CLUSTER ATTACK: Query-based Adversarial Attacks on Graphs with Graph-Dependent Priors

Zhengyi Wang[1,3], **Zhongkai Hao**[1], **Ziqiao Wang**[1], **Hang Su**[*1,2,3] and **Jun Zhu**[*1,2,3]
[1]Department of Computer Science & Technology, Institute for AI, BNRist Center
Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University
[2]Peng Cheng Laboratory
[3] Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute
{wang-zy21, hzj21, ziqiao-w20}@mails.tsinghua.edu.cn, {suhangss, dcszj}@tsinghua.edu.cn

Abstract

While deep neural networks have achieved great success in graph analysis, recent work has shown that they are vulnerable to adversarial attacks. Compared with adversarial attacks on image classification, performing adversarial attacks on graphs is more challenging because of the discrete and non-differential nature of the adjacent matrix for a graph. In this work, we propose Cluster Attack — a Graph Injection Attack (GIA) on node classification, which injects fake nodes into the original graph to degenerate the performance of graph neural networks (GNNs) on certain victim nodes while affecting the other nodes as little as possible. We demonstrate that a GIA problem can be equivalently formulated as a graph clustering problem; thus, the discrete optimization problem of the adjacency matrix can be solved in the context of graph clustering. In particular, we propose to measure the similarity between victim nodes by a metric of *Adversarial Vulnerability*, which is related to how the victim nodes will be affected by the injected fake node, and to cluster the victim nodes accordingly. Our attack is performed in a practical and unnoticeable query-based black-box manner with only a few nodes on the graphs that can be accessed. Theoretical analysis and extensive experiments demonstrate the effectiveness of our method by fooling the node classifiers with only a small number of queries.

1 Introduction

Graph neural networks (GNNs) have obtained promising performance in various applications to graph data [Ying *et al.*, 2018; Qiu *et al.*, 2018; Chen *et al.*, 2019]. Recent studies have shown that GNNs, like other types of deep learning models, are also vulnerable to adversarial attacks [Dai *et al.*, 2018; Zügner *et al.*, 2018]. However, there still exists a gap between most of the existing attack setups and practice where the capability of an attacker is limited. Instead of directly modifying the original graph (aka., Graph Modification Attack, GMA) [Yang and Long, 2021], we focus on a more practical setting to inject extra fake nodes into the original graph (aka., Graph Injection Attack, GIA) [Zou *et al.*, 2021]. We perform query-based attack, which indicates that the attacker has no knowledge on the victim model but can access the model with a limited number of queries. As an example in Figure 1, high-quality users in a social network may be misclassified as low-quality users after being connected with a fraudulent user whose features (maybe the meta data of the account) are carefully crafted utilizing query information from the target model.

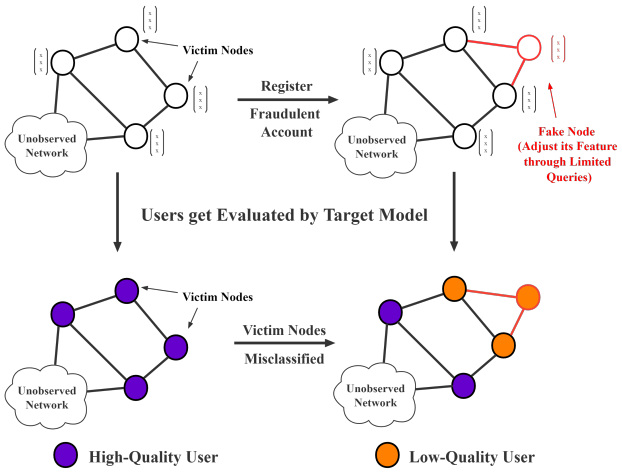


Figure 1: An illustration of query-based graph injection attack with partial information.

Compared to adversarial attacks on image classification [Chen *et al.*, 2017; Ilyas *et al.*, 2018; Dong *et al.*, 2018], the study of query-based adversarial attacks on graph data is still at an early stage. Existing attempts include training a surrogate model using query results [Wan *et al.*, 2021], a Reinforcement-Learning-based method [Ma *et al.*, 2021], derivative-free optimization [Yang and Long, 2021] and a gradient-based method [Mu *et al.*, 2021]. However, most of the existing attacks simply adopt the optimization methods from other fields, such as image adversarial attacks, without utilizing the rich structure of graph data, which has considerable potential to achieve a higher performance attack.

*Corresponding author.

Unlike most previous work, we only allow the adversary to access the information of a small part of the nodes since it is usually impossible to observe the whole graph, especially for large networks in practical scenarios. Moreover, we perform a *black-box* attack, which does not allow the adversary to have access to the model structures or parameters. The adversary has only a limited number of queries on the victim model about the predicted scores of certain nodes, which is more practical. It is also noted that, owing to the non-i.i.d. nature of graph data, connecting victim nodes to fake nodes may have side effects on the accuracy of victim nodes' neighbors, which is not our purpose. To our best knowledge, this is the first attempt to limit the influence to a certain range of victim nodes and protect the other nodes from being misclassified simultaneously.

We propose a unified framework for query-based adversarial attacks on graphs, which subsumes existing methods. In general, the attacker decides on the current perturbation as a conditional distribution on history query results and current graph status. Under this framework, we propose a novel attack method named *Cluster Attack*, which considers the graph-dependent priors by better utilizing the unique structure of the graph. In particular, we try to find an equivalent discrete optimization problem. We first demonstrate that a GIA problem can be formulated as an equivalent graph clustering problem. Because the discrete optimization problem of an adjacent matrix can be solved in the context of graph clustering, we prevent query-inefficient searching in the non-Euclidean space. The resulting cluster serves as a graph-dependent prior for the adjacent matrix, which utilizes the vulnerability of the local structure. Second, the key challenge in graph clustering is to define the similarity metric between nodes. We propose a metric to measure the similarity of victim nodes, called Adversarial Vulnerability; this is related to how the victim nodes will be affected by the injected fake node, and we cluster the victim nodes accordingly. The Adversarial Vulnerability is only related to the local structure of the graph and thus can be handled with only part of the graph observed.

Our contributions are summarized as follows:

- We propose a unified framework for query-based adversarial attacks on graphs, which formulates the current perturbation as a conditional distribution on the history of query results and on the current graph status.
- We propose *Cluster Attack*, an injection adversarial attack on a graph, which formulates a GIA problem as an equivalent graph clustering problem and thus solves the discrete optimization of an adjacent matrix in the context of clustering.
- After providing theoretical bounds on our method, we empirically show that our method achieves high performance in terms of success rate of attacks under an extremely strict setting with a limited number of queries and only part of the graph observed.

2 Background

In this section, we present recent works on node classification and adversarial attacks on graphs.

2.1 Node Classification on a Graph

Node classification on graphs is an important task, with a wide range of applications such as user classification in financial networks. It aims to carry out classification by aggregating the information from neighboring nodes [Kipf and Welling, 2017; Hamilton *et al.*, 2017; Veličković *et al.*, 2017]. Recent work has carried out node classification using graph convolutional networks (GCNs) [Kipf and Welling, 2017], which is one of the most representative GNNs. Specifically, let a graph be $G = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A}$ and $\mathbf{X}$ respectively represent the adjacency matrix and the feature matrix. Given a subset of labeled nodes in the graph, GCN aims to predict the labels of the remaining unlabeled nodes in the graph as

$$f(G) = f(A, X) = \text{softmax}\left(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)}\right), \quad (1)$$

where $\hat{\mathbf{A}}$ is the normalized adjacency matrix; $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are parameter matrices; σ is the activation function; and $f(G)$ is the prediction corresponding to each node.

2.2 Graph Adversarial Attacks

Numerous methods have been developed to perform adversarial attacks on graphs. Early works focused on modifying the original graph (i.e., Graph Modification Attack) [Dai *et al.*, 2018; Zügner *et al.*, 2018], while some recent works [Zou *et al.*, 2021; Tao *et al.*, 2021] have focused on a more practical setting to inject extra fake nodes into the original graph (i.e., Graph Injection Attack). For query-based graph adversarial attacks, as shown in Table 1, various efforts have been made. Some have focused on attacking the task of node classification [Yang and Long, 2021] while there also have been efforts to attack graph classification [Wan *et al.*, 2021; Ma *et al.*, 2021; Mu *et al.*, 2021], with gradient-based [Mu *et al.*, 2021] or gradient-free methods [Yang and Long, 2021; Wan *et al.*, 2021; Ma *et al.*, 2021]. Nevertheless, most of the existing attacks just adopt optimization methods from other fields (especially image adversarial attack), ignoring the unique structure of graph data. In this work, we propose to attack in a graph-specific manner utilizing the inherent structure of a graph.

3 A Unified Framework for Query-Based Adversarial Attacks on Graphs

We now present a unified framework for query-based adversarial attacks as well as the threat model and loss function.

3.1 Graph Injection Attack

Given a small set of victim nodes $\Phi_{\mathbf{A}} \subseteq \Phi$ in the graph, the goal of graph injection attack is to perform mild perturbations on the graph $G = (\mathbf{A}, \mathbf{X})$, leading to $G^+ = (\mathbf{A}^+, \mathbf{X}^+)$, such that the predicted labels of the victim nodes in $\Phi_{\mathbf{A}}$ are changed into the target labels. This goal is usually achieved by optimizing the adversarial loss $\mathcal{L}(\cdot)$ under constraints as:

$$\min_{G^+} \mathcal{L}(G^+) \; s.t. \; \text{dist}(G, G^+) \leq \Delta, \quad (2)$$

where $\text{dist}(G, G^+)$ denotes the magnitude of perturbation and has to be within the adversarial budget Δ. In this section, it

Method	Optimization Step	Target Task
Random	$\Delta G \sim$ Random	-
GRABNEL [Wan *et al.*, 2021]	$\Delta G = \underset{\Delta G}{\operatorname{argmin}} \mathcal{L}_{sur}(G + \Delta G)$	Graph Classification (GMA+GIA)
DFO [Yang and Long, 2021]	$\Delta G = F(G, \delta) - G$, $\delta \sim$ DFO	Node Classification (GMA)
[Mu *et al.*, 2021]	$\nabla p(\Delta A) = \frac{1}{Q}\sum_1^Q \operatorname{sgn}(\frac{p(\Delta A+\mu u_q)-p(\Delta A)}{\mu} u_q)$, $u_q \sim$ Gaussian	Graph Classification (GMA)
Rewatt [Ma *et al.*, 2021]	$\Delta G \sim p(\cdot\|G)$ from Reinforcement Learning Agent	Graph Classification (GMA)
Cluster Attack (Ours)	$\begin{cases} \Delta \mathbf{X}_{fake} = \mathbb{I}(\mathcal{L}(\mathbf{A}^+, \mathbf{X}^+) > \mathcal{L}(\mathbf{A}^+, \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{fake} + \delta \mathbf{X}_{ij} \end{bmatrix})) \cdot \delta \mathbf{X}_{ij} \\ \nabla_{\mathbf{X}_{fake}} \mathbb{E}\, \mathcal{L}(\mathbf{A}^+, \mathbf{X}^+) = \frac{1}{\sigma n}\sum_{i=1}^n Z_i \mathcal{L}(\mathbf{A}^+, \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{fake} + \sigma Z_i \end{bmatrix}) \end{cases}$ $\Delta A \sim$ cluster prior	Node Classification (GIA)

Table 1: Existing query-based methods on graph adversarial attacks.

can be specified as $|\Phi_{fake}| \leq \Delta_{fake}$ and $\sum_{u\in\Phi_{fake}} d(u) \leq \Delta_{edge}$ with $d(u)$ being the degree of node u.

In graph injection attacks, we have the augmented adjacent matrix $\mathbf{A}^+ = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{A}_{fake} \end{bmatrix}$ and the augmented feature matrix $\mathbf{X}^+ = \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{fake} \end{bmatrix}$. We further use $\Phi^+ = \Phi \cup \Phi_{fake}$ to denote the node set of G^+. In particular, $\mathbf{X}_{fake}$ corresponds to the feature of fake nodes; $\mathbf{B}$ denotes the connections between fake nodes and original nodes, and $\mathbf{A}_{fake}$ denotes the mutual connections between fake nodes. The malicious attacker manipulates $\mathbf{A}_{fake}, \mathbf{B}$ and $\mathbf{X}_{fake}$, leading to as low classification accuracy on $\Phi_{\mathbf{A}}$ as possible.

In a query-based adversarial attack on graphs, we uniformly formulate the update of a graph at time t as

$$\Delta G_{(t)} \sim p\big(\Delta G_{(t)} | \{f(G_i) | i = 1, 2, ..., q_t\}, G_t\big), \quad (3)$$

where $\{f(G_i)|i = 1, 2, ..., q_t\}$ denotes the feedback (hard labels or predicted values) of total q_t queries in the history from the target model. A curated list of current query-based graph adversarial attacks is shown in Table 1. In general, perturbation $\Delta G_{(t)}$ in time step t is conditioned on the history of query results $\{f(G_i)|i = 1, 2, ..., q_t\}$ and the current graph status G. Previous work has focused on using different optimization methods, including reinforcement learning [Ma *et al.*, 2021] and gradient-based optimization [Mu *et al.*, 2021] to decide ΔG without utilizing the graph structure explicitly.

3.2 Threat Model

Adversary Capability. We greatly restrict the attacker's ability so that we can only make connections between victim nodes and fake nodes. No connections can be made between fake nodes because connected malicious fake nodes are easier for detectors to locate. The number of new edges Δ_{edge} is set as barely the number of victim nodes Φ_A, which means that each victim node is connected by only one new edge.

Protected Nodes. As mentioned above, owing to the non-i.i.d nature of graph data, attacking victim nodes may have unintended side effects on their neighboring nodes. While attacking victim nodes, we simultaneously aim to keep the labels of untargeted nodes unchanged, to make our perturbation unnoticeable. In our setting, we try to protect $\mathcal{N}_k(\Phi_A)$, which are the neighbors of the victim nodes within k-hop.

Partial Information. It is practical to assume that the attacker has access to only part of the graph when conducting the attack. As mentioned above, we adopt an extremely strict setting so that we can only observe the features and connections of the observed nodes defined as

$$\Phi_o = \Phi_A \cup \mathcal{N}_k(\Phi_A) \cup \Phi_{fake}. \quad (4)$$

Limited Queries. It is practical in real scenarios that we have a limited number of queries to the victim model rather than full outputs of arbitrarily many chosen inputs. In our setting, we can query at most K times in total for the predicted scores of the observed nodes. The architecture and parameters about the victim model are unknown to the attacker.

3.3 Loss Function

We aim to make the classifier misclassify as many nodes as possible in the victim set of $\Phi_{\mathbf{A}}$. As it is nontrivial to directly optimize the number of misclassified nodes since the objective is discrete, we choose to optimize a surrogate loss function:

$$\min_{G^+} \mathcal{L}(G^+) \triangleq \sum_{v\in\Phi_{\mathbf{A}}} \ell(G^+, v) + \lambda \sum_{v\in\mathcal{N}_k(\Phi_{\mathbf{A}})} \ell_{\mathcal{N}}(G^+, v),$$
$$s.t.\ \operatorname{dist}(G, G^+) \leq \Delta, \quad (5)$$

where $\ell(G^+, v)$ and $\ell_{\mathcal{N}}(G^+, v)$ represent the loss functions for each victim node and for a protected node, respectively. A smaller $\ell(G^+, v)$ means that node v is more likely to be misclassified by the victim model f; by contrast, a smaller $\ell_{\mathcal{N}}(G^+, v)$ means that the predicted label of node v is less likely to be changed during our attack. In particular, we design our loss in the manner of the C&W loss [Carlini and Wagner, 2017], and define:

$$\ell(G^+, v) = \sigma\left(\max_{y_i \neq y_t} ([f(\mathbf{A}^+, \mathbf{X}^+)]_{v,y_i}) - [f(\mathbf{A}^+, \mathbf{X}^+)]_{v,y_t}\right), \quad (6)$$

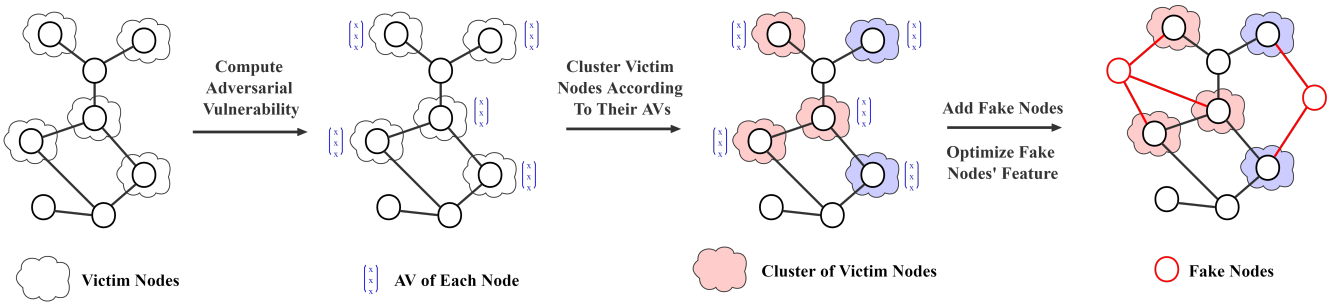


Figure 2: An illustration of Cluster Attack. We first compute Adversarial Vulnerability for each victim node with a limited number of queries; after that, we cluster the victim nodes and inject fake nodes accordingly; and finally we optimize the fake nodes' features.

where y_t stands for the target label of node v and the attacker succeeds only when node v is misclassified as y_t. $\sigma(x) = \max(x, 0)$. $[f(G^+)]_{v,y_i}$ denotes the output logit of node v of class y_i. For protected nodes, we define:

$$\ell_{\mathcal{N}}(G^+, v) = \\ \sigma\left(\max_{y_i \neq y_g} [f(\mathbf{A}^+, \mathbf{X}^+)]_{v,y_i}) - [f(\mathbf{A}^+, \mathbf{X}^+)]_{v,y_g}\right), \tag{7}$$

where y_g is the ground-truth label of v from the victim model.

4 Cluster Attack with Graph-Dependent Priors

4.1 Cluster Attack

The combinatorial optimization problem (5) is hard to solve owing to the non-Euclidean nature of the adjacent matrix and the complex structure of neural networks. To tackle this, we tried to find an equivalent combinatorial optimization problem and transform our GIA problem into a well-studied one. Here, we point out that every choice of adjacent matrix has an equivalent representation of a division of victim nodes into clusters. Thus, we get our key insight that this discrete optimization problem can be transformed into an equivalent graph clustering problem, which is a well-studied discrete optimization problem [Schaeffer, 2007].

Proposition 1 (GIA/Graph Clustering Equivalence)**.** *Given graph G and a set of nodes $\Phi_A \subseteq \Phi$, for a division of the victim nodes $\Phi_{\mathbf{A}}$ into N_{fake} clusters $C = \{C_1, C_2, ..., C_{N_{fake}}\}$, $\cup_{C_i \in C} C_i = \Phi_{\mathbf{A}}$, there exists a corresponding $\mathbf{B}$ and vice versa.*

Proof. We provide a one-to-one mapping between cluster C and adjacent matrix $\mathbf{B}$. Specifically, given $C = \{C_1, C_2, ..., C_{N_{fake}}\}, \cup_{C_i \in C} C_i = \Phi_{\mathbf{A}}$ we get $\mathbf{B}$ from

$$\mathbf{B}_{ij} = \begin{cases} 1, & \text{if } v_j \in C_i \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

We have

$$\Delta_{edge} = \sum_i \sum_j \mathbf{B}_{ij} = \sum_i |C_i| = |\Phi_A|. \tag{9}$$

Thus, we resultant $\mathbf{B}$ is valid in our setting.

Given $\mathbf{B}$ where $|\Phi_A| = \Delta_{edge} = \sum_i \sum_j \mathbf{B}_{ij}$, we derive the cluster C from

$$v_j \begin{cases} \in C_i, & \text{if } \mathbf{B}_{ij} = 1 \\ \notin C_i, & \text{if } \mathbf{B}_{ij} = 0 \end{cases}. \tag{10}$$

We have $\sum_i |C_i| = \sum_i \sum_j \mathbf{B}_{ij} = |\Phi_A|$.

In our setting, each victim node is connected to only one fake node, which indicates

$$\sum_i \mathbf{B}_{ij} = 1, \ \forall v_j \in \Phi_A. \tag{11}$$

In this case, each cluster gets disjoint with each other

$$C_i \cap C_j = \emptyset, \ \forall 1 \leq i, j \leq N_{fake}. \tag{12}$$

Then we get $\cup_{C_i \in C} C_i = \Phi_{\mathbf{A}}$ and C is a valid division. □

Because $\mathbf{A}_{fake} = \mathbf{0}$ is fixed in our setting, we formulate our graph injection attack problem as an equivalent graph clustering problem. As a result, the non-trivial discrete optimization problem of the adjacency matrix can be solved in the context of graph clustering. The resulting cluster serves as a graph-dependent prior for adjacent matrix $\mathbf{B}$, which prevents inefficient searching in non-Euclidean discrete space.

For graph clustering, the main concern is the metric of the similarity between victim nodes. To investigate how a fake node will affect the performance on a certain node, we propose Adversarial Vulnerability as the similarity metric for graph clustering. Adversarial Vulnerability of a victim node reflects its "most vulnerable angle" towards adversarial features of fake nodes, which is related only to the local structure of the graph and can be handled with only part of the graph observed. We have the insight that victim nodes sharing similar Adversarial Vulnerability are more likely to be affected simultaneously when they are connected to the same fake node.

Definition 1 (Adversarial Vulnerability)**.** *For victim node $v \in \Phi$, its Adversarial Vulnerability is defined as*

$$AV(v) = \underset{x_u}{argmin}\ \mathcal{L}(G^+), \tag{13}$$

where x_u denotes the feature of fake node u connected to node v. For the fake node itself, the Adversarial Vulnerability is defined as its own feature.

Here, we adopt Euclid's distance as distance metric between victim nodes' Adversarial Vulnerability which is in Euclidean feature space.

Definition 2 (Adversarial Distance Metric)**.** *$\forall v, u \in \Phi^+$, the Adversarial Distance Metric between node v and u is defined as $d(v,u) = ||AV(v) - AV(u)||_2^2$.*

After the Adversarial Vulnerability is computed, the objective of the cluster algorithm is to minimize the following cluster distance as

$$\min_{C} \sum_{C_i \in C} \sum_{v \in C_i} d(v, c_i), \tag{14}$$

where the cluster center c_i is the corresponding fake node of cluster C_i, and

$$\mathrm{AV}(c_i) = \frac{1}{|C_i|} \sum_{v \in C_i} \mathrm{AV}(v). \tag{15}$$

4.2 Optimization

To approximate Adversarial Vulnerability, we adopt zeroth-order optimization [Chen *et al.*, 2017], which is similar to query-based attacks on image classification, to better utilize limited queries. For graph with discrete features, the optimization of Eq. (13) can be

$$\Delta \mathbf{X}_{fake} = \mathbb{I}\left(\mathcal{L}(\mathbf{A}^+, \mathbf{X}^+) > \mathcal{L}\left(\mathbf{A}^+, \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{fake} + \delta \mathbf{X}_{ij} \end{bmatrix}\right)\right) \cdot \delta \mathbf{X}_{ij}, \tag{16}$$

where $\delta \mathbf{X}_{ij}$ denotes the tentative perturbation in dimension j of a feature of the ith fake node and $\mathbb{I}(\cdot)$ is the indicator function. A tentative perturbation is adopted only if it diminishes the adversarial loss. For continuous feature space, we adopt NES [Ilyas *et al.*, 2018] for gradient estimation as

$$\nabla_{\mathbf{X}_{fake}} \mathbb{E}\, \mathcal{L}(\mathbf{A}^+, \mathbf{X}^+) = \frac{1}{\sigma n} \sum_{i=1}^{n} Z_i \mathcal{L}\left(\mathbf{A}^+, \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{fake} + \sigma Z_i \end{bmatrix}\right), \tag{17}$$

where $\sigma > 0$ is the standard variance, n is the size of the NES population and $Z_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{N_{fake} \times D})$ is the perturbation of $\mathbf{X}_{fake}$. After gradient estimation, gradient-based optimization methods can be adopted. Here, we use Projected Gradient Descent (PGD) [Madry *et al.*, 2018] to optimize $\mathbf{X}_{fake}$.

Our method is outlined in Figure 2. With the resultant Adversarial Vulnerability, we solve the optimization of Eq. (14) by K-Means clustering. After that, the features of fake nodes, initialized as the cluster center in Eq. (15), are optimized using Eq. (16) and Eq. (17). More details of our algorithm are deferred to the appendix.

4.3 Theoretical Analysis

Connecting fake nodes to an original graph brings a victim node (1) one 1-hop neighbor (the fake node connected to it); (2) neighbors at a farther distance connected through this fake node; (3) fake nodes connected to other victim nodes which are at least 2-hop away. It is noted that 1-hop neighbors are often dominant. Here we leave out the influence of farther neighbors caused by the fake node and fake nodes connected to other victim nodes which are at least 2-hop or even farther. The loss function over the ith victim node $v_i \in \Phi_A$ in Eq. (5) can thus be seen as a function of fake nodes' features (here we set a trade-off parameter $\lambda = 0$ for analysis). We have

$$\mathcal{L}(G) = \sum_{v_i \in \Phi_A} l(G^+, v_i) = \sum_{i=1}^{|\Phi_A|} l_i(x_i), \tag{18}$$

where x_i denotes fake nodes' features connected to victim node v_i. Theoretically, we provide our bounds under certain smooth conditions which hold for numerous neural networks.

Definition 3 (W-condition)**.** *We say that a function $\mathcal{L}(G) = \sum_{i=1}^{|\Phi_A|} l_i(x_i)$ satisfies the W-condition, if and only if $\forall\, 1 \le i \le |\Phi_A|$, $l_i(\cdot)$ satisfies the Lipschitz condition of order 2. In this case, we have*

$$m_i ||x_i - x_i^*||_2^2 \le l_i(x_i) - l_i(x_i^*) \le M_i ||x_i - x_i^*||_2^2, \tag{19}$$

where $0 \le m_i \le M_i$ are constants and $1 \le i \le |\Phi_A|$. x_i^* is the minimum of $l_i(\cdot)$.

Note that M exists because the loss function satisfies the Lipschitz condition of order 2 under W-condition; and it also includes m because $m = 0$ always satisfies Eq. (19). Under W-condition, we derive our bounds on the difference in adversarial loss between our results and optimal adversarial examples.

Proposition 2. *If $\mathcal{L}(\cdot)$ in Eq. (18) satisfies the W-condition, G^m is the optimal choice of Eq. (2) and $G^{'}$ is the optimal given by the Cluster Attack of Eq. (14). Then, we have*

$$\mathcal{L}(G^{'}) - \mathcal{L}(G^m) \le |M - m| \min_{C} \sum_{C_i \in C} \sum_{v \in C_i} d(v, c_i), \tag{20}$$

where $M = \max_{1 \le i \le |\Phi_A|} M_i$ and $m = \min_{1 \le i \le |\Phi_A|} m_i$.

Proof. We have

$$\begin{aligned}
&\mathcal{L}(G^{'}) - \mathcal{L}(G^m) \\
=&\sum_{i=1}^{|\Phi_A|} l_i(x_i') - \sum_{i=1}^{|\Phi_A|} l_i(x_i^m) \\
=&\sum_{i=1}^{|\Phi_A|} (l_i(x_i') - l_i(x_i^*)) - \sum_{i=1}^{|\Phi_A|} (l_i(x_i^m) - l_i(x_i^*)) \\
\le&\sum_{i=1}^{|\Phi_A|} M_i ||x_i' - x_i^*||_2^2 - \sum_{i=1}^{|\Phi_A|} m_i ||x_i^m - x_i^*||_2^2 \\
\le&M \sum_{i=1}^{|\Phi_A|} ||x_i' - x_i^*||_2^2 - m \sum_{i=1}^{|\Phi_A|} ||x_i^m - x_i^*||_2^2 \\
=&M \min_{C} \sum_{C_i \in C} \sum_{v \in C_i} d(v, c_i) - m \sum_{i=1}^{|\Phi_A|} ||x_i^m - x_i^*||_2^2 \\
\le&M \min_{C} \sum_{C_i \in C} \sum_{v \in C_i} d(v, c_i) - m \min_{C} \sum_{C_i \in C} \sum_{v \in C_i} d(v, c_i) \\
=&|M - m| \min_{C} \sum_{C_i \in C} \sum_{v \in C_i} d(v, c_i),
\end{aligned} \tag{21}$$

Name	Nodes	Edges	Features	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3702	6
Reddit	232965	11606919	602	41
ogbn-arxiv	169343	1157799	128	40

Table 2: Statistics of the datasets.

where x_i^m and x_i' are features of the fake node connected to ith victim node provided by Eq. (2) and Cluster Attack of Eq. (14), respectively. □

Proposition 2 indicates that the difference in adversarial loss between our results and optimal adversarial examples is bounded by the minimal cluster distance in Eq. (14) and how each $l_i(\cdot)$ is linear to $||x_i - x_i^*||_2^2$.

5 Experiments

5.1 Experimental Setup

Dataset. We do our experiments on Cora and Citeseer [Sen *et al.*, 2008], which are two benchmark small citation networks with discrete node features, and on Reddit [Hamilton *et al.*, 2017] and ogbn-arxiv [Hu *et al.*, 2020], which are two large networks with continuous node features. The statistics of the datasets are shown in Table 2.

Parameters. For each experimental setting, we run the experiment for 100 times and report the average results. In each round, we randomly sample $|\Phi_A|$ nodes as victim nodes. We set $k = 1$ in $\mathcal{N}_k(\Phi_A)$, which means we aim to protect the 1-hop neighbors of victim nodes. Without specification, we compare our method with baselines with a trade-off parameter set as $\lambda = 0$ in Eq. (5).

Comparison Methods. Since this study is the first to perform query-based injection attack on node classification, most of the previous baselines on graph injection attacks cannot be easily adapted to our problem. We include the following baselines:

Random Attack, which decides the fake nodes' features and connections between fake nodes and original nodes randomly.

NETTACK, one of the most effective attacks by first adding several nodes and then adding many edges between the fake nodes and original nodes [Zügner *et al.*, 2018].

NETTACK - Sequential, which is a variant of NETTACK [Zügner *et al.*, 2018] by sequentially adding fake nodes.

Fake Node Attack, which adds fake nodes in a white-box attack scenario [Wang *et al.*, 2018].

G-NIA, a white-box graph injection attack [Tao *et al.*, 2021]. We refer to the reported results on Citeseer.

TDGIA, a black-box GIA method with superior performance to all the baselines in KDD Cup 2020[1] of graph injection attack. We mainly compare our method with this method. Note that TDGIA is not query-based [Zou *et al.*, 2021].

[1]https://www.kdd.org/kdd2020/kdd-cup

Among the above baselines, TDGIA is performed in a continuous feature space while NETTACK and Fake Node Attack are performed in discrete feature spaces.

5.2 Quantitative Evaluation

Without loss of generality, we uniformly set $N_{fake} = 4$ and let the number of victim nodes vary to see the performance under different $N_{fake} : |\Phi_A|$.

Performance on Small Datasets with Discrete Features

We first evaluate the performance of the Cluster Attack along with other baselines on Cora and Citeseer with discrete features. The number of queries K is set to $K = |\Phi_A| \cdot K_t + N_{fake} \cdot K_f$, where $K_t = K_f = D$ (feature dimension). The results are shown in Table 3. Our algorithm outperforms all baselines in terms of success rates. This is because our method prevents inefficiently searching the non-Euclidean space of the adjacent matrix and better utilizes the limited queries in searching the Euclidean feature space. The results also demonstrate that the Adversarial Vulnerability is a good metric for clustering the victim nodes.

Performance on Large Datasets with Continuous Features

In this section, we evaluate the performance of the Cluster Attack on Reddit (with $1500|\Phi_A| + 750N_{fake}$ queries) and obgn-arxiv (with $4000|\Phi_A| + 2000N_{fake}$ queries), two large networks with continuous feature whose victim nodes have a higher average degree. We compare our method with the state-of-the-art method which has superior performance to other baselines. In these challenging datasets, we perform an untargeted attack, which means attacker successes when the predicted labels of victim nodes are changed. The results are shown in Table 4. Our algorithm outperforms the baseline in terms of success rates. This is because, with the cluster prior of the adjacent matrix, our Cluster Attack prevents inefficiently searching in non-Euclidean space and make the best use of the limited queries to search the Euclidean feature space. Another reason is because of the good metric of the Adversarial Vulnerability, which provides an appropriate cluster prior.

5.3 Ablation Study

Performance with Different Trade-Off Parameters λ

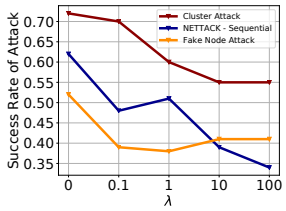
In this section, we examine the performance of Cluster Attack with different trade-off parameters λ between fake nodes and protected nodes in the Cora dataset. We uniformly set $N_{fake} = 4$, $|\Phi_A| = 10$. We choose two competitive baselines and adapt their loss functions to our trade-off format. The results are shown in Figure 3. It can be seen from Figure 3 that, when λ increases (which means that we pay more attention to the protected nodes), the percentage of protected nodes whose labels remain unchanged during the attack also increases. This is because we try to protect the labels of the protected nodes from being changed in a trade-off formulation in our loss function, Eq. (5). Also, our trade-off formulation can be generalized to other baselines, as shown in Figure 3. This is because we design our loss function in Eq. (5) in a generalizable manner independent of attack method.

Method	Cora				Citeseer			
	$T=3$	$T=5$	$T=7$	$T=10$	$T=3$	$T=5$	$T=7$	$T=10$
Random	0.07	0.08	0.04	0.05	0.04	0.02	0.03	0.03
NETTACK	0.61	0.57	0.55	0.53	0.75	0.71	0.66	0.61
NETTACK - Sequential	0.68	0.73	0.72	0.70	0.76	0.74	0.72	0.67
Fake Node Attack	0.61	0.58	0.54	0.52	0.76	0.68	0.62	0.60
G-NIA	-	-	-	-	0.86	0.76	0.70	0.65
Cluster Attack	**0.99**	**0.93**	**0.84**	**0.72**	**1.00**	**0.89**	**0.80**	**0.70**

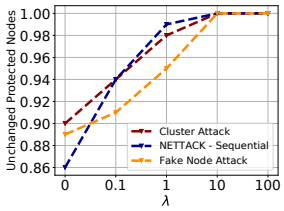
Table 3: Success rates of Cluster Attack along with other baselines with discrete feature space. T denotes number of victim nodes.

Method	ogbn-arxiv		Reddit	
	$T=12$	$T=16$	$T=12$	$T=16$
TDGIA	0.45	0.38	0.09	0.07
Cluster Attack	**0.67**	**0.59**	**0.15**	**0.12**

Table 4: Success rates of Cluster Attack along with other baseline with continuous features. T denotes number of victim nodes.



(a) Success rates of attack



(b) Percentage of unchanged protected nodes

Figure 3: Cluster Attack in Cora with different λ.

Performance with Different Number of Queries

In this section, we examine the performance of Cluster Attack with a different number of queries. We set $K_t = K_f = \alpha \cdot D$ and examine the performance under different α in Cora and Citeseer dataset. We uniformly set $N_{fake} = 4$, $|\Phi_A| = 10$ with $\lambda = 0$ and $\lambda = 1$. The results are shown in Figure 4. The success rate of Cluster Attack drops as the number of queries drops. Our algorithm still performs well when the number of queries drops slightly, especially when $\alpha \geq 0.4$. This demonstrates that our Cluster Attack can work in a query-efficient manner. This is because cluster algorithm provides graph-dependent priors for the adjacent matrix and thus prevents inefficient searching. Searching in the Euclidean feature space is more efficient.

We provide additional experiments in the appendix. The experiments show that nodes with a lower degree are more likely to get misclassified under attack. Also, when the number of fake nodes increases, the success rate of the attack increases too, which is consistent with our intuitive understanding. We provide an ablation study on the cluster metric of Adversarial Vulnerability. We find that original Cluster Attack performs better than Cluster Attack without Adversarial Vulnerability, i.e., the victim nodes' Adversarial Vulnerabilities are randomly

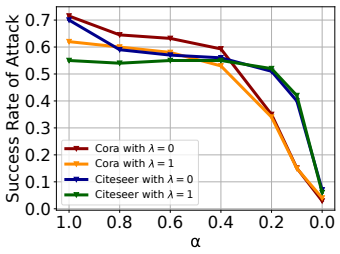


Figure 4: Success rates of Cluster Attack with different number of queries in Cora and Citeseer.

set. This result demonstrates the effectiveness of our Adversarial Vulnerability.

6 Conclusion

In this paper, we provide a unified framework for query-based adversarial attacks on graphs. Under the framework, we propose Cluster Attack, a query-based black-box graph injection attack with partial information. We demonstrate that a graph injection attack can be formulated as an equivalent clustering problem. The difficult discrete optimization problem of the adjacent matrix can thus be solved in the context of clustering. After providing theoretical bounds on our method, we empirically show that our method has strong performance in terms of the success rate of attacking.

Ethical Statement

The safety and robustness of AI are attracting more and more attention. In this work, we propose a method of adversarial attack. We hope our work reveals the potential weakness of current graph neural networks to some extent, and more importantly inspires future work to develop more robust graph neural networks.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Nos. 2020AAA0104304, 2017YFA0700904), NSFC Projects (Nos. 62061136001, 61621136008, 62076147, U19B2034, U19A2081, U1811461), the major key project of PCL (No. PCL2021A12), Tsinghua-Alibaba Joint Research Program, Tsinghua-OPPO Joint Research Center, and the High Performance Computing Center, Tsinghua University.

References

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[Chen *et al.*, 2017] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 15–26, New York, NY, USA, 2017. Association for Computing Machinery.

[Chen *et al.*, 2019] Weijian Chen, Yulong Gu, Zhaochun Ren, Xiangnan He, Hongtao Xie, Tong Guo, Dawei Yin, and Yongdong Zhang. Semi-supervised user profiling with heterogeneous graph attention networks. In *IJCAI*, volume 19, pages 2116–2122, 2019.

[Dai *et al.*, 2018] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR, 2018.

[Dong *et al.*, 2018] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[Ilyas *et al.*, 2018] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[Ma *et al.*, 2021] Yao Ma, Suhang Wang, Tyler Derr, Lingfei Wu, and Jiliang Tang. Graph adversarial attack via rewiring. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1161–1169, New York, NY, USA, 2021. Association for Computing Machinery.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[Mu *et al.*, 2021] Jiaming Mu, Binghui Wang, Qi Li, Kun Sun, Mingwei Xu, and Zhuotao Liu. A hard label black-box adversarial attack against graph neural networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 108–125, New York, NY, USA, 2021. Association for Computing Machinery.

[Qiu *et al.*, 2018] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2110–2119, 2018.

[Schaeffer, 2007] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

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

[Tao *et al.*, 2021] Shuchang Tao, Qi Cao, Huawei Shen, Junjie Huang, Yunfan Wu, and Xueqi Cheng. Single node injection attack against graph neural networks. *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, Oct 2021.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR 18*, 2017.

[Wan *et al.*, 2021] Xingchen Wan, Henry Kenlay, Robin Ru, Arno Blaas, Michael A Osborne, and Xiaowen Dong. Adversarial attacks on graph classifiers via bayesian optimisation. In *Advances in Neural Information Processing Systems*, volume 34, pages 6983–6996, 2021.

[Wang *et al.*, 2018] Xiaoyun Wang, Minhao Cheng, Joe Eaton, Cho-Jui Hsieh, and Felix Wu. Attack graph convolutional networks by adding fake nodes. *arXiv preprint arXiv:1810.10751*, 2018.

[Yang and Long, 2021] Runze Yang and Teng Long. Derivative-free optimization adversarial attacks for graph convolutional networks. *PeerJ Computer Science*, 7:e693, 2021.

[Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.

[Zou *et al.*, 2021] Xu Zou, Qinkai Zheng, Yuxiao Dong, Xinyu Guan, Evgeny Kharlamov, Jialiang Lu, and Jie Tang. Tdgia: Effective injection attacks on graph neural networks. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug 2021.

[Zügner *et al.*, 2018] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, 2018.