2) Considering the group identity of hyperedges, a node generator should be proposed to enhance the imperceptibility of attacks.

Methodology

Elite Hyperedges Sampler The phenomenon of node spanning in the hypergraph indicates varying abilities of nodes to propagate information within hyperedges, providing us with insights into selecting hyperedges for node injection. Initially, we analyze the feature aggregation process of HGNNs, $\mathcal{X}^{l^h+1} = \sigma(D_{\mathcal{V}}^{-\frac{1}{2}} H \mathcal{W} D_{\mathcal{E}}^{-1} H^\mathsf{T} D_{\mathcal{V}}^{-\frac{1}{2}} \mathcal{X}^{l^h} \Theta^{l^h})$. The purpose of $D_{\mathcal{V}}$ and $D_{\mathcal{E}}$ is to normalize the matrix H, further simplified to $\mathcal{X}^{l^h+1} = H \mathcal{W} H^\mathsf{T} \mathcal{X}^{l^h} \Theta^{l^h}$. Considering the $\mathcal{W}$ as the identity matrix, $\mathcal{X}^{l^h+1} = H H^\mathsf{T} \mathcal{X}^{l^h} \Theta^{l^h}$ is obtained. Subsequently, the feature aggregation process can be simplified and represented as follows:

$$\mathcal{X}^{l^h+1} = H \cdot H^\mathsf{T} \cdot \mathcal{X}^{l^h}. \tag{4}$$

During node feature updates, we first aggregate hyperedge features using $H^\mathsf{T} \cdot \mathcal{X}^{l^h}$, followed by updating node features using $H \cdot H^\mathsf{T} \cdot \mathcal{X}^{l^h}$. The $H \cdot H^\mathsf{T}$ signifies the weight adjustments post feature updates, reflecting the frequency of shared hyperedges between pairs of nodes. Consequently, a higher correlation coefficient arises from the increased frequency of shared hyperedges between two nodes, resulting in more enriched features during the aggregation process. Given that nodes exist across multiple hyperedges, does this suggest that these nodes offer substantial feature information during the aggregation process?

Examining the cyclic structure of the network tackles this challenge and introduces the notion of "cycle ratio" quantifying a node's involvement in the shortest cycles of other

nodes (Fan et al. 2021). Inspired by this, if we consider hyperedge as cycles, then $H^{cycle} = H \cdot H^{\mathsf{T}}$ symbolizes the "cyclenumber matrix" within the hypergraph. The sum of the participation rates of node v_i in the hyperedge involving itself and other nodes is the "cycle ratio", which is:

$$p_{v_i} = \sum\nolimits_{j=1}^{\mu_{\mathcal{E}}} \frac{H_{ij}^{cycle}}{H_{jj}^{cycle}}, \tag{5}$$

where H_{ij}^{cycle} denotes the count of hyperedges in which both nodes v_i and v_j are participants, while H_{jj}^{cycle} represents the number of hyperedges involving node v_j. $\mu_{\mathcal{E}}$ denotes the number of hyperedges. p_{v_i} reflects the importance of node v_i within the hypergraph. This novel idea suggests that the more involved an individual is within a community (sharing hyperedges with neighbors) and the wider range of social roles they assume (including hyperedges they are associated with), the more significant their role becomes. Subsequently, we calculate the "cycle ratio" of all nodes in the hypergraph through Eq. (5) and sorting them, the most important elite node v_{elite} is obtained. The hyperedge that includes elite node v_{elite} is recognized as elite hyperedges $\mathcal{E}_{elite}$.

Homogeneous Node Generator To generate homogeneous nodes with features resembling the elite node, we employ Kernel Density Estimation function $\hat{f}_{K_{de}}(x)$ (Wkeglarczyk 2018), defined as:

$$\hat{f}_{K_{de}}(x) = \frac{1}{\mathcal{F}} \sum\nolimits_{i=1}^{\mathcal{F}} K_{de}(x - x_i). \tag{6}$$

Here, $\mathcal{F}$ denotes the sample size, which is equivalent to the dimensionality of the features. x_i represents i-th data point of features vector x. K_{de} refers to the kernel function, utilized to assess the density contribution in the vicinity of data points. By placing kernel functions around data points and blending them with weights, a smoothed density estimation is achieved over the entire spatial domain.

Initially, we utilize the KDE function $\hat{f}_{K_{de}}(x)$ to fit features of the elite node v_{elite}. The fitting process is as follows:

$$\Phi_{elite} = \hat{f}_{K_{de}}(z_{elite}), \tag{7}$$

where z_{elite} denotes the feature of v_{elite}. Φ_{elite} is the probability density function obtained by fitting the features z_{elite}.

Introducing an excessive number of homogeneous nodes elevate the dimensionality of the H and escalate the computational complexity of the HGNNs. Consequently, Φ_{elite} are sampled to obtain preliminary features $\hat{z}_m$ of single homogeneous node v_{mal} with a similar distribution of features as the elite node v_{elite}. The dimension $\mathcal{F}$ of $\hat{z}_m$ is the same as features of v_{elite}. This process can be formalized as:

$$\hat{z}_m = \Phi_{elite}.sample(1 \times \mathcal{F}). \tag{8}$$

While the v_{mal} exhibits group identity of the elite hyperedge similarity to v_{elite}, features of nodes other than v_{elite} within the hyperedge are also essential. Given that v_{elite} exists in multiple elite hyperedges, selecting all elite hyperedges will increase complexity. Therefore, we randomly select an elite hyperedge $e_{elite} = \{v_1, \ldots, v_t\}$ and further optimize the generation of $\hat{z}_m$ using the feature information

from other nodes in hyperedge e_{elite}, represented as follows:

$$\hat{z}_{ms} = \sigma(((\hat{z}_m \otimes \Theta^1) \otimes \Theta^2) \cdots \otimes \Theta^{l^h}), \tag{9}$$
$$\hat{z}_{msd} = r_{elite} \oplus r_{mean} \oplus \hat{z}_{ms}, \tag{10}$$
$$\hat{z}_{mal} = NN(\hat{z}_{msd}). \tag{11}$$

Herein, $r_{mean} = \frac{r_1 + \cdots + r_t}{t-1}$, $\{r_1, \ldots, r_t\}$ denote the node embedding acquired for nodes $\{v_1, \ldots, v_t\}$ via the surrogate model HGNNs. r_{elite} is node embedding of v_{elite} in e_{elite}. t signifies the size of the elite hyperedge e_{elite}. $\{\Theta^1, \Theta^2, \ldots, \Theta^{l^h}\}$ represent the inter-layer weight parameters trained within the surrogate model HGNNs. The $\otimes$ denotes the element-wise multiplication operation between vectors, while $\oplus$ signifies the concatenation operation between vectors. $\hat{z}_{ms}$ denotes the homogeneous feature vector further computed using the trained weights in HGNNs. Subsequently, the homogeneous feature vector $\hat{z}_{msd}$ undergoes training through a linear neural network layer NN, where NN is expressed as $Y = W_{NN}\hat{z}_{msd} + b_{offset}$. Here, W_{NN} denotes the weight parameters of NN, and b_{offset} represents the bias term. Consequently, we obtain the homogeneous node v_{mal} with homogeneous feature $\hat{z}_{mal}$.

Injection Attack for HGNNs To achieve excellent attack performance with a lower attack cost, we inject single homogeneous node v_{mal} into $\mathcal{E}_{elite}$. The generated homogeneous node v_{mal} serves as an attacker injected into elite hyperedges $\mathcal{E}_{elite}$, thereby propagating malicious information during feature aggregation of HGNNs and increasing the imperceptibility of attacks.

Given the hyperedge incidence matrix of the original hypergraph $\mathcal{G}$ as $H^{\mu_{\mathcal{V}} \times \mu_{\mathcal{E}}}$, where $\mu_{\mathcal{V}}$ is the number of nodes and $\mu_{\mathcal{E}}$ is the number of hyperedges. The single homogeneous node v_{mal} is injected into elite hyperedges $\mathcal{E}_{elite}$, resulting in the attacked hypergraph $\hat{\mathcal{G}}$. The hyperedge incidence matrix of $\hat{\mathcal{G}}$ is updated to $\hat{H}^{(\mu_{\mathcal{V}}+1) \times \mu_{\mathcal{E}}}$.

For instance, if $\mathcal{E}_{elite} = \{e_{elite1}, \ldots, e_{elitej}\}$ and $e_{elitej} = \{v_1, v_5, v_i\}$, injecting the homogeneous node v_{mal} into e_{elitej}, results in the attacked hyperedge $\hat{e}_{elitej} = \{v_1, v_5, v_i, v_{mal}\}$. Consequently, the value at the corresponding position $\{(\mu_{\mathcal{V}}+1), j\}$ in the $\hat{H}^{(\mu_{\mathcal{V}}+1) \times \mu_{\mathcal{E}}}$ matrix is set to 1. $(\mu_{\mathcal{V}}+1)$ is the position of v_{mal} in the node set $\mathcal{V}$. j denotes the j-th hyperedge of the elite hyperedge e_{elitej} in the hyperedges set $\mathcal{E}$.

The attacked $\hat{H}$ resulting from injecting the homogeneous node v_{mal} into elite hyperedges $\mathcal{E}_{elite}$ is described as:

$$\hat{H} = \begin{cases} \hat{H}_{ij} = 1, & if \quad v_i \in e_j, \\ \hat{H}_{ij} = 0, & if \quad v_i \notin e_j, \\ \hat{H}_{(\mu_{\mathcal{V}}+1)j} = 1, & if \quad v_{mal} \in e_{elitej}. \end{cases} \tag{12}$$

It is evident that the node dimension of $\hat{H}$ has increased by one, while the hyperedge dimension remains unchanged. Additionally, the feature matrix $\mathcal{X}$ has been augmented with the injected homogeneous node feature $\hat{z}_{mal}$. Therefore, we obtain the input data for adversarial attacks on HGNNs, comprising the attacked incidence matrix $\hat{H}$ and the perturbed feature attributes $\hat{\mathcal{X}} = \{\mathcal{X}, \hat{z}_{mal}\}$.

HGNNs aggregate the malicious feature information in the attacked hypergraph, leading to a degradation in HGNNs. $\widehat{\mathcal{Z}}$ is the output of HGNNs after being attacked:

$$\widehat{\mathcal{Z}} = \sigma(\widehat{D}_{\mathcal{V}}^{-\frac{1}{2}} \widehat{H} \widehat{\mathcal{W}} \widehat{D}_{\mathcal{E}}^{-1} \widehat{H}^{\mathsf{T}} \widehat{D}_{\mathcal{V}}^{-\frac{1}{2}} \widehat{\mathcal{X}}^{l^h} \Theta^{l^h}). \quad (13)$$

Optimization Finally, we aim to enhance the attack effectiveness by training the IE-Attack, thereby improving the *Misclassification rate* of HGNNs. Consequently, the objective of model training is to minimize the discrepancy between predicted scores of correct labels and the predicted scores of the targeted incorrect labels, being depicted as:

$$\min_{\widehat{\mathcal{G}}} \mathcal{L}_{atk} = \sum_{q \in \mathcal{V}_{train}} \max_{z \neq y_q}(0, \widehat{\mathcal{Z}}_{q,y_q} - \widehat{\mathcal{Z}}_{q,z}) +$$
$$\| \widehat{z}_{mal} - z_{elite} \|, \quad (14)$$

where q denotes the index of training samples in the training set $\mathcal{V}_{train}$, y_q indicates the label of sample q, $\widehat{\mathcal{Z}}$ represents the classification prediction results of HGNNs on the attacked hypergraph $\widehat{\mathcal{G}}$, and z is the label of sample q obtained by the surrogate model HGNNs. The ReLU function $\max(0, \cdot)$ is employed to ensure the non-negativity of the loss function. The term $\| \widehat{z}_{mal} - z_{elite} \|$ represents the distance between the embeddings of the elite node v_{elite} and the generated homogeneous node v_{mal}. The purpose is to make the generated homogeneous node v_{mal} closer to the elite node v_{elite} with the group identity of the elite hyperedges, thereby improving the imperceptibility of attacks. The training process of the IE-Attack is guided by the attack loss $\mathcal{L}_{atk}$, optimizing iteratively with the gradient descent method until convergence.

Experiments

Experimental Setting

Datasets To validate the superiority of our method, five datasets (Cora, Citeseer, Pubmed, Chameleon, Lastfm) (Maurya, Liu, and Murata 2021) are adopted. Following the hypergraph generation methods in the HGNNs, we apply the Hyper-KNN and Hyper-$L1$ methods for hypergraph generation (Gao et al. 2022). Furthermore, we introduce a novel approach for constructing hypergraphs by considering higher-order neighbors of nodes (Hyper-HOR). Three distinct hypergraph generation strategies are employed to verify the efficacy and robustness of the proposed IE-Attack. According to the datasets partitioning strategy for node classification in Graph Convolutional Networks (GCNs) (Kipf and Welling 2017), datasets are divided into training/validation/test sets.

Parameter Setting In this study, we set the elite hyperedge perturbation budget η, which involves selecting $\eta \times \omega$ hyperedges within elite hyperedge $\mathcal{E}_{elite}$. The value of η ranges from 0.1 to 1 and ω is the number of $\mathcal{E}_{elite}$. Regarding the value of K in the Hyper-KNN hypergraph construction method, we set it to 10. The order in Hyper-HOR is set to 1-order and γ in Hyper-$L1$ is set to 0.1.

Evaluating Metrics This paper utilizes the *Misclassification rate* as an evaluation metric for the performance of IE-Attack. The *Misclassification rate* indicates the success rate

of misclassifications by HGNNs, where a higher *Misclassification rate* signifies a more effective attack.

Baselines IE-Attack is the node injection attack against HGNNs. Hence, we modify node injection methods from GNNs to align with the strategy proposed for hypergraphs in this study. We set up six baseline methods, including random methods (GIA-Random (GIA-R), DICE (Waniek et al. 2018)), gradient methods (FGA (Chen et al. 2018), IGA (Wu et al. 2019)), and adversarial generation (G-NIA (Tao et al. 2021), G-NIA_CANA (G-NIA*) (Tao et al. 2023)). These baseline methods adapt graph attacks to hypergraphs, making them comparable to proposed IE-Attack. Moreover, IE-Attack and baselines attack HGNNs in evasion attack scenarios. All experiments are conducted on a workstation equipped with four NVIDIA RTX 3090 GPUs, which are conducted under the same parameter settings. Except for Table 1, the random seed for other experiments is set 2024.

Model Performance and Parameter Analysis

Performance Comparison with State-of-the-art Methods Table 1 displays the *Misclassification rate* of IE-Attack and baseline methods across three hypergraph generation models on five datasets. The "Clean" shows *Misclassification rate* of the hypergraph without attacks in HGNNs. Except for G-NIA and G-NIA*, the other four baseline methods show only slight improvements in attack performance on hypergraphs compared to the "Clean" results. This suggests that randomly selecting hyperedges and using gradient guidance for node injections are not effective for hypergraph attacks. In contrast, G-NIA introduces injected nodes into hyperedges generated by a generator after multiple training iterations, leading to improved attack performance. G-NIA* incorporates a discriminator to enhance the imperceptibility of attacks. Nonetheless, IE-Attack achieves the highest *Misclassification rate* across all datasets and hypergraph generation models. These findings show attack effectiveness on HGNNs by injecting the homogeneous node into elite hyperedges. Moreover, the Cora hypergraph generated by the Hyper-HOR shows higher susceptibility to attacks, indicating weaker robustness. Conversely, the Citeseer hypergraph generated by Hyper-KNN and Hyper-$L1$ is more vulnerable to such attacks. These differences arise from how models capture graph structural information and node features.
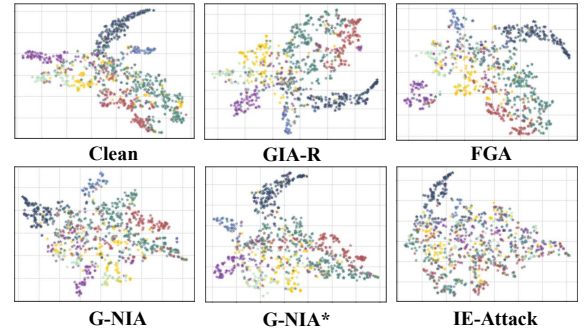

Clean   **GIA-R**   **FGA**

G-NIA   **G-NIA***   **IE-Attack**

Figure 3: Visualization of node classification obtained by HGNNs after attack on Cora under Hyper-KNN.

Datasets	Hyper-model	Clean	GIA-R	DICE	FGA	IGA	G-NIA	G-NIA*	IE-Attack
Cora	Hyper-KNN	25.24±2.69	25.60±2.71	25.65±2.71	25.86±2.71	25.21±2.69	52.09±3.09	34.50±2.94	**55.92±3.02**
	Hyper-HOR	25.35±2.70	25.48±2.70	25.68±2.71	25.68±2.71	25.53±2.70	37.51±2.99	29.57±2.83	**46.37±3.09**
	Hyper-$L1$	35.30±2.96	35.77±2.97	36.48±2.98	36.16±2.98	35.75±2.97	42.15±3.05	39.60±2.30	**55.60±3.08**
Citeseer	Hyper-KNN	42.23±3.07	42.26±3.06	42.32±3.06	42.55±3.06	42.96±3.07	52.93±5.62	56.93±3.06	**72.00±2.78**
	Hyper-HOR	45.51±3.09	45.31±3.09	45.57±3.09	45.33±3.09	45.35±3.09	45.30±5.64	49.27±3.10	**53.20±3.09**
	Hyper-$L1$	43.20±3.07	43.71±3.08	43.14±3.07	44.59±3.08	43.60±3.07	59.53±4.55	49.67±5.66	**70.61±2.82**
Pubmed	Hyper-KNN	18.05±1.06	18.15±1.07	18.17±1.07	18.34±1.07	18.16±1.07	20.64±1.03	20.15±1.06	**22.42±1.15**
	Hyper-HOR	16.91±1.04	16.66±1.03	16.79±1.04	16.77±1.04	16.74±1.04	23.18±1.45	23.94±1.37	**26.13±1.22**
	Hyper-$L1$	22.80±1.24	22.81±1.16	22.71±1.16	22.77±1.16	22.92±1.16	24.06±1.06	24.85±1.07	**27.02±1.18**
Chameleon	Hyper-KNN	40.98±4.32	41.24±4.32	40.50±4.31	41.10±4.31	41.04±4.32	46.64±4.36	43.60±4.16	**47.98±4.38**
	Hyper-HOR	55.56±4.35	57.16±4.34	56.62±4.35	56.42±4.34	56.54±4.35	58.86±4.32	59.40±4.31	**70.26±3.96**
	Hyper-$L1$	74.20±3.84	74.92±3.08	75.24±3.78	75.04±3.79	75.50±3.77	75.80±2.67	75.20±3.79	**79.00±3.57**
Lastfm	Hyper-KNN	63.85±1.71	66.52±1.69	66.58±1.69	66.47±1.69	66.26±1.69	65.26±4.18	65.96±1.67	**67.81±1.67**
	Hyper-HOR	56.71±1.76	56.90±1.78	56.38±1.77	56.95±1.77	56.80±1.77	59.58±4.30	62.53±1.57	**64.57±1.71**
	Hyper-$L1$	70.00±1.61	70.44±1.26	70.42±1.62	70.27±1.62	70.43±1.67	70.40±3.01	69.77±1.44	**72.87±1.16**

Table 1: Comparison of *Misclassification rate* (%) of IE-Attack and baselines. The results are averaged over 10 runs and the half width of the 95% confidence interval. The best results are highlighted in bold.

Dataset	Hyper-model	PCA							HBOS						
		GIA-R	DICE	FGA	IGA	G-NIA	G-NIA*	IE-Attack	GIA-R	DICE	FGA	IGA	G-NIA	G-NIA*	IE-Attack
Cora	Hyper-KNN	25.40	26.10	26.00	25.90	45.40	25.30	**49.50**	26.70	26.70	26.90	27.10	45.40	26,40	**57.80**
Citeseer	Hyper-KNN	40.70	40.00	40.90	40.20	42.60	41.10	**79.20**	43.00	43.70	42.70	42.80	43.10	44.30	**78.40**
Pubmed	Hyper-KNN	17.92	17.80	17.96	17.80	19.05	19.23	**20.02**	18.04	18.06	18.50	18.64	20.14	20.21	**22.62**
Chameleon	Hyper-KNN	39.60	39.60	39.60	39.60	44.60	44.20	**45.60**	42.00	42.20	41.40	41.80	46.20	39.40	**46.57**
Lastfm	Hyper-KNN	64.76	65.00	64.83	64.86	65.00	66.10	**70.16**	65.23	65.23	65.26	65.26	66.00	67.23	**70.80**

Table 2: Comparison of *Misclassification rate* (%) of IE-Attack and baselines under two detection methods.

To demonstrate the impact of IE-Attack on HGNNs' classification performance, we visualize the node classification results of Cora under baselines in Figure 3. Under the IE-Attack, HGNNs experience a significant decrease in classification performance with substantial data point dispersion.

Performance Comparison under Detection Methods To validate the imperceptibility of IE-Attack, we further attack HGNNs under two detection models (i.e., PCA, HBOS) to obtain attack results, as shown in Table 2. The results demonstrate that even under the detection models, IE-Attack maintains excellent attack performance compared to other baselines. This indicates that proposed node generator with group identity generates homogeneous nodes, while possessing the capability to propagate malicious information. Furthermore, injecting the single homogeneous node into elite hyperedges is also an outstanding strategy to enhance the imperceptibility of attacks.

Performance of Other Methods for Elite Hyperedges To verify that elite nodes identified by the "cycle ratio" effectively enhance attack performance, we compare IE-Attack with other node importance metrics on five datasets (Cora, Citeseer, Pubmed, Chameleon, Lastfm). In Figure 4(a), the bar graph illustrates the attack performance using different metrics for the hypergraph generation methods. The height of the bar represents the attack performance of the model caused by the elite hyperedges obtained under this importance metric. The *Misclassification rate* of IE-Attack outperforms the other metrics, suggesting that obtain

elite hyperedges by utilizing the node spanning amplify the spread of malicious information.

Analysis of Perturbation Budget (η) of Elite Hyperedges Figure 5 compares the performance of IE-Attack and five baseline methods on four datasets (i.e., Cora, Citeseer, Pubmed, Chameleon) under different hypergraph models as elite hyperedge budget η varies from 0.1 to 1. For the hypergraphs constructed by Hyper-KNN and Hyper-$L1$, the attack performance of IE-Attack shows an increasing trend across the four datasets as the elite hyperedge budget η rises. Surprisingly, GIA-R, FGA, and IGA exhibit minimal changes in *Misclassification rate* as η increases, indicating that injecting the node into hyperedges randomly selecting hyperedges and using gradient guidance has little impact on attack performance. Additionally, G-NIA and G-NIA*, being node injection attack for graphs applied to hypergraphs, demonstrate lower performance than IE-Attack.
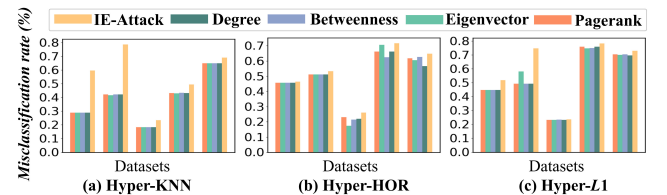

Figure 4: *Misclassification rate* (%) of IE-Attack compared to other methods for obtaining elite hyperedges.
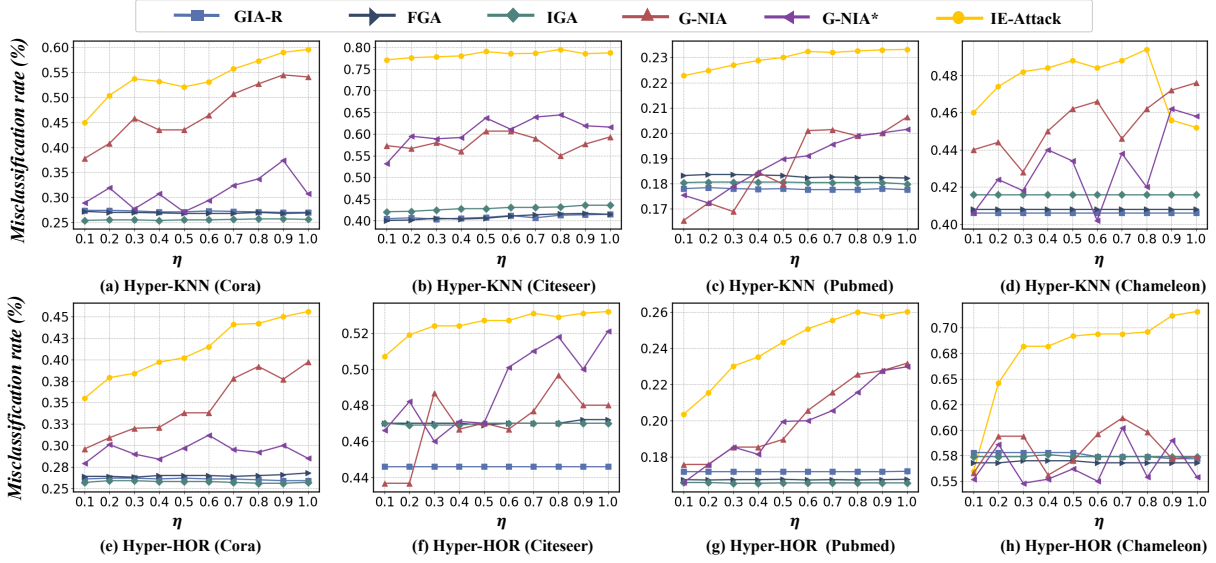
Figure 5: Attack performance of IE-Attack compared to baselines under elite hyperedge perturbation budget η.

Analysis of Kernel Function K_{de} Table 3 illustrates *Misclassification rate* when adopting different kernel functions, i.e., Gaussian(M_1), Tophat(M_2), and Epanechnikov(M_3). Generally, most datasets show optimal attack performance when generating node features using the Gaussian kernel function. Specifically, for the Chameleon, Tophat performs best in Hyper-KNN, while for Lastfm, Epanechnikov demonstrates superior performance in Hyper-HOR. Consequently, the Gaussian kernel function is chosen for generating node features in three citation datasets, while appropriate kernel functions are selected for the remaining two datasets.

K_{de}	Hyper-model	Cora	Citeseer	Pubmed	Chameleon	Lastfm
M_1	Hyper-KNN	**59.60**	**78.70**	**23.32**	45.20	**69.10**
	Hyper-HOR	**45.60**	**53.20**	**26.04**	**71.60**	64.70
	Hyper-$L1$	**51.70**	**74.70**	**23.66**	**78.20**	72.87
M_2	Hyper-KNN	41.60	63.30	22.70	**46.20**	68.70
	Hyper-HOR	41.60	52.60	24.50	67.40	64.73
	Hyper-$L1$	50.50	55.50	23.34	76.40	73.33
M_3	Hyper-KNN	38.60	63.90	22.76	45.20	68.87
	Hyper-HOR	41.60	52.20	24.58	67.20	**64.77**
	Hyper-$L1$	49.70	54.70	23.01	77.80	**73.37**

Table 3: Attack performance of different K_{de} in IE-Attack.

Ablation Study and Analysis

To evaluate the effectiveness of the proposed algorithm, we conduct ablation experiments by systematically removing the "Elite Hyperedges (R_1)", "KDE (R_2)" and "Node Generator (R_3)". Table 4 shows that excluding three strategies lead to a decrease in the model's attack performance, resulting in lower *Misclassification rate*. Particularly, the absence of the "Elite Hyperedges" and "Node Generator" have a more significant impact on the model's performance. Lever-

aging the elite hyperedge to maximize the malicious influence of the homogeneous node and utilizing the "Node Generator" to generate the homogeneous node, IE-Attack enhances its attack performance and improves the imperceptibility of attacks. Ablation experiments confirm the remarkable effectiveness of the proposed IE-Attack on HGNNs.

Methods	Hyper-model	Cora	Citeseer	Pubmed	Chameleon	Lastfm
w/o R_1	Hyper-KNN	30.80	40.70	18.28	43.80	64.83
	Hyper-HOR	27.10	45.10	16.84	59.00	56.50
	Hyper-$L1$	37.30	51.60	23.98	75.40	69.13
w/o R_2	Hyper-KNN	39.40	65.10	22.72	45.80	68.70
	Hyper-HOR	42.00	52.20	24.68	71.40	64.23
	Hyper-$L1$	50.80	54.90	25.92	78.20	**73.33**
w/o R_3	Hyper-KNN	29.50	54.40	19.40	44.40	64.60
	Hyper-HOR	25.80	48.10	16.88	53.00	55.17
	Hyper-$L1$	35.80	48.40	23.10	73.20	70.70
IE-Attack	Hyper-KNN	**59.60**	**78.70**	**23.32**	**49.40**	**69.10**
	Hyper-HOR	**45.60**	**53.20**	**26.04**	**71.60**	**64.70**
	Hyper-$L1$	**51.70**	**74.70**	**27.02**	**78.20**	72.87

Table 4: Attack performance of different IE-Attack variants.

Conclusion

This paper explores the phenomenon of node spanning in the hypergraph and the group identity of hyperedges, introducing the node injection attack framework IE-Attack for HGNNs. The key idea lies in injecting the homogeneous node into elite hyperedges, IE-Attack maximizes the spread of malicious information in feature aggregation of HGNNs and enhances the imperceptibility of attacks. Extensive experiments demonstrate IE-Attack's superior performance over other attack models. Future research will explore attack techniques for different HGNNs variants and varying levels of knowledge, including black-box attacks.

Acknowledgments

References

Antelmi, A.; Cordasco, G.; Polato, M.; Scarano, V.; Spagnuolo, C.; and Yang, D. 2023. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1): 1–38.

Bai, S.; Zhang, F.; and Torr, P. H. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110: 107637.

Battiston, F.; Amico, E.; Barrat, A.; Bianconi, G.; Ferraz de Arruda, G.; Franceschiello, B.; Iacopini, I.; Kéfi, S.; Latora, V.; Moreno, Y.; et al. 2021. The physics of higher-order interactions in complex systems. *Nature Physics*, 17(10): 1093–1098.

Benson, A. R.; Gleich, D. F.; and Leskovec, J. 2016. Higher-order organization of complex networks. *Science*, 353(6295): 163–166.

Bojchevski, A.; and Günnemann, S. 2019. Adversarial attacks on node embeddings via graph poisoning. In *International Conference on Machine Learning*, 695–704.

Chang, H.; Rong, Y.; Xu, T.; Huang, W.; Zhang, H.; Cui, P.; Zhu, W.; and Huang, J. 2020. A restricted black-box adversarial framework towards attacking graph embedding models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3389–3396.

Chen, J.; Wu, Y.; Xu, X.; Chen, Y.; Zheng, H.; and Xuan, Q. 2018. Fast gradient attack on network embedding. *arXiv preprint arXiv:1809.02797*.

Chen, Y.; Picek, S.; Ye, Z.; Wang, Z.; and Zhao, H. 2023. Momentum gradient-based untargeted attack on hypergraph neural networks. *arXiv preprint arXiv:2310.15656*.

Cheng, L.; Zhu, P.; Gao, C.; Wang, Z.; and Li, X. 2024a. A heuristic framework for sources detection in social networks via graph convolutional networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(11): 7002–7014.

Cheng, L.; Zhu, P.; Tang, K.; Gao, C.; and Wang, Z. 2024b. GIN-SD: source detection in graphs with incomplete nodes via positional encoding and attentive fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 55–63.

Dong, Y.; He, D.; Wang, X.; Li, Y.; Su, X.; and 0001, D. J. 2023. A generalized deep markov random fields framework for fake news detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 4758–4765.

Fan, T.; Lü, L.; Shi, D.; and Zhou, T. 2021. Characterizing cycle structure in complex networks. *Communications Physics*, 4(1): 272.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3558–3565.

Gao, Y.; Feng, Y.; Ji, S.; and Ji, R. 2022. HGNN+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3181–3199.

Howard, J. A. 2000. Social psychology of identities. *Annual Review of Sociology*, 26(1): 367–393.

Hu, C.; Yu, R.; Zeng, B.; Zhan, Y.; Fu, Y.; Zhang, Q.; Liu, R.; and Shi, H. 2023. Hyperattack: Multi-gradient-guided white-box adversarial structure attack of hypergraph neural networks. *arXiv preprint arXiv:2302.12407*.

Jin, D.; Feng, B.; Guo, S.; Wang, X.; Wei, J.; and Wang, Z. 2023. Local-global defense against unsupervised adversarial attacks on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8105–8113.

Jin, T.; Yu, Z.; Gao, Y.; Gao, S.; Sun, X.; and Li, C. 2019. Robust L2- Hypergraph and its applications. *Information Sciences*, 501: 708–723.

Ju, M.; Fan, Y.; Zhang, C.; and Ye, Y. 2023. Let graph be the go board: gradient-free node injection attack for graph neural networks via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4383–4390.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 1–14.

Li, Q.; Li, X.; Cui, X.; Tang, K.; and Zhu, P. 2023. Hept attack: heuristic perpendicular trial for hard-label attacks under limited query budgets. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 4064–4068.

Liu, S.; Wu, G.; Deng, X.; Lu, H.; Wang, B.; Yang, L.; and Park, J. J. 2023. Graph sampling based fairness-aware recommendation over sensitive attribute removal. In *2023 IEEE International Conference on Data Mining*, 428–437.

Maurya, S. K.; Liu, X.; and Murata, T. 2021. Improving graph neural networks with simple architecture design. *arXiv preprint arXiv:2105.07634*.

Shayegh, J.; Storey, L.; Turner, R. N.; and Barry, J. 2022. A social identity approach to how elite outgroups are invoked by politicians and the media in nativist populism. *Political Psychology*, 43(6): 1009–1025.

Spears, R. 2021. Social influence and group identity. *Annual Review of Psychology*, 72(1): 367–390.

Sun, L.; Dou, Y.; Yang, C.; Zhang, K.; Wang, J.; Philip, S. Y.; He, L.; and Li, B. 2022. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(8): 7693–7711.

Sun, Y.; Wang, S.; Tang, X.; Hsieh, T.-Y.; and Honavar, V. 2019. Node injection attacks on graphs via reinforcement learning. *arXiv preprint arXiv:1909.06543*.

Tang, K.; Ma, Y.; Miao, D.; Song, P.; Gu, Z.; Tian, Z.; and Wang, W. 2022a. Decision fusion networks for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

Tang, K.; Shi, Y.; Lou, T.; Peng, W.; He, X.; Zhu, P.; Gu, Z.; and Tian, Z. 2022b. Rethinking perturbation directions for imperceptible adversarial attacks on point clouds. *IEEE Internet of Things Journal*, 10(6): 5158–5169.

Tao, S.; Cao, Q.; Shen, H.; Huang, J.; Wu, Y.; and Cheng, X. 2021. Single node injection attack against graph neural networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1794–1803.

Tao, S.; Cao, Q.; Shen, H.; Wu, Y.; Hou, L.; Sun, F.; and Cheng, X. 2023. Adversarial camouflage for node injection attack on graphs. *Information Sciences*, 649: 119611.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. In *International Conference on Learning Representations*, 1–12.

Wang, J.; Luo, M.; Suya, F.; Li, J.; Yang, Z.; and Zheng, Q. 2020. Scalable attack on graph data by injecting vicious nodes. *Data Mining and Knowledge Discovery*, 34: 1363–1389.

Wang, X.; Dong, Y.; Jin, D.; Li, Y.; Wang, L.; and Dang, J. 2023. Augmenting affective dependency graph via iterative incongruity graph learning for sarcasm detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4702–4710.

Wang, X.; Liu, J.; Cheng, Y.; Liu, A.; and Chen, E. 2018. Dual hypergraph regularized PCA for biclustering of tumor gene expression data. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2292–2303.

Wang, Z.; Chen, J.; Shao, Z.; and Wang, Z. 2024. Dual-View desynchronization hypergraph learning for dynamic hyperedge prediction. *IEEE Transactions on Knowledge and Data Engineering*, 1–16.

Waniek, M.; Michalak, T. P.; Wooldridge, M. J.; and Rahwan, T. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2: 139–147.

Wei, J.; Yaxin, L.; Han, X.; Yiqi, W.; Shuiwang, J.; and Charu, A. 2020. Adversarial attacks and defenses on graphs: A review, a tool, and empirical studies. *arXiv preprint arXiv:2003.00653*, 00653.

Wkeglarczyk, S. 2018. Kernel density estimation and its application. In *ITM Web of Conferences*, volume 23, 00037.

Wu, H.; Wang, C.; Tyshetskiy, Y.; Docherty, A.; Lu, K.; and Zhu, L. 2019. Adversarial examples on graph data: Deep insights into attack and defense. *arXiv preprint arXiv:1903.01610*.

Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; and Talukdar, P. 2019. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in Neural Information Processing Systems*, 32.

Yu, Y.; Yang, E.; Guo, G.; Jiang, L.; and Wang, X. 2023. Basket representation learning by hypergraph convolution on repeated items for next-basket recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2415–2422.

Zeng, Y.; Jin, Q.; Bao, T.; and Li, W. 2023. Multi-modal knowledge hypergraph for diverse image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 3376–3383.

Zhang, R.; Zou, Y.; and Ma, J. 2019. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. *arXiv preprint arXiv:1911.02613*.

Zhu, P.; Fan, Z.; Guo, S.; Tang, K.; and Li, X. 2024a. Improving adversarial transferability through hybrid augmentation. *Computers & Security*, 139: 103674.

Zhu, P.; Pan, Z.; Liu, Y.; Tian, J.; Tang, K.; and Wang, Z. 2024b. A general black-box adversarial attack on graph-based fake news detectors. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 568–576.

Zhu, P.; Pan, Z.; Tang, K.; Cui, X.; Wang, J.; and Xuan, Q. 2024c. Node injection attack based on label propagation against graph neural network. *IEEE Transactions on Computational Social Systems*, 11(5): 5858–5870.

Zou, X.; Zheng, Q.; Dong, Y.; Guan, X.; Kharlamov, E.; Lu, J.; and Tang, J. 2021. Tdgia: Effective injection attacks on graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2461–2471.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2847–2856.