Adversarial Attack Framework on Graph Embedding Models with Limited Knowledge

Heng Chang, Yu Rong, Tingyang Xu, Wenbing Huang, Honglei Zhang, Peng Cui, Xin Wang, Wenwu Zhu, *Fellow, IEEE*, and Junzhou Huang, *Fellow, IEEE*

Abstract—With the success of the graph embedding model in both academic and industrial areas, the robustness of graph embeddings against adversarial attacks inevitably becomes a crucial problem in graph learning. Existing works usually perform the attack in a white-box fashion: they need to access the predictions/labels to construct their adversarial losses. However, the inaccessibility of predictions/labels makes the white-box attack impractical for a real graph learning system. This paper promotes current frameworks in a more general and flexible sense – we consider the ability of various types of graph embedding models to remain resilient against black-box driven attacks. We investigate the theoretical connection between graph signal processing and graph embedding models, and formulate the graph embedding model as a general graph signal process with a corresponding graph filter. Therefore, we design a generalized adversarial attack framework: *GF-Attack*. Without accessing any labels and model predictions, *GF-Attack* can perform the attack directly on the graph filter in a black-box fashion. We further prove that *GF-Attack* can perform an effective attack without assumption on the number of layers/window-size of graph embedding models. To validate the generalization of *GF-Attack*, we construct *GF-Attack* on five popular graph embedding models. Extensive experiments validate the effectiveness of *GF-Attack* on several benchmark datasets.

Index Terms—Adversarial attack, deep graph learning, graph neural networks, graph representation learning.

1 Introduction

Graph Embedding Models (GEMs) [1]–[4], which elaborate the expressive power of deep learning on graph-structured data, have achieved remarkable success in various domains, such as drug discovery [5]–[7], social network analysis [8]–[10], computer version [11], [12], medical imaging [12], [13], financial surveillance [14], structural role classification [15], [16] and automated machine learning [17]. Given the increasing popularity and success of these methods, several recent papers have investigated the risk of GEMs against adversarial attacks, as other researchers had examined for convolutional neural networks [18]. The papers [19]–[21] have already shown that various kinds of graph embedding methods, including GCN [22], DeepWalk [23], *etc*., are vulnerable to adversarial attacks.

Undoubtedly, the potential attack risk is rising for modern graph learning systems. For instance, by sophisticated constructed social bots and following connections, it's possible to fool the recommendation system equipped with GEMs to give wrong recommendations. Another example is from the credit prediction model. The model tends to suppose that users connecting with high-credit users also have high credits. By constructing fake connections with high-credit users, fraudsters can easily fool the credit prediction model and lead to severe consequences. This potential risk calls for the attention on strengthening the security of GEMs. In this vein, the need for a new adversarial attack framework is especially essential for a better understanding of the adversarial examples existing in graphs as well as the design of more robust GEMs.

Regarding the amount of information from both the target model and data required for the generation of adversarial examples, all graph adversarial attackers fall into three categories (arranged in ascending order of difficulties):

- White-box Attack (**WBA**): the attacker can access any information, namely, input data (e.g., adjacency matrix and feature matrix), labels, gradients, model parameters, model predictions, etc. However, this situation could be impractical since such information usually is well protected or inaccessible in the real world.
- Practical White-box Attack (**PWA**) (or Grey-box Attack): the attacker can access any information except the model gradients and parameters. Still, such information of GEMs is also difficult for attackers to obtain. For example, users in the credit prediction model are usually encoded to be anonymous and the labels of users are hard to be reached.
- Restricted Black-box Attack (**RBA**): the attacker can only access the adjacency matrix and attribute matrix. Access to parameters, labels, and predictions is prohibited. Being the most difficult but most practical setting, RBA is more natural in a real-world scenario, because the input data is always the only information we can easily obtain in most situations.

Table 1 summarizes the information accessibility under

- *H. Chang is with the Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Beijing, China. E-mail: changh17@mails.tsinghua.edu.cn*
- *Y. Rong, T. Xu and H. Zhang are with the Tencent AI Lab, Tencent, Shenzhen, China. E-mail: yu.rong@hotmail.com, Tingyangxu@tencent.com*
- *W. Huang is with Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China. E-mail: hwenbing@126.com.*
- *P. Cui, X. Wang and W. Zhu are with the Department of Computer Science and Technology, Tsinghua University. E-mail: cuip@tsinghua.edu.cn, xin_wang@tsinghua.edu.cn, wwzhu@tsinghua.edu.cn*
- *J. Huang is with the Department of Computer Science and Engineering, University of Texas at Arlington. E-mail: jzhuang@uta.edu*
- *Yu Rong and Wenwu Zhu are the corresponding authors.*

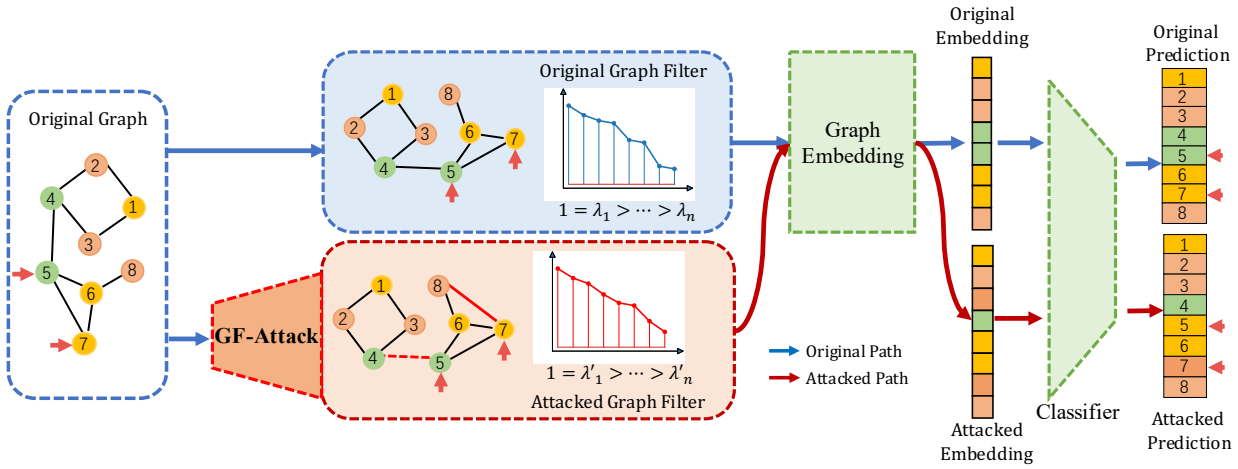


Fig. 1. The overview of the whole attack procedure of *GF-Attack*. Given target vertices 5 and 7, *GF-Attack* aims to misclassify them by attacking the graph filter and producing adversarial edges (edge e_{45} deleted and edge e_{78} added) on the graph structure. The common graph embedding block refers to the general target GEM and can be any kind of the potential GEMs, illustrating the flexibility and extensibility of GF-Attack. In this vein, *GF-Attack* would not change the target embedding model.

TABLE 1
Summary of information accessibility under three attack settings.

Settings	Parameters	Predictions	Labels	Input Data
WBA	✓	✓	✓	✓
PWA		✓	✓	✓
RBA				✓

different adversarial attack settings. Despite the fruitful results [20], [24], [25] which absorb ingredients from exiting adversarial methods on convolutional neural networks, obtained in attacking graph embeddings under both WBA and PWA setting, however, the target model parameters/gradients, the labels, and predictions are seldom accessible in real-life applications. In other words, it is almost impossible for the WBA and PWA attackers to perform a threatening attack on real systems. Meanwhile, current RBA attackers are either reinforcement learning-based [19], which has low computational efficiency or derived merely only from the structure information without considering the feature information [26]. Therefore, how to perform the effective adversarial attack toward GEM relying on the input adjacency matrix and attribute matrix, a.k.a., RBA setting, is still more challenging yet meaningful in practice.

The core task of the adversarial attack on the GEM is to damage the quality of output embeddings to harm the performance of downstream tasks within the manipulated features or graph structure, i.e., vertex or edge insertion/deletion. Namely, finding the embedding quality measure to evaluate the damage on graph embeddings is vital. For the WBA and PWA attackers, they have enough information to construct this quality measure, such as the loss function of the target model. In this vein, the attack can be performed by simply maximizing the loss function reversely given the known labels, either through gradient ascent [19] or a surrogate model [20], [25]. However, the RBA attacker cannot employ the limited information to recover the loss function of the target model. In a nutshell, the biggest challenge of the RBA attacker is: how to figure out the goal of the target model barely by the input data.

In this paper, we try to understand GEMs from a new perspective and propose an attack framework: *GF-Attack*, which can perform an adversarial attack on various kinds of GEMs. Specifically, we formulate a GEM as a general graph signal processing with a corresponding graph filter which can be computed by the input adjacency matrix. Therefore, we employ the graph filter as well as the corresponding feature matrix to construct the embedding quality measure as a T-rank approximation problem. In this vein, instead of attacking the loss function, we aim to directly attack the graph filter of given GEMs without knowing the labels and predictions. Therefore, *GF-Attack* can perform an attack in a restricted black-box fashion by only assuming what type is the victim model. Furthermore, through evaluating this T-rank approximation problem, *GF-Attack* is capable of performing the adversarial attack on any GEM that can be formulated as a general graph signal processing. Figure 1 provides an overview of the whole attack procedure of *GF-Attack*. Moreover, by theoretically analyzing the alternate adversarial loss on graph filter, we show that when we construct the attack loss in *GF-Attack* with higher-order polynomial, the generated adversarial edges could perform more effective attacks on GEMs with a smaller number of layers/window-size. To demonstrate the effectiveness of *GF-Attack* attacking various kinds of GEMs, we give the quality measure construction for four popular GEMs (GCN, SGC, DeepWalk, LINE). Empirical results show that our general attack method is capable of effectively performing adversarial attacks on popular unsupervised/semi-supervised GEMs on real-world datasets in a restricted black-box fashion.

The primary version has been published in the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20) [27]. The contributions of the conference version are summarized as follows:

- We construct the theoretical connection between GEM and graph signal processing with the corresponding graph filters.
- We formulate the embedding quality measure as a T-rank approximation problem via graph filters, and the

RBA setting is satisfied in this way. A general attack framework *GF-Attack* is proposed accordingly.
- Experiments towards attacking four popular GEMs on real-world datasets reveal the effectiveness of the proposed framework *GF-Attack*.

In the conference version [27], *GF-Attack* performs attack with the additional assumption on the number of layers/window-size in GEMs. In this extended version, we further analyze the generalization ability of *GF-Attack* on attacking GEMs with different layers/window-size to further remove the dependency on this assumption, especially from the perspective of theoretical findings. We list the key additional contributions here, independently:
- By investigating the adversarial loss of *GF-Attack*, we prove that *GF-Attack* can perform the effective attack without additional assumption on the number of layers/window-size of GCNs and sampling-based GEMs, which is an important step further to a more ideal black-box attack setting.
- A parameterized-filter variant of GCNs, ChebyNet, is included as victim model in experiments to further demonstrate the attack ability of *GF-Attack* aside from fixed-filter GCNs.
- We adopt a more black-box setting, *i.e.* using the same attack loss for all victim models, to further empirically validate the effectiveness of *GF-Attack* under both poisoning and evasion settings.
- Ablation studies on more benchmarks focusing on computational efficiency and multi-edge attack complete the empirical results. The additional results further demonstrate that *GF-Attack* enjoys both effectiveness and efficiency on all benchmarks.

2 Related work

Graph Learning and Graph Embedding Models. Graph embedding models (GEMs) [1], [28], [29] are essential techniques for graph analytic tasks. A taxonomy of GEMs can be broadly divided into four kinds [28]: (i) factorization methods, (ii) random walk (sampling-based) techniques, (iii) deep learning, and (iv) other miscellaneous strategies. Among them, random walk and deep learning-based methods are the most representative categories. For random walk techniques, DeepWalk [23], LINE [30] and node2vec [31] adopt SkipGram, a neural language model that aims to maximize the co-occurrence probability among the words that appear within a window, for graph embeddings. These methods then preserve different orders of network proximity with the learned low-dimensional vectors. We denote the methods from this category as sampling-based GEMs. As for deep learning, Graph Convolutional Networks (GCNs) such as GCN [22] and SGC [32] generalize the deep neural model to non-Euclidean domains and learn the low-dimensional graph embeddings to maintain different scales of structural similarity. Regarding to whether the graph filters in GCNs parameterized, we can category GCNs as fixed-filter (*e.g.*, GCN and SGC), and parameterized-filter (*e.g.*, ChebyNet [33], GAT [34] and GraphHeat [35]) variants. In this work, our theoretical analysis focuses on fixed-filter GCNs, and the empirical experiments are evaluated by viewing both types of variants as victim models. For an explanation of GEMs, Qiu *et al.* [36] shows some insights on the understanding of sampling-based GEMs. However, they focus on proposing new graph embedding methods rather than building up a theoretical connection.

Adversarial Attacks on Graphs. Recently, adversarial attacks on deep learning for graphs have drawn unprecedented attention from researchers. Dai *et al.* [19] exploits a reinforcement learning-based framework under the RBA setting. However, they restrict their attacks on edge deletions only for vertex classification. Even more, they do not evaluate the *transferability* [37], which denotes the phenomenon that the adversarial examples generated for a specific model can also be harmful when they are used on another model. Transferability is an important ability of adversarial examples. Zügner, Akbarnejad, and Günnemann [20] proposes attacks based on a surrogate model and they can do both edge insertion/deletion in contrast to Dai *et al.* [19]. But their method utilizes additional information from labels, which is under the PWA setting. Further, Zügner and Günnemann [25] utilizes meta-gradients to conduct attacks under black-box setting by assuming the attacker uses a surrogate model same as Zügner, Akbarnejad, and Günnemann [20]. Their performance highly depends on the assumption of the surrogate model, and also requires label information. Moreover, they focus on the global attack setting. Xu *et al.* [38] proposes a gradient-based method under the WBA setting and overcomes the difficulty brought by discrete graph data. In the meantime, Wu *et al.* [39] also suggests using the integrated gradients to search for edges and features as adversarial examples under the WBA setting.

Bojchevski and Günnemann [26] considers a different adversarial attack task on vertex embeddings under the RBA setting. Inspired by Qiu *et al.* [36], they maximize the loss obtained by DeepWalk with matrix perturbation theory while only considering the information from the adjacency matrix. Besides, several other works also open doors for interesting research directions in many ways. Li *et al.* [40] proposes an iterative learning framework to hide targeted individuals from the community detection task by GEMs in a black-box fashion. Entezari *et al.* [41] finds that only the high-rank singular components of the graph are affected by the attack method Nettack [20]. Then Entezari *et al.* [41] suggests that the power of Nettack can be greatly reduced if a low-rank approximation of the graph is utilized in contrast to the original clean graph. This finding is consistent with our analysis in measuring the embedding quality from Section 4 that we can optimize the low-rank approximation of the output embeddings reversely to generate adversarial edges. Meanwhile, Ma, Ding, and Mei [42] studies the problem of the black-box attacks on graph neural networks by enforcing a novel constraint. In Ma, Ding, and Mei [42], attackers can only have access to a subset of vertices. Meanwhile, only a small number of candidates can be selected as target vertices. At the same time, Vidanage *et al.* [43] and Zhang *et al.* [44] consider the adversarial attack on graph neural networks from a new perspective. They focus on perturbing the graph structure to degrade the quality of the task of deep graph matching. Some efforts [45]–[48] have also been paid on the defense against the adversarial attack on GEMs recently.

Remarkably, despite all the above-introduced works except Dai *et al.* [19] showing the existence of transferability

in GEMs by experiments, they all lack theoretical analysis on this implicit connection. In this work, for the first time, we theoretically connect different kinds of GEMs and propose a general optimization problem from parametric graph signal processing. An effective algorithm is developed afterwards under the RBA setting.

3 PRELIMINARIES

Let $G(\mathcal{V}, \mathcal{E})$ be an attributed graph, where $\mathcal{V}$ is a vertex set with size $n = |\mathcal{V}|$ and $\mathcal{E}$ is an edge set with $|\mathcal{E}|$ edges. Denote $A \in \{0,1\}^{n\times n}$ as an adjacency matrix and $X \in \mathbb{R}^{n\times l}$ as a feature matrix with dimension l. $D_{ii} = \sum_j A_{ij}$ refers to the degree matrix. $\text{vol}(G) = \sum_i \sum_j A_{ij} = \sum_i D_{ii}$ denotes the volume of G. For consistency, we denote the perturbed adjacency matrix as A' and the normalized adjacency matrix as $\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. Symmetric normalized Laplacian and random walk normalized Laplacian are referred as $L^{sym} = I_n - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ and $L^{rw} = I_n - D^{-1}A$, respectively. We also denote the attributed graph after attack as $G'(\mathcal{V}', \mathcal{E}')$, and the corresponding adjacency matrix as A'. The other notations of the perturbed graph are defined analogously.

To cope with the data with graph structure in ML tasks, GEMs aim to encode sufficient features in graphs. Concretely, given a graph G, the goal is to learn a mapping function $\mathscr{M} : (A, X) \to \mathbb{R}^{n\times d}$ on the graph that represent vertex into a d-dimensional vector space with the preservation of structural (A) and non-structural (X) properties. According to the demand of random walk paths (RWs), deep learning-based GEMs generally fall into two categories [49]: convolution-based Graph Neural Networks (GCNs), *e.g.* GCN [22], and sampling-based GEMs, *e.g.* DeepWalk [50].

Given a GEM $\mathscr{M}_\Theta$ parameterized by Θ and a graph $G(\mathcal{V}, \mathcal{E})$, the adversarial attack on graph aims to perturb the learned vertex representation $Z = \mathscr{M}_\Theta(A, X)$ to damage the performance of the downstream learning tasks. In a summary, three components in graphs can be attacked as targets:

- Attack on $\mathcal{V}$: Add/delete vertices in graphs. This operation may change the dimension of the adjacency matrix A.
- Attack on $\mathcal{E}$: Add/delete edges in graphs. This operation would lead to the changes of entries in the adjacency matrix A. This kind of attack is also known as *structural attack*.
- Attack on X: Modify the attributes attached on vertices.

In this paper, we mainly focus on studying the adversarial attacks on the graph structure, i.e., adding/deleting the edges in graphs, since attacking $\mathcal{E}$ is more practical than others in real applications [51].

Meanwhile, considering in which stage the adversarial attack happens, we can also category the attack that happens at the test time as *evasion* attack, and at the training time as *poisoning* attack [20]. In this work, we mainly focus on evasion attack, since it is more realistic in comparison to the accessibility to training data.

3.1 Graph Signal Filtering

Graph Signal Processing (GSP) extends the concepts in Discrete Signal Processing and focuses on the analysis and processing of the data points whose relations are modeled as graphs [52], [53]. Similar to DSP, these data points can be treated as *signals*. Thus the definition of *graph signal* is:

Definition 1 (graph signal)**.** Given a graph $G(\mathcal{V}, \mathcal{E})$, a *graph signal* $\mathbf{x}$ is a mapping from vertex set $\mathcal{V}$ to real numbers:

$$\begin{aligned} \mathbf{x} : &\mathcal{V} \to \mathbb{R}, \\ &v_i \mapsto x_i. \end{aligned} \tag{1}$$

In Definition 1, each signal $\mathbf{x}$ is isomorphic in G. We can rewrite it into a vector: $\mathbf{v} = [v_1, \ldots, v_n]$. In this sense, the feature matrix X can be treated as graph signals with l channels.

To understand the graph signal x, it's essential to consider the graph structure. In general, a graph filter $\mathscr{H}$ is a system that takes a graph signal $\mathbf{x}$ as input and produces a new signal as an output. Namely, $\mathscr{H}$ performs a signal transformation on the original graph signals. In traditional DSP, *shift filter (z-transform)* is a basic but non-trivial transformation which delays the signals in the time domain. Thus we can extend the definition of *shift filter* to graph signals:

Definition 2 (graph-shift filter)**.** Given a graph $G(\mathcal{V}, \mathcal{E})$, a graph-shift filter $S \in \mathbb{R}^{n\times n}$ is a matrix satisfying: $\forall i \neq j$ and $e_{ij} \notin \mathcal{E}$, $S_{ij} = 0$.

The graph-shift filter S reflects the locality property of graphs, i.e., it represents a linear transformation of the signals on one vertex and its neighbors. It's the basic building blocks to construct $\mathscr{H}$. Some common choices of S include the adjacency matrix A and the Laplacian $L = D - A$, where D is the degree matrix $D_{ii} = \sum_{j=1}^{n} A_{ij}$.

3.2 Adversarial Attack Definition

Formally, given a fixed budget β indicating that the attacker is only allowed to modify 2β entries in A (undirected), the adversarial attack on a GEM $\mathscr{M}$ can be formulated as [26]:

$$\begin{aligned} \arg\max_{A'} \ & \mathcal{L}_{\text{attack}}(A', X; \Theta) = \mathcal{L}_{\text{attack}}(Z) \\ \text{s.t. } & Z = \mathscr{M}(A', X; \Theta^*), \\ & \Theta^* = \arg\min_{\Theta} \mathcal{L}_{\text{model}}(A', X; \Theta), \\ & \|A' - A\|_0 = 2\beta, \end{aligned} \tag{2}$$

where $Z = \mathscr{M}(A', X; \Theta^*)$ is the embedding output of the model $\mathscr{M}$ with the optimal model parameters Θ^*. $\mathcal{L}_{\text{model}}(A', X; \Theta)$ is the loss function of the victim model minimized by Θ. $\mathcal{L}_{\text{attack}}(Z)$ is defined as the attack loss function measuring the damage on output embeddings. For the WBA setting, $\mathcal{L}_{\text{attack}}(Z)$ can be defined as the minimization of the target loss, i.e., $\mathcal{L}_{\text{attack}}(Z) = \inf_{\Theta} \mathcal{L}_{\text{model}}(A', X; \Theta)$. This is generally a bi-level optimization problem since we need to re-train the model during attack to keep Θ^* as optimal in Z. In this work, we consider the evasion attack scenario, where $\Theta^* = \arg\min_\Theta \mathcal{L}_{\text{model}}(A, X; \Theta)$ are learned on the clean graph and remains unchanged during attack. In this way, we can treat the model parameters Θ^* as constants, which eases the construction of the attack loss from $\mathcal{L}_{\text{attack}}(Z)$ to $\mathcal{L}_{\text{attack}}(A', X)$.

Theoretically analyzing poisoning attacks is usually harder since the subsequent learning of $\Theta^* = \arg\min_\Theta \mathcal{L}_{\text{model}}(A, X; \Theta)$ should be considered [20], therefore we choose to concentrate on evasion setting and leave

the analysis under poisoning setting as future work. Note that though our loss is designed under the evasion setting, our main experimental results are under both settings, which demonstrate that our proposed attack loss can effectively destroy the performance of GEMs in practice.

4 METHODOLOGIES

From the perspective of GSP, we can formulate the process of generating embeddings $Z = \mathscr{M}(A, X; \Theta^*)$ as a generalization of signal processing, according to the graph filtering together with feature transformation:

$$\begin{aligned} \text{graph filtering} :& \tilde{X} = \mathscr{H}(X) = h(S)X, \\ \text{feature transformation} :& Z = \sigma(\tilde{X}\Theta^*), \end{aligned} \tag{3}$$

where $\sigma(\cdot)$ denotes the activation function, and $\Theta \in \mathbb{R}^{l \times l'}$ denotes the transformation weights from l input channels to l' output channels. $\mathscr{H} = h(S)$ denotes a graph signal filter, where $S = f(A)$ is the graph-shift filter and a function of adjacency matrix A, where the function is decided by a specific GEM. $\mathscr{H}$ is usually constructed by a polynomial function $h(x) = \sum_{i=0}^{L} a_i x^i \in \mathbb{R}^{n \times n}$ with graph-shift filter S. Many GEMs, including GCN, Deepwalk, etc, can be formulated as Eq. (3) with different graph signal filter $\mathscr{H}$. Table 2 summarizes the graph filter of different GEMs. We can find that the formulation from the process in Eq. (3) is so general that we can have the following assumption on the victim model:

Assumption 1. For a given victim GEM $\mathscr{M}$, the output embedding of $\mathscr{M}$ is learned through the process of the generalization of GSP as analyzed in (3).

Under Assumption 1, since the model parameters Θ^* are kept as constant as discussed before, it's intuitively adequate to focus on attacking the process of graph filtering $\tilde{X} = \mathscr{H}(X) = h(S)X$ for most GEMs. As a result, we can directly damage the quality of the output embedding Z through attacking $\tilde{X}$ by destroying the graph signal filter $\mathscr{H} = h(S)$. In this way, the optimization problem under our setting will be collapsed to:

$$\begin{aligned} \arg\max_{A'} \ & \mathcal{L}_{\text{attack}}(A', X) = \mathcal{L}_{\text{attack}}(\tilde{X}') \\ & \text{s.t. } \tilde{X}' = h(S')X, S' = f(A'), \|A' - A\|_0 = 2\beta. \end{aligned} \tag{4}$$

We name this way of constructing attack loss $\mathcal{L}_{\text{attack}}$ targeting the graph signal filter in the victim GEM under Assumption 1 as a general framework, *Graph Filter Attack (GF-Attack)*. Since the attack loss $\mathcal{L}_{\text{attack}}$ in *GF-Attack* does not involve the model parameters Θ and predictions, *GF-Attack* is a RBA framework for generating adversarial examples as discussed in the Introduction.

4.1 Embedding Quality Measure $\mathcal{L}_{\text{attack}}$ of *GF-Attack*

Now that we have the formulation (4) of the optimization problem under our general framework *GF-Attack*, the next step is to design an effective measure for evaluating the quality of the output embeddings. Recent works [54], [55] demonstrate that the output embeddings of GEMs can have a very low rank. Therefore, we establish the general measure of embedding quality in (4) accordingly as a T-rank approximation problem [36]:

$$\mathcal{L}_{\text{attack}}(A', X) = \|\tilde{X}' - \tilde{X}'_T\|_F^2 = \|h(S')X - h(S')_T X\|_F^2,$$

where $h(S')$ is the polynomial graph filter, S' is the graph shift filter constructed from the perturbed adjacency matrix A'. $h(S')_T$ is the T-rank approximation of $h(S')$. According to the low-rank approximation, $\mathcal{L}_{\text{attack}}(A', X)$ can be rewritten as:

$$\begin{aligned} \mathcal{L}_{\text{attack}}(A', X) &= \| \sum_{i=T+1}^{n} \lambda'_i \mathbf{u}_i \mathbf{u}_i^T X\|_F^2 \qquad (5) \\ &\leq \sum_{i=T+1}^{n} \|\lambda'_i\|_F^2 \|\mathbf{u}_i\|_F^2 \|\mathbf{u}_i^T X\|_F^2 \\ &\leq \sum_{i=T+1}^{n} {\lambda'_i}^2 \cdot \sum_{i=T+1}^{n} \|\mathbf{u}_i^T X\|_2^2, \qquad (6) \end{aligned}$$

where n is the number of vertices. $h(S) = U\Lambda U^{\mathrm{T}}$ is the eigen-decomposition of the graph filter $h(S)$. $h(S)$ is a symmetric matrix. $\Lambda = diag(\lambda_1, \cdots, \lambda_n)$, $U = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$ are the eigenvalue and eigenvector of graph filter $\mathscr{H}$, respectively, in order of $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. λ'_i is the corresponding eigenvalue after perturbation.

As the output embedding of a well-learned GEM has the desired low-rank property, we can view the training process of GEM as implicitly minimizing the attack loss $\mathcal{L}_{\text{attack}}$. On the opposite, for the attack purpose, we need to maximize $\mathcal{L}_{\text{attack}}$ for generating effective adversarial edges. While $\| \sum_{i=T+1}^{n} \lambda'_i \mathbf{u}_i \mathbf{u}_i^T X\|_F^2$ in Eq. (5) is hard to optimize, we can find its upper bound as in Eq. (6). Then during the generation of graph embeddings, the minimizing of this upper bound will be induced when the GEMs minimize the attack loss. Accordingly, the goal of adversarial attack can be maximizing the upper bound of the loss reversely, since (5) and (6) generally have the same monotonicity w.r.t. λ'_i as we show in the following Theorem 1:

Theorem 1. *When all λ'_i for $i \in [T+1, n]$ have the same signs,* (5) *and* (6) *are monotonically related w.r.t. λ'.*

Proof. We denote $f(\lambda'_i) = \| \sum_{i=T+1}^{n} \lambda'_i \mathbf{u}_i \mathbf{u}_i^T X\|_F^2$ and $g(\lambda'_i) = \sum_{i=T+1}^{n} {\lambda'_i}^2 \cdot \sum_{i=T+1}^{n} \|\mathbf{u}_i^T X\|_2^2$. Then for all non-negative λ'_i, it is easy to check that both $f(\lambda'_i)$ and $g(\lambda'_i)$ are non-decreasing w.r.t. λ'_i. Then for any pair of values, ${\lambda'_i}^{(a)}$ and ${\lambda'_i}^{(b)}$, if $f({\lambda'_i}^{(a)}) \leq f({\lambda'_i}^{(b)})$ holds then $g({\lambda'_i}^{(a)}) \leq g({\lambda'_i}^{(b)})$ also holds. By the definition, we can have that the two functions $f(\lambda'_i)$ and $g(\lambda'_i)$ are monotonically related. It is trivial to extend the same monotonicity for all non-positive λ'_i, which concludes the proof. □

For both GCNs and sampling-based GEMs that are chosen as examples in this work, all λ'_i for $i \in [T+1, n]$ can be chosen to have the same signs with a proper T, which reveals that Theorem 1 generally holds in our framework. Thus the restricted black-box adversarial attack loss (4) under *GF-Attack* framework is equivalent to optimize:

$$\arg\max_{\lambda'_i} \sum_{i=T+1}^{n} {\lambda'_i}^2 \cdot \sum_{i=T+1}^{n} \|\mathbf{u}_i^T X\|_2^2. \tag{7}$$

TABLE 2
From the perspective of general graph signal processing, we can formulate the GEMs with corresponding graph filters.

Graph Embedding Models	Graph-shift filter S	Polynomial Function $h(x)$	Input Signal	Parameters Θ
GCN [22]	$L^{sym} - I_n$	$h(x) = x$	X	Any
SGC [32]	$L^{sym} - I_n$	$h(x) = x$	X	Any
ChebyNet [33]	$L^{sym} - I_n$	$h(x) = \sum_{k=0}^{K} T_k(x)$	X	Any
LINE [30]	$I_n - L^{rw}$	$h(x) = x$	$\frac{1}{b} I_n$	$vol(G)D^{-1}$
DeepWalk [50]	$I_n - L^{rw}$	$h(x) = \sum_{k=0}^{K} x^k$	$\frac{1}{b} I_n$	$vol(G)D^{-1}$

According to (7), we can attack any GEM that can be described by the corresponding graph filter $\mathscr{H}$. Meanwhile, our general attack framework also provides a view of theoretical explanation on the transferability of adversarial examples created by [20], [25], [26], since modifying edges in adjacency matrix A implicitly perturbs the eigenvalues of graph filters. In the following, we will analyze two kinds of popular GEMs and aim to construct the corresponding adversarial attack losses under *GF-Attack* according to (7).

4.2 GF-Attack on Graph Convolutional Networks (GCNs)

4.2.1 Formulation of GCNs with the corresponding graph filter $\mathscr{H}$

Graph Convolution Networks (GCNs) extend the definition of convolution to the irregular graph structure and learn a representation vector of a vertex with feature matrix X. Namely, the Fourier transform is generalized on graphs to define the convolution operation: $g_\theta * \mathbf{x} = U g_\theta(\Lambda) U^T \mathbf{x}$. To accelerate the calculation, ChebyNet [33] proposes a polynomial filter $g_\theta(\Lambda) = \sum_{k=0}^{K} \theta_k \Lambda^k$ and approximates $g_\theta(\Lambda)$ by a truncated expansion concerning the Chebyshev polynomials $T_k(x)$:

$$g_{\theta'} * \mathbf{x} \approx \sum_{k=0}^{K} \theta'_k T_k(\widetilde{L})\mathbf{x}, \tag{8}$$

where $\widetilde{L} = \frac{2}{\lambda_{\max}} L - I_n$ and $\lambda_{\max}$ is the largest eigenvalue of Laplacian matrix L. $\theta' \in \mathbb{R}^K$ is now the parameters of Chebyshev polynomials $T_k(x)$. K denotes the K_{th} order Chebyshev polynomial. Due to the natural connection between Fourier transform and signal processing, it's easy to formulate the loss for ChebyNet under *GF-Attack*:

Lemma 1. *The K-localized single-layer ChebyNet with activation function $\sigma(\cdot)$ and weight matrix Θ is equivalent to filter graph signal X with a polynomial filter $\mathscr{H} = \sum_{k=0}^{K} T_k(S)$ with graph-shift filter $S = 2\frac{L^{sym}}{\lambda_{max}} - I_n$. $T_k(S)$ represents the Chebyshev polynomial of order k. Eq.* (3) *can be rewritten as:*

$$\tilde{X} = \sum_{k=0}^{K} T_k(2\frac{L^{sym}}{\lambda_{max}} - I_n)X, \quad X' = \sigma(\tilde{X}\Theta).$$

Proof. The K-localized single-layer ChebyNet with activation function $\sigma(\cdot)$ is $\sigma(\sum_{k=0}^{K} \theta'_k T_k(2\frac{L^{sym}}{\lambda_{max}} - I_n)X)$. Thus, we can directly write the graph-shift filter as $S = 2\frac{L^{sym}}{\lambda_{max}} - I_n$, and write the linear and shift-invariant filter as $\mathscr{H} = \sum_{k=0}^{K} T_k(S)$. □

GCN [22] constructs the layer-wise model by simplifying the ChebyNet with $K = 1$, $\theta'_0 = 1$ and $\theta'_1 = -1$. Then the *re-normalization trick* is used to avoid gradient exploding/vanishing:

$$X^{(l+1)} \quad = \quad \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^{(l)} \Theta^{(l)}\right), \tag{9}$$

where $\tilde{A} = A + I_n$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $\Theta^{(l)}$ are the parameters in the l_{th} layer and $\sigma(\cdot)$ is an activation function.

SGC [32] further utilizes a single linear transformation to achieve computationally efficient graph convolution, i.e., $\sigma(\cdot)$ in SGC is a linear activation function. We can formulate the multi-layer SGC in the sense of generalization of GSP, in favor of the construction of the corresponding attack loss under GF-Attack, through its theoretical connection to ChebyNet:

Corollary 1. The K-layer SGC is equivalent to the K-localized single-layer ChebyNet with K_{th} order polynomials of the graph-shift filter $S^{sym} = 2I_n - L^{sym}$. Eq. (3) can be rewritten as:

$$\tilde{X} = (2I_n - L^{sym})^K X, \quad X' = \sigma(\tilde{X}\Theta).$$

Proof. We can write the K-layer SGC as $(2I_n - L^{sym})^K X\Theta$. Since Θ are the learned parameters in the neural network, we can employ the reparameterization trick to use $(2I_n - L^{sym})^K$ to approximate the same order polynomials $\sum_{k=0}^{K} T_k(2I_n - L^{sym})$ with a new $\widetilde{\Theta}$. Then we rewrite the K-layer SGC by polynomial expansion as $\sum_{k=0}^{K} T_k(2I_n - L_{sym})X\widetilde{\Theta}$. Therefore, we can directly write the graph-shift filter $S^{sym} = 2I_n - L^{sym}$ with the same linear and shift-invariant filter $\mathscr{H}$ as K-localized single-layer ChebyNet. □

Note that SGC and GCN are identical when $K = 1$. Even though the non-linearity disturbs the explicit expression of the graph-shift filter of multi-layer GCN, the spectral analysis from [32] demonstrates that both GCN and SGC share similar graph filtering behavior. Thus, we extend the general attack loss from multi-layer SGC to multi-layer GCN under the non-linear activation function scenario. Our experiments confirm that the attack loss for multi-layer SGC also shows excellent performance on multi-layer GCN.

4.2.2 GF-Attack loss for SGC/GCN

As stated in Corollary 1, the graph-shift filter S of SGC/GCN is defined as $S^{sym} = 2I_n - L^{sym} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} + I_n = \hat{A} + I_n$, where $\hat{A}$ denotes the normalized adjacency matrix. Thus, for K-layer SGC/GCN, we can decompose the graph filter $\mathscr{H}$ as $\mathscr{H}^{sym} = (S^{sym})^K = U_{\hat{A}}(\Lambda_{\hat{A}} + I_n)^K U_{\hat{A}}^T$, where $\Lambda_{\hat{A}}$ and $U_{\hat{A}}$ are the eigen-pairs of $\hat{A}$. The corresponding adversarial attack loss for K_{th} order SGC/GCN can be written as:

$$\arg\max_{A'} \sum_{i=T+1}^{n} (\lambda'_{\hat{A}',i} + 1)^{2K} \cdot \sum_{i=T+1}^{n} \|\mathbf{u}_{\hat{A},i}^T X\|_2^2, \tag{10}$$

where $\lambda'_{\hat{A}',i}$ refers to the i_{th} largest eigenvalue of the perturbed normalized adjacency matrix $\hat{A}'$.

Directly calculating $\lambda'_{\hat{A}',i}$ from attacked normalized adjacency matrix A' will need an eigen-decomposition operation, which is extremely time consuming. Therefore, we introduce the eigenvalue perturbation theory [56] to fast estimate $\lambda'_{\hat{A}',i}$ in a linear time:

Lemma 2. *Let $A' = A + \Delta A$ be a perturbed version of A by adding/removing edges and ΔD be the respective change in the degree matrix. $\lambda_{\hat{A},i}$ and $\mathbf{u}_{\hat{A},i}$ are the i_{th} eigen-pair of eigenvalue and eigenvector of $\hat{A}$ and also solve the generalized eigen-problem $A\mathbf{u}_{\hat{A},i} = \lambda_{\hat{A},i} D\mathbf{u}_{\hat{A},i}$. Then the perturbed generalized eigenvalue $\lambda'_{\hat{A},i}$ is approximately as:*

$$\lambda'_{\hat{A}',i} \approx \lambda_{\hat{A},i} + \frac{\mathbf{u}^T_{\hat{A},i}\Delta A\mathbf{u}_{\hat{A},i} - \lambda_{\hat{A},i}\mathbf{u}^T_{\hat{A},i}\Delta D u_{\hat{A},i}}{\mathbf{u}^T_{\hat{A},i} D\mathbf{u}_{\hat{A},i}}. \quad (11)$$

Proof. Please kindly refer to [57]. □

Remark. With Theorem 2, we can directly derive the explicit formulation (Eq. (11)) of $\lambda'_{\hat{A}'}$ perturbed by ΔA on the original adjacency matrix A.

Order K irrelevant loss for SGC/GCN. As shown in (10), *GF-Attack* should know (or assume) the order K to perform the attack on the victim model. To further relax this constraint and make our framework for adversarial attack adapted to stricter RBA settings, we investigate the formulation of Eq. (10) without the impact from order K.

Since our aim is finding the proper $\lambda'_{\hat{A}',i}$ to maximize the loss, thus we can find the lower bound of Eq. (10) and maximize the lower bound correspondingly. Thus, the information from order K can be omitted properly. Following this approach, We figure out the relationship between the order K and the lower bound of Eq. (10):

Theorem 2. *The eigenvalues of $\hat{A}'$ are denoted as $1 \geq \lambda'_{\hat{A}',1} \geq \lambda'_{\hat{A}',2} \geq \cdots \geq \lambda'_{\hat{A}',n} \geq -1$. Suppose a large enough T is chosen to ensure the smallest $n - T$ eigenvalues, the optimization variables of Eq.* (10)*, all negative from $[-1, 0)$, then Eq.* (10) *is a monotonically decreasing function of K, and the corresponding adversarial attack loss for K_{th} order SGC/GCN is the lower bound for losses with orders less than K.*

Proof. Since K is irrelevant to the eigenvector part $\sum_{i=T+1}^{n} \|\mathbf{u}^T_{\hat{A},i} X\|_2^2$, our aim is to find the lower bound of $f(K) = \sum_{i=T+1}^{n} (\lambda'_{\hat{A}',i} + 1)^{2K}$. Taking the derivative of $f(x)$ directly, we can have $f'(K) = \sum_{i=T+1}^{n} (\lambda'_{\hat{A}',i} + 1)^{2K} \cdot 2\ln(\lambda'_{\hat{A}',i} + 1)$. As we ensure that the choice of T is large enough to make $\lambda'_{\hat{A}',i} \in [-1, 0)$, thus $2\ln(\lambda'_{\hat{A}',i} + 1) < 0$ and $f'(K) < 0$. This makes the attack loss function (10) for K_{th} order SGC/GCN a monotonically decreasing function of K, which indicates that it is the lower bound for the losses with orders less than K. □

Remark. From Theorem 2, instead of knowing the number of the layer K, we can conduct effective attacks for the target SGC/GCN models by optimizing the lower bound of the adversarial attack loss function (10). Therefore, we can choose a relatively large K in the loss function (10) for SGC/GCN to perform effective attacks in practice.

4.3 GF-Attack on Sampling-based GEMs

4.3.1 Formulation of Sampling-based GEMs with the corresponding graph filter $\mathscr{H}$

Sampling-based GEMs learns vertex representations according to the sampled vertices [31], vertex sequences [58], or network motifs [59]. For instance, LINE [30] with the second order proximity intends to learn two graph representation matrices X', Y' by maximizing the NEG loss of the skip-gram model:

$$\mathcal{L} = \sum_{i=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{V}|} A_{i,j}\Big(\log\sigma(x_i'^T y'_j) + b\mathbb{E}_{j'\sim P_N}[\log\sigma(-x_i'^T y'_j)]\Big), \quad (12)$$

where x'_i, y'_i are rows of X', Y', respectively. σ is the activation function and chosen as sigmoid here. b is the negative sampling parameter. P_N denotes the noise distribution generating negative samples. Meanwhile, DeepWalk [50] adopts the similar loss function except that $A_{i,j}$ is replaced with an indicator function indicating whether vertices v_i and v_j are sampled in the same sequence within the given context window-size K. Most of sampling-based GEMs only consider the structural information and ignore the feature matrix X. The output representation matrix is purely learned from the graph topology.

From the perspective of sampling-based GEMs, the embedded matrix is obtained by generating a training corpus for the skip-gram model from an adjacency matrix or a set of random walks. Qiu *et al.* [36] shows that Point-wise Mutual Information (PMI) matrices are implicitly factorized in the sampling-based embedding approaches. It indicates that LINE/DeepWalk can be rewritten into a matrix factorization form:

Lemma 3. *[36] Given the context window-size K and the number of negative sample b, the result of DeepWalk in matrix form is equivalent to factorize the matrix:*

$$M = \log\Big(\frac{vol(G)}{bK}(\sum_{k=1}^{K}(D^{-1}A)^k)D^{-1}\Big), \quad (13)$$

where $vol(G) = \sum_i\sum_j A_{ij} = \sum_i D_{ii}$ denotes the volume of graph G. And LINE can be viewed as a special case of DeepWalk with $K = 1$.

For the proof of Lemma 3, please kindly refer to [36]. Inspired by this insight, we prove that LINE can be viewed from a GSP manner as well:

Theorem 3. *LINE is equivalent to filter a graph signal $X = \frac{1}{b}I_n$ with a polynomial filter $\mathscr{H}$ and fixed parameters $\Theta = vol(G)D^{-1}$. $\mathscr{H} = S$ is constructed by graph-shift filter $S^{rw} = I_n - L^{rw}$. Eq.* (3) *can be rewritten as:*

$$\tilde{X} = \frac{1}{b}(I_n - L^{rw})D^{-1}I_n, \quad X' = log(vol(G)\tilde{X}).$$

Note that LINE is formulated from an optimized unsupervised NEG loss of a skip-gram model. Therefore, the parameter Θ and the value of the NCG loss are fixed with given graph signals.

We can extend Theorem 3 to DeepWalk since LINE can be viewed as a 1-window special case of DeepWalk:

Corollary 2. The output of K-window DeepWalk with b negative samples is equivalent to filtering a set of graph signals $X = \frac{1}{b} I_n$ with given parameters $\Theta = \text{vol}(G)D^{-1}$. Eq. (3) can be rewritten as:

$$\tilde{X} = \frac{1}{bK} \sum_{k=1}^{K} (I_n - L^{rw})^k D^{-1} I_n, \quad X' = log(\text{vol}(G)\tilde{X}).$$

Proof of Theorem 3 and Corollary 2. With Lemma 3, we can explicitly write DeepWalk as $\exp(M) = \frac{\text{vol}(G)}{b}(\sum_{k=1}^{K} \frac{1}{K}(I_n - L^{rw})^k D^{-1} I_n)$. Therefore, we can directly have the explicit expression of Eq. (3) on LINE/DeepWalk. □

As stated in Corollary 2, the graph-shift filter S of DeepWalk is defined as $S^{rw} = I_n - L^{rw} = D^{-1}A = D^{-\frac{1}{2}}\hat{A}D^{\frac{1}{2}}$. Therefore, the graph filter $\mathscr{H}$ of the K-window DeepWalk can be decomposed as $\mathscr{H}^{rw} = \frac{1}{K}\sum_{k=1}^{K}(S^{rw})^k$, which satisfies $\mathscr{H}^{rw}D^{-1} = D^{-\frac{1}{2}}U_{\hat{A}}(\frac{1}{K}\sum_{k=1}^{K}\Lambda_{\hat{A}}^k)U_{\hat{A}}^T D^{-\frac{1}{2}}$.

4.3.2 GF-Attack loss for LINE/DeepWalk

Since multiplying $D^{-\frac{1}{2}}$ in *GF-Attack* loss brings extra complexity, [36] provides us a way to well approximate the perturbed $\lambda'_{\mathscr{H}^{rw}D^{-1}}$ without this term:

Lemma 4. *[36] Let $\hat{A} = U\Lambda U^T$ and $\mathscr{H}_{rw} = \sum_{r=1}^{K} S_{rw}^r$ be the graph-shift filter of DeepWalk. The decreasing order s^{th} eigenvalue of $\mathscr{H}_{rw}$ are bounded as: $\lambda_{rw,s} \leq \frac{1}{d_{min}}|\frac{1}{K}\sum_{r=1}^{K}\lambda_{\pi_s}^r|$, where $\{\pi_1, \pi_2, \ldots, \pi_n\}$ is a permutation of $\{1, 2, \ldots, n\}$ ensuring the eigenvalue λ in the non-increasing order and d_{min} is the smallest degree in A. Then the smallest eigenvalue of $\mathscr{H}_{rw}$ is bounded as:*

$$\lambda_{min}(\mathscr{H}_{rw}) \geq \frac{1}{d_{min}} \lambda_{min}(U(\frac{1}{K}\sum_{r=1}^{K}\Lambda^r)U^T).$$

For the proof of Lemma 4, please kindly refer to [36]. Inspired by Lemma 4, we can find that both the magnitude of eigenvalues and smallest eigenvalue of $\mathscr{H}^{rw}D^{-1}$ are always well-bounded. Thus we have $\lambda'_{\mathscr{H}^{rw}D^{-1}} \approx \frac{1}{d_{\min}}\lambda'_{U_{\hat{A}}(\frac{1}{K}\sum_{k=1}^{K}\Lambda_{\hat{A}}^k)U_{\hat{A}}^T}$. Therefore, the corresponding adversarial attack loss of K_{th} order DeepWalk can be written as:

$$\arg\max_{A'} \sum_{i=T+1}^{n} (\frac{1}{d_{\min}}|\frac{1}{K}\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i}|)^2 \cdot \sum_{i=T+1}^{n} \|\mathbf{u}_{\hat{A},i}^T X\|_2^2. \quad (14)$$

Corollary 3. From Lemma 3, we can easily extend Eq. (14) for DeepWalk to LINE by setting $K = 1$, since LINE is a special case of DeepWalk with $K = 1$.

Similarly, Theorem 2 is utilized to estimate $\lambda'_{\hat{A}'}$ in the loss of LINE/DeepWalk.

Order K irrelevant loss for LINE/DeepWalk. Similar to the strategy we employ on the order K irrelevant adversarial attack loss (Eq. (10)) for GCNs, we can also relax the constraint of assuming the window-size K when performing the attack with loss Eq. (14). More specifically, the following Theorem 4 establishes the relationship:

Theorem 4. *Finding the lower bound for objective function* (14) *of order K is equivalent to find the lower bound for*

$$f(K) = \sum_{i=T+1}^{n} \frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2. \quad (15)$$

The smallest eigenvalue of $\hat{A}'$ other that -1 is denoted as $\lambda'_{\hat{A}',\min}$. Suppose a large enough T is chosen to make sure the smallest $n-T$ eigenvalues, the optimization variables of Eq. (14)*, all negative from $[-1, 0)$, then as long as K satisfies*

$$K \geq \sqrt{\frac{n-T}{\min f(K)}} \frac{1}{1+\lambda'_{\hat{A}',\min}}, \quad (16)$$

the corresponding attack loss with K_{th} order LINE/DeepWalk is the lower bound for losses with orders smaller than K, where $\min f(K)$ is the minimum of $f(K)$, and $\lambda'_{\hat{A}',\min}$ is the minimum eigenvalue in series $\lambda'_{\hat{A}',i}$ except -1s.

Proof. Since $\lambda'_{\hat{A}',i} \in [-1, 0)$, we can directly have Eq. (14) equal to $\sum_{i=T+1}^{n}\frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2 \cdot \frac{1}{d_{\min}^2} \cdot \sum_{i=T+1}^{n}\|\mathbf{u}_{\hat{A},i}^T X\|_2^2$. By eliminating the parts that irrelevant to order K, our aim is equivalent to finding the lower bound of $f(K) = \sum_{i=T+1}^{n}\frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2$. We conduct category discussion w.r.t. K here:

When K is even, *i.e.*, $K = 2z, z \in \mathbb{N}$, for the top m smallest $\lambda'_{\hat{A}',i} = -1$, we have $f(K)$ the following:

$$\begin{aligned} f(K) &= \sum_{i=T+1}^{n} \frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2 \quad (17)\\ &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2 + \sum_{i=n-m+1}^{n} \frac{1}{K^2}(\sum_{k=1}^{K}(-1)^k)^2 \\ &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2 + \sum_{i=n-m+1}^{n} \frac{1}{K^2} * 0 \\ &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2. \quad (18) \end{aligned}$$

Since $\lambda'_{\hat{A}',i} \neq -1$ in Eq. (18), we can have the following from Maclaurin Series:

$$\begin{aligned} f(K) &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K}\lambda'^k_{\hat{A}',i})^2 \\ &< \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K}(-\lambda'_{\hat{A}',i})^k)^2 \\ &< \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\frac{1}{1+\lambda'_{\hat{A}',i}})^2 \end{aligned}$$

For the minimum eigenvalue $\lambda'_{\hat{A}',\min}$ in series $\lambda'_{\hat{A}',i}$ except -1s, because $\frac{1}{1+\lambda'_{\hat{A}',i}} < \frac{1}{1+\lambda'_{\hat{A}',\min}}$, the following inequality holds

$$\begin{aligned} f(K) &< \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 \\ &= (n-m-T)\frac{1}{K^2}(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2. \end{aligned}$$

Assume $\min f(K)$ is the minimum of $f(K)$ for $K \in 1, 2, \ldots, K$, which can be easily obtained in practice, it follows that

$$\min f(K) \leq f(K) < (n-m-T)\frac{1}{K^2}(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2.$$

In order to find a K that satisfies the adversarial attack loss (14) for K_{th} order LINE/DeepWalk is the lower bound, we need to have

$$\min f(K) * K^2 \geq (n-m-T)(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2$$

Given $\min f(K) \geq 0$, which is obvious from Eq. (17) as It turns out that as long as K satisfies:

$$K \geq \sqrt{\frac{n-m-T}{\min f(K)}} \frac{1}{1+\lambda'_{\hat{A}',\min}}, \tag{19}$$

the adversarial attack loss (14) for K_{th} order LINE/DeepWalk is the lower bound for the losses with orders less than K.

When K is odd, *i.e.*, $K = 2z+1$, for the top m smallest $\lambda'_{\hat{A}',i} = -1$, we have $f(K)$ the following:

$$\begin{aligned} f(K) &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K} \lambda'^k_{\hat{A}',i})^2 + \sum_{i=n-m+1}^{n} \frac{1}{K^2}(\sum_{k=1}^{K}(-1)^k)^2 \\ &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K} \lambda'^k_{\hat{A}',i})^2 + \sum_{i=n-m+1}^{n} \frac{1}{K^2}(-1)^2 \\ &= \sum_{i=T+1}^{n-m} \frac{1}{K^2}(\sum_{k=1}^{K} \lambda'^k_{\hat{A}',i})^2 + \frac{m}{K^2}. \end{aligned} \tag{20}$$

Then we can have the similar result as the situation K is even from Maclaurin Series:

$$\begin{aligned} f(K) &< (n-m-T)\frac{1}{K^2}(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 + \frac{m}{K^2} \\ &= \left((n-T)(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 - m(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 + m\right)\frac{1}{K^2} \end{aligned}$$

Since $(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 > 1$, then $-m(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 < -m$, thus we have

$$\begin{aligned} f(K) &< \big((n-T)(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2 - m + m\big)\frac{1}{K^2} \\ &= (n-T)\frac{1}{K^2}(\frac{1}{1+\lambda'_{\hat{A}',\min}})^2. \end{aligned}$$

With the same analysis as K is even, we can have the desired condition for an odd K as

$$K \geq \sqrt{\frac{n-T}{\min f(K)}} \frac{1}{1+\lambda'_{\hat{A}',\min}}. \tag{21}$$

While $n-T \geq n-m-T$, combining the result from (19), we can have the overall desired condition for K is (21), which concludes the proof. □

Remark. Similar to *GF-Attack* on GCNs, by choosing a relatively large order K of the loss (14) for LINE/DeepWalk in practice, we can effectively attack the target LINE/DeepWalk model without the knowledge about the orders (window-size).

Algorithm 1 Algorithm in *GF-Attack* under the RBA setting

Input:
adjacency matrix A; feature matrix X; target vertex t; number of top-T smallest singular values/vectors selected T; order of graph filter K; fixed budget β.

Output:
Perturbed adjacency matrix A'.

Initial the candidate flips set as $\mathcal{C} = \{(v,t)|v \neq t\}$, eigenvalue decomposition of $\hat{A} = U_{\hat{A}}\Lambda_{\hat{A}}U^T_{\hat{A}}$;
for $(v,t) \in \mathcal{C}$ **do**
  Approximate $\Lambda'_{\hat{A}}$ resulting by removing/inserting edge (v,t) via Eq. (11);
  Update $Score_{(v,t)}$ from loss Eq. (10) or Eq. (14) w.r.t. the assumption on the type of victim model;
end for
$\mathcal{C}_{sel} \leftarrow$ edge flips with top-β $Score$;
$A' \leftarrow A \pm \mathcal{C}_{sel}$;
return A'

4.4 Attack Algorithm

Based on the general attack loss, the goal of our adversarial attack is to misclassify a target vertex t from an attributed graph $G(\mathcal{V},\mathcal{E})$ given a downstream vertex classification task. We start by defining the candidate flips then the general attack loss is responsible for scoring the candidates.

Directly solving Eq. (10) or Eq. (14) is complex and time-consuming. Though the search space for graph-structured data is discrete, searching over full graphs is still $\mathcal{O}(n^2)$, which also could be pretty complex for large graphs. To alleviate this issue, we first adopt the hierarchical strategy in [19] to decompose the single edge selection into two ends of this edge in practice. Then we let the candidate set $\mathcal{C}$ for edge selection contains all vertices (edges and non-edges) directly accessory to the target vertex, i.e. $\mathcal{C} = \{(v,t)|v \neq t\}$, which is consistent with [19], [26]. Intuitively, the further away the vertices from target t, the less influence they impose on t. Meanwhile, experiments in [20], [26] also show that their candidates from $\mathcal{C} = \{(v,t)|v \neq t\}$ can do significantly more damage compared to candidate flips chosen from the other parts of the graph. Thus, in our experiments we also choose to restrict our candidates within the same choice of set $\mathcal{C} = \{(v,t)|v \neq t\}$.

Overall, for a given target vertex t, we establish the targeted attack by sequentially calculating the corresponding *GF-Attack* loss w.r.t. graph-shift filter S for each flip in the candidate set as scores. Then with a fixed budget β, the adversarial attack is accomplished by selecting flips with top-β scores as perturbations on the adjacency matrix A of the clean graph. Details of the algorithm in *GF-Attack* under RBA setting are depicted in Algorithm 1.

5 Experiments

Datasets. We evaluate our approach on three real-world datasets: Cora, Citeseer, and Pubmed. In all three citation network datasets, vertices are documents with corresponding bag-of-words features and edges are citation links. The data preprocessing settings follow the benchmark setup in [22]. Only the largest connected component (LCC) is considered to

TABLE 3
Summary of the change in classification accuracy (in percent) compared to the clean/original graph. Single edge perturbation under the RBA and poisoning settings. Lower is better. We use *Cheby* for ChebyNet and *DW* for DeepWalk here to save space.

Dataset	Cora					Citeseer					Pubmed				
Models	GCN	SGC	Cheby	DW	LINE	GCN	SGC	Cheby	DW	LINE	GCN	SGC	Cheby	DW	LINE
(unattacked)	80.20	78.82	80.33	77.23	76.75	72.50	69.68	70.96	69.74	65.15	80.40	80.21	79.45	78.69	72.12
Random	-1.81	-2.01	-2.30	-1.84	-2.61	-1.57	-1.74	-1.92	-1.44	-1.13	-2.01	-2.18	-1.28	-1.95	-1.34
Degree	-3.50	-6.06	-5.59	-2.91	-4.59	-4.17	-4.24	-4.14	-7.55	-8.35	-3.20	-3.91	-3.68	-2.28	-8.41
RL-S2V	-4.10	-5.12	-6.48	-4.52	-5.39	-4.05	-4.08	-4.55	-11.13	-10.05	-5.64	-6.71	-4.46	-5.10	-12.21
$\mathcal{A}_{class}$	-3.89	-6.54	-8.10	-8.63	-7.12	-5.42	-6.14	-5.96	**-13.24**	-9.47	-4.34	-4.55	-5.92	-4.56	-11.98
GF-Attack	**-5.56**	**-7.09**	**-9.10**	**-9.95**	**-9.74**	**-8.47**	**-9.04**	**-8.06**	-12.38	**-10.91**	**-7.06**	**-7.20**	**-7.64**	**-7.14**	**-13.26**

be consistent with [20]. For a statistical overview of datasets, please kindly refer to [20].

Baselines. In the current literature, few studies strictly follow the restricted black-box attack setting. They utilize the additional information to help construct the attackers, such as labels [20], gradients [19], etc. Therefore, we compare four baselines with our proposed attack framework under the RBA setting as follows:

- *Random* [19]: for each perturbation, randomly choosing insertion or removing of an edge in graph G. We report averages over 10 different seeds to alleviate the influence of randomness.
- *Degree* [51]: for each perturbation, inserting or removing an edge based on degree centrality, which is equivalent to the sum of degrees in original graph G.
- *RL-S2V* [19]: a reinforcement learning-based attack method, which learns the generalizable attack policy for GCN under the RBA scenario.
- $\mathcal{A}_{class}$ [26]: a matrix perturbation theory based black-box attack method designed for DeepWalk. Then $\mathcal{A}_{class}$ evaluates the targeted attacks on vertex classification by learning a logistic regression.

Target Models. To validate the generalization ability of *GF-Attack*, we choose four popular GEMs: GCN [22], SGC [32], DeepWalk [50] and LINE [30] for evaluation. GCN and SGC are GCNs and the others are sampling-based GEMs. Considering that both GCN and SGC are fixed-filter GCNs, we additionally choose a representative parameterized-filter variant, ChebyNet [33], as a victim model to further evaluate the effectiveness of our framework. The attack loss for ChebyNet is consistent with GCN due to their theoretical connection as we analyzed in Section 4.2. For ChebyNet, we set the order of Chebyshev polynomials K as 2. For DeepWalk, we set the window-size as 5. For both LINE and DeepWalk, the number of negative sampling in skip-gram is set as 1, and the embedding dimension is chosen as 32. A logistic regression classifier is connected to the output embeddings of sampling-based methods for classification. Without other specification, all GCNs contain two layers.

Attack Configuration. A small budget β is applied to regulate all the attackers. To make this attack task more challenging, the budget β is set to 1. Specifically, the attacker is limited to only adding/deleting a single edge given a target vertex t. For our method, we set the parameter T in our general attack model as $n - T = 128$, which means that we choose the top-T smallest eigenvalues for T-rank approximation in the embedding quality measure. Unless otherwise indication, the order of graph filter in *GF-Attack* model is set as $K = 2$. Following the setting in [20], we split the graph into labeled (20%) and unlabeled vertices (80%). Further, the labeled vertices are splitted into equal parts for training and validation. The labels and classifier are invisible to the attacker due to the RBA setting. The attack performance is evaluated by the decrease of vertex classification accuracy following [19]. Without otherwise specification, the attack is conducted on GCNs under the evasion setting and on sampling-based GEMs under the poisoning setting.

TABLE 4
Summary of the change in classification accuracy (in percent) compared to the clean/original graph. Single edge perturbation under the RBA and evasion settings. Lower is better. We use *Cheby* for ChebyNet and *DW* for DeepWalk here to save space.

Dataset	Cora			Citeseer		
Models	GCN	SGC	Cheby	GCN	SGC	Cheby
(unattacked)	80.20	78.82	80.33	72.50	69.68	70.96
Random	-1.22	-1.90	-1.05	-1.73	-1.86	-1.80
Degree	-2.21	-4.59	-3.54	-2.71	-2.91	-1.77
RL-S2V	-3.25	-3.74	-4.18	-2.30	-3.80	-2.78
$\mathcal{A}_{class}$	-2.83	-4.03	-3.54	-1.92	-3.78	-3.67
GF-Attack	**-4.76**	**-5.23**	**-4.54**	**-4.14**	**-5.33**	**-4.96**

5.1 Attack Performance Evaluation

In this section, we combine the original results from the AAAI version [27] and evaluate the overall attack performance of different attackers. Note that the sampling-based GEMs can only be attacked under the poisoning setting [26], since sampling-based GEMs rely on training to generate new embeddings for perturbed graphs. Thus here we choose to perform the attack on all victim models under this setting and damage GCNs under the evasion setting alone in Section 5.4. Meanwhile, we choose to use Eq. (10) as the attack loss for all victim models here. In contrast to our AAAI version [27] which uses different attack losses for different types of GEMs, we take a step further to better demonstrate the effectiveness of *GF-Attack* under a more black-box setting, since this new setting removes the assumption of what type the victim model is.

Attack on GCNs. Table 3 summarizes the attack results of different attackers on GCNs. Our *GF-Attack* outperforms other attackers on all datasets and all models, even on the more complex parameterized-filter model ChebyNet. Moreover, *GF-Attack* performs quite well on 2 layers GCN

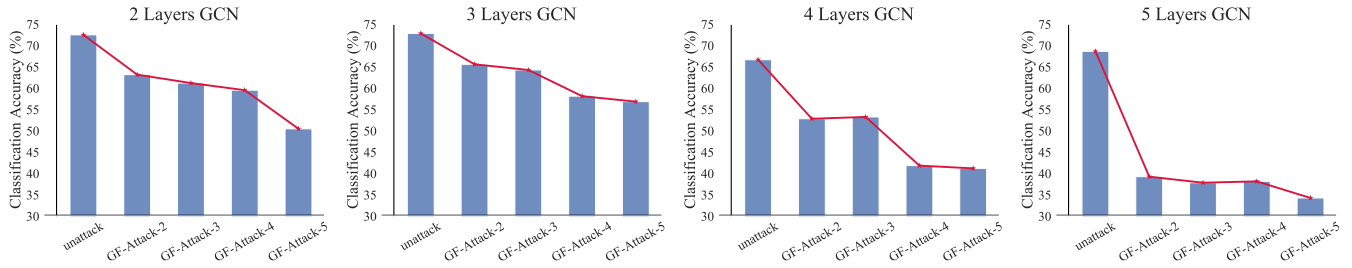


Fig. 2. Comparison of the classification performance between order K in *GF-Attack* (x-axis) and the number of layers in GCN. Lower is better.

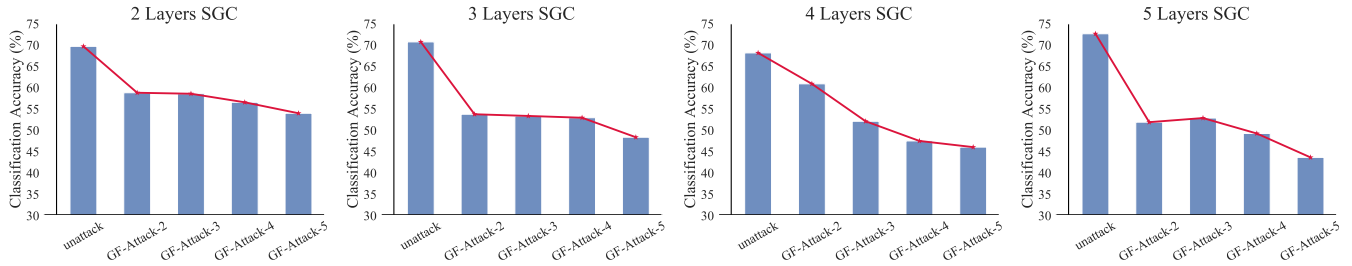


Fig. 3. Comparison of the classification performance between order K of *GF-Attack* (x-axis) and the number of layers in SGC. Lower is better.

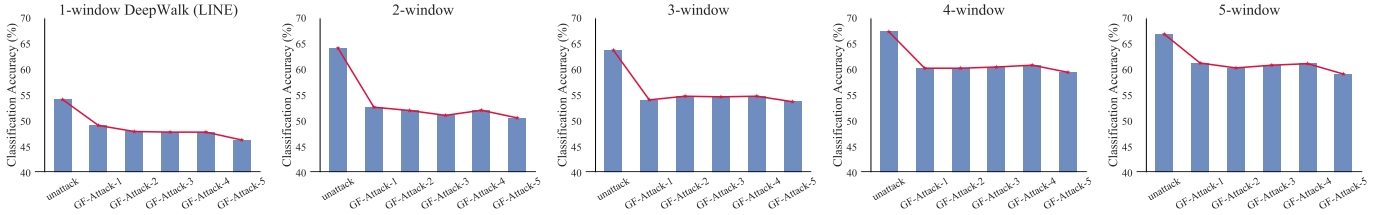


Fig. 4. Comparison of the classification performance between order K of *GF-Attack* (x-axis) and the number of window-size in DeepWalk. Lower is better. Note that LINE is a 1-window special case of DeepWalk.

with nonlinear activation. This implies the generalization ability of *GF-Attack* on GCNs as discussed in Section 4.2.

Attack on Sampling-based GEMs. Table 3 also summarizes the results of different attackers on the sampling-based GEMs. As expected, *GF-Attack* achieves the best performance nearly on all victim models. It validates the effectiveness of *GF-Attack* on attacking sampling-based GEMs.

Another interesting observation is that the attack performance on LINE is much better than that on DeepWalk. This result may due to the deterministic structure of LINE, while the random sampling procedure in DeepWalk may help raise its resistance to adversarial attacks. Moreover, *GF-Attack* on all graph filters successfully drop the classification accuracy on both GCNs and sampling-based GEMs, which again indicates the transferability of the adversary examples generated by our general framework in practice.

5.2 Evaluation of Multi-layer GCNs and Multi-window-size Sampling-based GEMs.

To further investigate the transferability of our framework, we conduct attacks towards different multi-layer GCNs and multi-window-size sampling-based GEMs w.r.t. the order of graph filter under our *GF-Attack* framework supplementary to the original AAAI version [27].

Figure 2, Figure 3 and Figure 4 present the attack results on 2, 3, 4 and 5 layers GCN and SGC, and DeepWalk with window-size 1 (LINE), 2, 3, 4 and 5 on Citeseer. The number followed by *GF-Attack* indicates the graph filter order K used in the attack loss. From Figure 2 to Figure 4, we can have some interesting observations:

- All the adversarial losses with different orders K can perform successful attacks on all models, which again indicates the effectiveness of *GF-Attack*.
- Particularly, *GF-Attack-5* achieves the best-attack performance in most cases. It implies that the higher-order filter contains more fruitful information and has positive effects on the attacks targeting simpler models. This finding is consistent with the Theorem 2 from Section 4.2.2.
- The attack performance on SGC seems better than GCN under most of the settings. We conjecture that the non-linearity between layers in GCN can enhance the robustness of GCN.
- The performance of the adversarial attack on DeepWalk is better when the window-size grows for window-size ranging from 2 to 5. This is consistent with the mechanism of DeepWalk since when the window-size is larger, vertices from the further neighborhood of the target vertex will participate in learning embeddings.

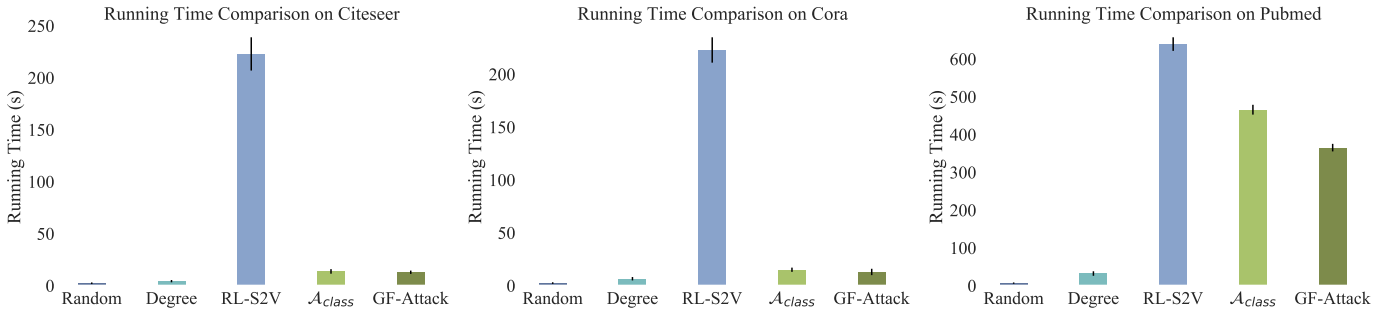
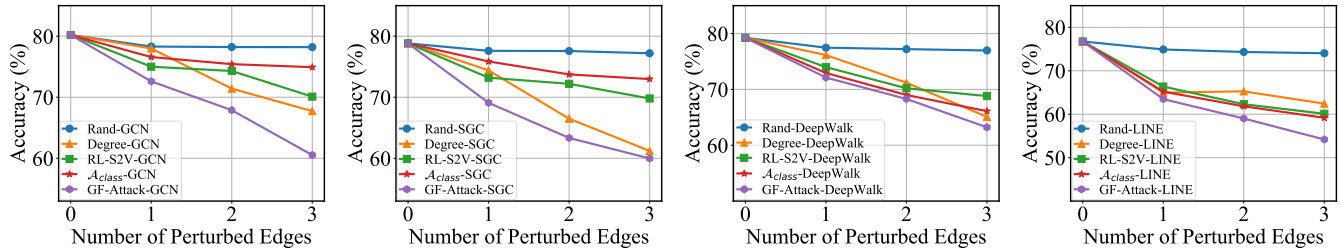


Fig. 5. Running time (s) comparison overall baseline methods on all datasets. We report the 10 times average running time of processing a single vertex for each model and the error bars are at the top of each bar.



(a) GCN (b) SGC (c) DeepWalk (d) LINE

Fig. 6. Multiple-edge attack results on Cora under RBA setting. Lower is better.

5.3 Evaluation under Multi-edge Perturbation Setting

In this section, we evaluate the performance of attackers with multi-edge perturbation, i.e. $\beta \geq 1$, on all models supplementary to the original AAAI version [27]. The results of multi-edge perturbations on the Cora dataset under the RBA setting are reported in Figure 6.

Clearly, with the increase of the number of perturbed edges, the attack performance gets better for each attacker. *GF-Attack* outperforms all the other baselines in all cases. It validates that *GF-Attack* can still perform well when the fixed budget β becomes larger.

5.4 Evaluation under Evasion Setting

Since we mainly conduct analysis under the evasion setting in this work, we further investigate the performance of our framework under this setting with one-edge perturbation to demonstrate the ability of *GF-Attack*. As shown in Table 4, we observe that the performance of all attack methods is degraded under the evasion attack setting, which implies that the GEMs could be misled by the adversarial examples during training under the poisoning setting. Further, *GF-Attack* still consistently outperforms all baselines, though it is not specifically designed for the poisoning attacks.

5.5 Computational Efficiency Analysis

In this section, we empirically evaluate the computational efficiency of our *GF-Attack*. A comparison of the average values of the running time for 10 runs of our algorithm for all datasets is given in Figure 5. While being less efficient than two native baselines (*Random* and *Degree*), our *GF-Attack* is much faster than the novel baselines *RL-S2V* and $\mathcal{A}_{class}$. Combining the performance in Table 3, it reads that *GF-Attack* is not only effective in performance but also efficient computationally.

6 Conclusion

In this paper, we consider the adversarial attack on different kinds of GEMs under the restricted black-box attack scenario. From the view of graph signal processing, we try to formulate the procedure of graph embedding methods as a general graph signal processing with the corresponding graph filters. Then we construct a restricted adversarial attack framework which aims to attack the graph filter only by the adjacency matrix and the feature matrix. Thereby, a general optimization problem is constructed by measuring the embedding quality and an effective algorithm is derived accordingly to solve it. Experiments show the vulnerability of different kinds of novel GEMs to our general attack framework.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (No. 2020AAA0106300, No. 2018AAA0102004), the National Natural Science Foundation of China (No. 62102222, No.62006137, No. U1936219, No. 62141607), Tencent AI Lab Rhino-Bird Visiting Scholars Program (VS2022TEG001), and the 2020 Tencent AI Lab Rhino-Bird Elite Training Program. We would like to thank Daniel Zügner from the Technical University of Munich for his valuable suggestions and discussions.

REFERENCES

[1] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *TKDE*, vol. 31, no. 5, pp. 833–852, 2018.

[2] Y. Rong, T. Xu, J. Huang, *et al.*, "Deep graph learning: Foundations, advances and applications," in *KDD*, 2020, pp. 3555–3556.

[3] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," in *NeurIPS*, 2018, pp. 4563–4572.

[4] Z. Peng, W. Huang, M. Luo, *et al.*, "Graph representation learning via graphical mutual information maximization," in *Proceedings of The Web Conference 2020*, 2020, pp. 259–270.

[5] D. K. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, *et al.*, "Convolutional networks on graphs for learning molecular fingerprints," *NeurIPS*, pp. 2224–2232, 2015.

[6] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *AAAI*, 2018.

[7] Y. Rong, Y. Bian, T. Xu, *et al.*, "Self-supervised graph transformer on large-scale molecular data," in *NeurIPS*, vol. 33, 2020, pp. 12 559–12 571.

[8] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *The WebConf*, 2019, pp. 3049–3055.

[9] T. Bian, X. Xiao, T. Xu, *et al.*, "Rumor detection on social media with bi-directional graph convolutional networks," in *AAAI*, 2020.

[10] J. Li, Y. Rong, H. Cheng, H. Meng, W. Huang, and J. Huang, "Semi-supervised graph classification: A hierarchical graph perspective," in *The WebConf*, ACM, 2019, pp. 972–982.

[11] R. Zeng, W. Huang, M. Tan, *et al.*, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019, pp. 7094–7103.

[12] S. Wang, Z. Xu, C. Yan, and J. Huang, "Graph convolutional nets for tool presence detection in surgical videos," in *IPMI*, Springer, 2019, pp. 467–478.

[13] A. Raju, J. Yao, M. M. Haq, J. Jonnagaddala, and J. Huang, "Graph attention multi-instance learning for accurate colorectal cancer staging," in *MICCAI*, Springer, 2020, pp. 529–539.

[14] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *WSDM*, 2017, pp. 601–610.

[15] H. Chang, Y. Rong, T. Xu, *et al.*, "Spectral graph attention network with fast eigen-approximation," in *CIKM*, 2021, pp. 2905–2909.

[16] F. Gu, H. Chang, W. Zhu, S. Sojoudi, and L. El Ghaoui, "Implicit graph neural networks," *NeurIPS*, vol. 33, pp. 11 984–11 995, 2020.

[17] C. Guan, Z. Zhang, H. Li, *et al.*, "AutoGL: A library for automated graph learning," in *ICLR 2021 Workshop GTRL*, 2021.

[18] N. Akhtar and A. S. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[19] H. Dai, H. Li, T. Tian, *et al.*, "Adversarial attack on graph structured data," *ICML*, pp. 1115–1124, 2018.

[20] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *KDD*, 2018, pp. 2847–2856.

[21] M. Jin, H. Chang, W. Zhu, and S. Sojoudi, "Power up! robust graph convolutional network via graph powering," in *AAAI*, 2021.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2017.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.

[24] L. Sun, J. Wang, P. S. Yu, and B. Li, "Adversarial attack and defense on graph data: A survey.," *arXiv preprint arXiv:1812.10528*, 2018.

[25] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," *ICLR*, 2019.

[26] A. Bojchevski and S. Günnemann, "Adversarial attacks on node embeddings via graph poisoning," in *ICML*, PMLR, 2019.

[27] H. Chang, Y. Rong, T. Xu, *et al.*, "A restricted black-box adversarial framework towards attacking graph embedding models," in *AAAI*, vol. 34, 2020, pp. 3389–3396.

[28] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[29] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *TKDE*, 2020.

[30] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *The WebConf*, 2015, pp. 1067–1077.

[31] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.

[32] F. Wu, T. Zhang, A. H. Souza Jr., C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019.

[33] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *NeurIPS*, pp. 3844–3852, 2016.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[35] B. Xu, H. Shen, Q. Cao, K. Cen, and X. Cheng, "Graph convolutional networks using heat kernel for semi-supervised learning," in *IJCAI*, AAAI Press, 2019, pp. 1928–1934.

[36] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *WSDM*, 2018, pp. 459–467.

[37] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[38] K. Xu, H. Chen, S. Liu, *et al.*, "Topology attack and defense for graph neural networks: An optimization perspective," in *IJCAI*, AAAI Press, 2019, pp. 3961–3967.

[39] H. Wu, C. Wang, Y. Tyshetskiy, A. Docherty, K. Lu, and L. Zhu, "Adversarial examples for graph data: Deep insights into attack and defense," in *IJCAI*, AAAI Press, 2019, pp. 4816–4823.

[40] J. Li, H. Zhang, Z. Han, Y. Rong, H. Cheng, and J. Huang, "Adversarial attack on community detection by hiding individuals," in *The WebConf*, 2020, pp. 917–927.

[41] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, and E. E. Papalexakis, "All you need is low (rank) defending against adversarial attacks on graphs," in *WSDM*, 2020, pp. 169–177.

[42] J. Ma, S. Ding, and Q. Mei, "Towards more practical adversarial attacks on graph neural networks," *NeurIPS*, vol. 33, 2020.

[43] A. Vidanage, P. Christen, T. Ranbaduge, and R. Schnell, "A graph matching attack on privacy-preserving record linkage," in *CIKM*, 2020, pp. 1485–1494.

[44] Z. Zhang, Z. Zhang, Y. Zhou, Y. Shen, R. Jin, and D. Dou, "Adversarial attacks on deep graph matching," *NeurIPS*, vol. 33, 2020.

[45] P. Elinas, E. V. Bonilla, and L. Tiao, "Variational inference for graph convolutional networks in the absence of graph data and adversarial settings," *NeurIPS*, vol. 33, 2020.

[46] X. Zhang and M. Zitnik, "Gnnguard: Defending graph neural networks against adversarial attacks," *NeurIPS*, vol. 33, 2020.

[47] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," *NeurIPS*, vol. 33, 2020.

[48] H. Chang, Y. Rong, T. Xu, *et al.*, "Not all low-pass filters are robust in graph convolutional networks," *NeurIPS*, 2021.

[49] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *TKDE*, vol. 30, no. 9, pp. 1616–1637, 2018.

[50] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, ACM, 2014, pp. 701–710.

[51] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *CIKM*, 2012, pp. 245–254.

[52] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[53] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[54] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *IJCAI*, 2015.

[55] K. Nar, O. Ocal, S. S. Sastry, and K. Ramchandran, "Cross-entropy loss and low-rank features have responsibility for adversarial examples," *arXiv preprint arXiv:1901.08360*, 2019.

[56] R. A. Horn and J. guang Sun, *Matrix perturbation theory*. 1990.

[57] D. Zhu, P. Cui, Z. Zhang, J. Pei, and W. Zhu, "High-order proximity preserved embedding for dynamic networks," *TKDE*, vol. 30, pp. 2134–2144, 2018.

[58] C. Li, J. Ma, X. Guo, and Q. Mei, "Deepcas: An end-to-end predictor of information cascades," in *The WebConf*, 2017, pp. 577–586.

[59] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "Struc2vec: Learning node representations from structural identity," in *KDD*, 2017, pp. 385–394.

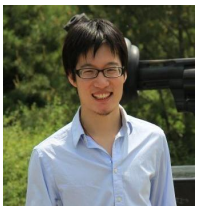
Heng Chang is currently pursuing a Ph.D. Degree in the Tsinghua-Berkeley Shenzhen Institute at Tsinghua University. He received his B.S. from the Department of Electronic Engineering, Tsinghua University in 2017. His research interests focus on representation learning, adversarial robustness and machine learning on graph/relational structured data. He has published several papers in prestigious conferences including NeurIPS, AAAI, CIKM, AISTATS, *etc*.

Yu Rong is a senior researcher of Machine Learning Center in Tencent AI Lab. He received the Ph.D. degree from The Chinese University of Hong Kong in 2016. He joined Tencent AI Lab in June 2017. His main research interests include social network analysis, graph neural networks, and large-scale graph systems. In Tencent AI Lab, he is working on building the large-scale graph learning framework and applying the deep graph learning model to various applications, such as ADMET prediction and malicious detection. He has published several papers on data mining, machine learning top conferences, including the Proceedings of KDD, WWW, NeurIPS, ICLR, CVPR, ICCV, *etc*.

Tingyang Xu is a senior researcher of Machine Learning Center in Tencent AI Lab. He obtained the Ph.D. degree from The University of Connecticut in 2017 and joined Tencent AI Lab in July 2017. In Tencent AI Lab, he is working on deep graph learning, graph generations and applying the deep graph learning model to various applications, such as molecular generation and rumor detection. His main research interests include social network analysis, graph neural networks, and graph generations, with particular focus on design deep and complex graph learning models for molecular generations. He has published several papers on data mining, machine learning top conferences KDD, WWW, NeurIPS, ICLR, CVPR, ICML, *etc*.

Wenbing Huang is now an assistant researcher at Tsinghua University. He received his Ph.D. degree of computer science and technology from Tsinghua University in 2017. His current research mainly lies in the areas of machine learning, computer vision, and robotics, with particular focus on learning on irregular structures including graphs and videos. He has published about 30 peer-reviewed top-tier conference and journal papers, including the Proceedings of NeurIPS, ICLR, ICML, CVPR, etc. He served (will serve) as a Senior Program Committee of AAAI 2021, Area Chair of ACMMMM workshop HUMA 2020, and Session Chair of IJCAI 2019.

Honglei Zhang is working on advertising recommendation in Tencent. He received his M.S.in Electrical and Computer Engineering Institute, from Georgia Institution of Technology in 2021 and received his B.S. from Tianjin University in 2018. His main research interests include graph represention learning and advertising recommendation. He has published several top-tier conference papers including AAAI, WWW, NeurIPS, *etc*.



Peng Cui is an Associate Professor with tenure in Tsinghua University. He got his PhD degree from Tsinghua University in 2010. His research interests include causally-regularized machine learning, network representation learning, and social dynamics modeling. He has published more than 100 papers in prestigious conferences and journals in data mining and multimedia. His recent research won the IEEE Multimedia Best Department Paper Award, SIGKDD 2016 Best Paper Finalist, ICDM 2015 Best Student Paper Award, SIGKDD 2014 Best Paper Finalist, IEEE ICME 2014 Best Paper Award, and MMM13 Best Paper Award. He is PC co-chair of CIKM2019 and MMM2020, SPC or area chair of WWW, ACM Multimedia, IJCAI, AAAI, etc., and Associate Editors of IEEE TKDE, IEEE TBD, ACM TIST, and ACM TOMM etc. He received ACM China Rising Star Award in 2015, and CCF-IEEE CS Young Scientist Award in 2018. He is now a Senior Member of IEEE, and Member of ACM.

Xin Wang is currently an Assistant Professor at the Department of Computer Science and Technology, Tsinghua University. He got both of his Ph.D. and B.E degrees in Computer Science and Technology from Zhejiang University, China. He also holds a Ph.D. degree in Computing Science from Simon Fraser University, Canada. His research interests include relational media big data analysis, multimedia intelligence and recommendation in social media. He has published several high-quality research papers in top conferences including ICML, KDD, WWW, SIGIR ACM Multimedia etc. He is the recipient of 2017 China Postdoctoral innovative talents supporting program. He receives the ACM China Rising Star Award in 2020.

Wenwu Zhu is currently a Professor in the Department of Computer Science and Technology at Tsinghua University, the Vice Dean of National Research Center for Information Science and Technology. Prior to his current post, he was a Senior Researcher and Research Manager at Microsoft Research Asia. He was the Chief Scientist and Director at Intel Research China from 2004 to 2008. He worked at Bell Labs New Jersey as Member of Technical Staff during 1996-1999. He received his Ph.D. degree from New York University in 1996. His current research interests are in the area of data-driven multimedia networking and multimedia intelligence. He has published over 350 referred papers, and is inventor or co-inventor of over 50 patents. He received eight Best Paper Awards, including ACM Multimedia 2012 and IEEE Transactions on Circuits and Systems for Video Technology in 2001 and 2019. He served as EiC for IEEE Transactions on Multimedia (2017-2019). He serves as the chair of the steering committee for IEEE Transactions on Multimedia, and he serves Associate EiC for IEEE Transactions for Circuits and Systems for Video technology. He serves as General Co-Chair for ACM Multimedia 2018 and ACM CIKM 2019, respectively. He is an AAAS Fellow, IEEE Fellow, SPIE Fellow, and a member of The Academy of Europe (Academia Europaea).

Junzhou Huang is an Associate Professor in the Computer Science and Engineering department at the University of Texas at Arlington. He also served as the director of machine learning center in Tencent AI Lab. He received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China, the M.S. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the Ph.D. degree in Computer Science at Rutgers, The State University of New Jersey. His major research interests include machine learning, computer vision and imaging informatics. He was selected as one of the 10 emerging leaders in multimedia and signal processing by the IBM T.J. Watson Research Center in 2010. His work won the MICCAI Young Scientist Award 2010, the FIMH Best Paper Award 2011, the MICCAI Young Scientist Award Finalist 2011, the STMI Best Paper Award 2012, the NIPS Best Reviewer Award 2013, the MICCAI Best Student Paper Award Finalist 2014 and the MICCAI Best Student Paper Award 2015. He received the NSF CAREER Award in 2016.