Wang, Xinjue; Ollila, Esa; Vorobyov, Sergiy A.

Graph Convolutional Neural Networks Sensitivity under Probabilistic Error Model

Published in:
IEEE Transactions on Signal and Information Processing over Networks

DOI:
10.1109/TSIPN.2024.3485532

Published: 01/01/2024

Document Version
Publisher's PDF, also known as Version of record

Graph Convolutional Neural Networks Sensitivity Under Probabilistic Error Model

Xinjue Wang, *Graduate Student Member, IEEE*, Esa Ollila, *Senior Member, IEEE*, and Sergiy A. Vorobyov, *Fellow, IEEE*

***Abstract*—Graph Neural Networks (GNNs), particularly Graph Convolutional Neural Networks (GCNNs), have emerged as pivotal instruments in machine learning and signal processing for processing graph-structured data. This paper proposes an analysis framework to investigate the sensitivity of GCNNs to probabilistic graph perturbations, directly impacting the graph shift operator (GSO). Our study establishes tight expected GSO error bounds, which are explicitly linked to the error model parameters, and reveals a linear relationship between GSO perturbations and the resulting output differences at each layer of GCNNs. This linearity demonstrates that a single-layer GCNN maintains stability under graph edge perturbations, provided that the GSO errors remain bounded, regardless of the perturbation scale. For multilayer GCNNs, the dependency of system's output difference on GSO perturbations is shown to be a recursion of linearity. Finally, we exemplify the framework with the Graph Isomorphism Network (GIN) and Simple Graph Convolution Network (SGCN). Experiments validate our theoretical derivations and the effectiveness of our approach.**

***Index Terms*—Graph convolutional neural network, graph shift operator, sensitivity analysis, structural perturbation.**

I. Introduction

GRAPH neural networks (GNNs) have steadily gained prominence as an innovative tool in machine learning and signal processing, exhibiting unparalleled efficiency in processing data encapsulated within complex graph structures [1], [2], [3]. Uniquely designed, GNNs utilize a system of intricately coupled graph filters (GFs) with nonlinear activation functions, enabling the effective transformation and propagation of information within the graph [4].

Different GNN architectures can be delineated based on the GFs, which are an integral to the functioning of GNNs. A notable example of these architectures uses graph-convolutional filters. The GNN employing this design is known as the Graph Convolutional Neural Network (GCNN). Some examples of GCNNs include the vanilla Graph Convolutional Network (GCN) [5], Graph Isomorphism Network (GIN) [6], Simple Graph Convolution Network (SGCN) [7], [8], and Cayley Graph Convolutional Network (CayleyNet) [9]. In contrast to the aforementioned GCNNs, there exist non-convolutional GNNs such as the Graph Attention Network (GAT) [10] and Edge Varying Graph Neural Network (EdgeNet) [11], which utilize edge-varying graph filters [12].

This paper delves into the GCNN, which blends graph convolutional filters with nonlinear activation functions. Graph convolutional filters couple the data and graph with the underlying graph matrix, named graph shift operator (GSO), which can be, for example, the graph adjacency matrix or graph Laplacian, encoding the interactions between data samples [13]. Based on the GSO, the graph filter captures the structural information by aggregating the data propagated within its $k-$hop neighborhoods, and feeds it to the next layer after processing, which can be applying graph coarsening and pooling [6], [14]. As the key component of GCNNs, GSO presents the graph structure, and is typically assumed to be perfectly known. The precise estimation of the hidden graph structure is essential for successfully performing feature propagation in a convolution layer [15], [16], [17].

GSOs form the foundation of GCNN structures. Any perturbation in the graph structure has a direct bearing on the operations of a GCNN. Previous studies in graph signal processing (GSP) and GNN have examined both deterministic and probabilistic perturbations affecting GSOs. A probabilistic graph perturbation model for a partially correct estimation of the adjacency matrix is proposed in [18], where a perturbed graph is modeled as a combination of the true adjacency matrix and a perturbation term specified by Erdős-Rényi (ER) graph. The work [19] explores perturbations in graphs using random edge sampling, a scheme characterized by randomly deleting existing edges. In [20], a GSO perturbation strategy is formulated leveraging a general first-order optimization method, which concurrently imposes a constraint on the extent of edge perturbation. In [21], the authors propose to perturb eigenvector pairs of the graph Laplacian, considering single and multiple edge perturbations, under small perturbation assumption. Here, small perturbations refer to changes in a small percentage of edges.

The stability of GFs and GCNNs under GSO perturbations is one of the key research areas in signal processing (SP) and computer science (CS). In the SP community, research focuses on the relationship between the system's output differences and the GSO differences under evasion attacks, emphasizing changes in the learned representation. In [22], the authors provide bounds on the output changes of spectral GFs resulting from double edge rewiring on normalized augmented adjacency matrices.

Received 24 July 2023; revised 10 January 2024, 5 April 2024, and 21 August 2024; accepted 13 October 2024. Date of publication 23 October 2024; date of current version 6 November 2024. This work was supported by the Research Council of Finland under Grant 359848 and Grant 357715. The associate editor coordinating the review of this article and approving it for publication was Prof. Xiaowen Dong. *(Corresponding author: Esa Ollila.)*

The authors are with the Department of Information and Communications Engineering, Aalto University, 02150 Espoo, Finland (e-mail: xinjue.wang@aalto.fi; esa.ollila@aalto.fi; svor@ieee.org).

Digital Object Identifier 10.1109/TSIPN.2024.3485532

This study extends the stability results to SGCN and gives theoretical bounds. In [23], the authors present interpretable bounds to verify the stability of spectral GFs against graph edge perturbations. These bounds are derived under the constraint that the degree of any node after perturbation cannot exceed twice its original degree. In [24], the authors apply an additive error model with norm-bounded perturbations on unspecified GSOs to provide stability bounds for multi-layer GCNNs. This model is not generic as it does not explicitly account for the perturbation of graph edges. It primarily considers perturbations resembling a uniform scaling of edge weights, a limitation noted in [25]. Additionally, the bound of error matrix is defined based on the smallest operator norm achievable via node permutation. However, this permutation assumption may not suit social or citation networks where node identification is label-dependent, as noted in [22]. In [19], authors consider random edge deletions as the perturbation on GSOs, specifically focusing on adjacency matrices and graph Laplacians. It concludes that both the GF and GCNN are linearly stable with respect to several factors, including the probability of edge dropping, nonlinearity, and the width and depth of the network architecture. Nevertheless, in the experiments of [19], the maximum edge deletion probability is set to 6%, indicating a limited scale on perturbation. Works in CS [26], [27], [28], [29], [30] focus on the effects of adversarial attacks affecting GCNN accuracy, considering both evasion and poisoning attacks. The focus is on the impacts of such attacks on the downstream task. For instance, under evasion attacks, [27] demonstrates the reduction on GCNN's accuracy under small perturbations, while maintaining the degree distributions after the attack, and [30] demonstrates the significant drop of accuracy of GCN when 5% of edges are altered.

In this paper, we introduce a sensitivity analysis framework for GCNN under the probabilistic edge perturbation model [18]. We understand *stability* as the characteristic of a system to maintain bounded output under perturbations, while *sensitivity analysis* is an examination of how variations in the output depend on influencing factors. Our analysis concentrates on studying the effects of evasion attacks. We use statistical analysis to give expected bounds for GSO errors (Theorem 1 and Proposition 1). These error bounds are explicitly dependent on the parameters of the error model. Then, we establish a sensitivity analysis framework for both GF (Theorem 2) and multilayer GCNN (Theorem 3) by giving expected bounds for differences of outputs because of GSO errors. Finally, we exemplify the framework with GIN (Corollary 1) and SGCN (Corollary 2), and empirically show that under large-scale graph perturbations (significant edge modifications), GCNNs maintain stability.

Our detailed contributions are summarized as follows.

1) Probabilistic error model: The probabilistic edge perturbation model considered is general and practically appealing. It is grounded in stochastic block models, supports both deletion and addition of edges, and permits a broader perturbation scale. The corresponding analysis approach contrasts with the constrained perturbations in existing GCNN analyses, which involve such restrictions as permitting only edge deletions in [19], double edge rewiring in [22], and small norm bounded errors in [24].

2) Tight GSO error bound: We give tighter expected bounds on GSO errors compared to our previous conference work [31], in which the bounds are deterministic. We use the ℓ_1 norm suggested in [23] to bound the ℓ_2 norm and make this bound interpretable by specifically tracking the changed node degrees, which can be directly linked to parameters of the error model (probabilities of deleting and adding edges). Additionally, our bound does not require the eigendecomposition of GSO [19], [24], which is computationally heavy for large graphs.

3) Generic sensitivity analysis framework: Compared to previous works [19], [22], [24], our proposed analysis framework is more generic in the following aspects. *(i)* We remove the assumption on limited scale perturbation and allow for a large perturbation budget, for instance that 50% of edges are deleted and 70% of edges are added (compared to the original number of edges). Our analysis is shown empirically to be valid even under such perturbation, while the maximum edge perturbation addressed in the current literature is 10% of edges [23]. *(ii)* We provide expected bounds under a probabilistic perspective, while the deterministic perturbations can be seen as special cases of our analysis. *(iii)* This framework is applicable to general GCNN models, with specific adjustments for GSO, graph shifts count, network layer count, and activation functions.

Outline: The remainder of this paper is structured as follows. In Sections II and III, we establish the fundamentals of GCNNs and proceed to formulate the problem. Section IV bounds the difference between original and perturbed GSOs, with particular emphasis on two cases: the adjacency matrix and its normalized version. Section V encompasses both GFs and GCNNs like GIN and SGCN, and demonstrates that variations in the output of each GCNN layer in response to graph perturbations are linearly bounded. Empirical validations presented in Section VI use numerical experiments with both synthetic and real-world data to corroborate the proposed theorems, thereby attesting to the reliability of our sensitivity analysis model. Section VII concludes the paper and discusses the future work.

Notation: Boldface lower case letters such as $\mathbf{x}$ represent column vectors, while boldface capital letters like $\mathbf{X}$ denote matrices. A vector full of ones is symbolized as $\mathbf{1}_N$, and a $N \times N$ matrix full of ones is expressed as $\mathbf{1}_{N\times N} = \mathbf{1}_N \mathbf{1}_N^\top$. The identity matrix of size $N \times N$ is represented as $\mathbf{I}_{N\times N}$. The i-th row or column of the matrix $\mathbf{A}$ is given as $\mathbf{A}_i$, and the (i, j)-th element in matrix $\mathbf{A}$ is denoted as $[\mathbf{A}]_{i,j}$ or $\mathbf{A}_{i,j}$. Vector ℓ_1 norm is defined as follows: $\|\mathbf{a}\|_1 = \sum_j |\mathbf{a}_j|$. Matrix norms are defined as follows: the ℓ_1 norm is represented as $\|\mathbf{A}\|_1 = \max_j \sum_i |\mathbf{A}_{i,j}|$, the ℓ_2 norm as $\|\mathbf{A}\| = \|\mathbf{A}\|_2 = \sqrt{\max(\text{eig}(\mathbf{A}^\top \mathbf{A}))}$ (largest singular value of $\mathbf{A}$), and the ℓ_∞ norm as $\|\mathbf{A}\|_\infty = \max_i \sum_j |\mathbf{A}_{i,j}|$. In addition, the Hadamard product is expressed with the symbol $\circ$. We use $\Pr(\cdot)$ for probability, $\mathbb{E}(\cdot)$ for expectation, $\text{Var}(\cdot)$ for variance, and $\text{Cov}(\cdot, \cdot)$ for covariance.

II. Preliminaries

Graph theory, GSP, and GCNN form the cornerstone of data analysis in irregular domains. The GSO plays a key role in

directing information flow across the graph, thereby enabling the creation of GFs and the design of GCNNs.

The sensitivity analysis of the GSO, which essentially involves matrix sensitivity analysis, provides an empirical insight into the system's resilience to perturbations. The GCNN, with its local architecture, maintains most of the properties of the graph convolutional filter, making it an ideal tool for sensitivity analysis. These preliminary concepts are essential for the implementation of sensitivity analysis in a graph-based context.

Graph Basics: Consider an undirected and unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where the node set $\mathcal{V} = \{1, \ldots, N\}$ consists of N nodes, the edge set $\mathcal{E}$ is a subset of $\mathcal{V} \times \mathcal{V}$, and the edge weighting function $\mathcal{W} : \mathcal{V} \times \mathcal{V} \to \{0, 1\}$ assigns binary edges. For an edge $(i, j) \in \mathcal{E}$, we have $\mathcal{W}(i, j) = \mathcal{W}(j, i) = 1$ due to our focus on undirected and unweighted graphs. We define the 1-hop neighboring set of a node i as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$, the degree of node i as d_i, and the minimum degree of nodes around i as $\tau_i = \min_{j \in \mathcal{N}_i} d_j$.

GSO: The Graph Shift Operator (GSO) $\mathbf{S} \in \mathbb{R}^{N \times N}$ symbolizes the structure of a graph and guides the passage and fusion of signals between neighboring nodes. It is often represented by the adjacency matrix $\mathbf{A}$, the Laplacian $\mathbf{L}$, or their normalized counterparts. These representations capture the graph's connectivity patterns, marking them indispensable tools for data analysis in both regular and irregular domains [32]. The adjacency matrix, denoted by $\mathbf{A}$, incorporates both the weighting function and the graph topology $\mathcal{G}$, where $[\mathbf{A}]_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $[\mathbf{A}]_{ij} = 0$ if $(i, j) \notin \mathcal{E}$. The Laplacian matrix $\mathbf{L}$ is defined by the adjacency matrix and a diagonal degree matrix $\mathbf{D}$. Specifically, $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1}_N)$ is a diagonal matrix, and $[\mathbf{D}]_{ii} = d_i$. The value $d_i = \sum_{j \in \mathcal{N}_i} [\mathbf{A}]_{ij}$ denotes the degree of node i. Moreover, normalized versions of the adjacency and Laplacian matrices are defined as $\mathbf{A}_\text{n} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\mathbf{L}_\text{n} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, respectively. These normalized versions help maintain consistency and manage potential variations in the scale of the data.

Graph Convolutional Filter: Using GSO, graph signals undergo shifting and averaging across their neighboring nodes. The signal on the graph is denoted by $\mathbf{x} \in \mathbb{R}^N$. Its i-th entry $[\mathbf{x}]_i = x_i$ specifies the data value at the node v_i. The one time shift of graph signal is simply $\mathbf{S}\mathbf{x}$, whose value at node i is $[\mathbf{S}\mathbf{x}]_i = \sum_{j \in \mathcal{N}_i} s_{ij} x_j$. After one graph shift, the value at node i is given by moving a local linear operator over its neighborhood values $\{x_j\}_{j \in \mathcal{N}}$. Based on the graph shifting, a graph convolutional filter $\mathbf{h}(\mathbf{S})$ with K taps is defined via polynomials of GSO and the filter weights $\mathbf{h} = \{h_k\}_{k=0}^K$ in the graph convolution

$$\mathbf{y} = h_0\mathbf{S}^0\mathbf{x} + \cdots + h_K\mathbf{S}^K\mathbf{x} = \sum_{k=0}^{K} h_k\mathbf{S}^k\mathbf{x} = \mathbf{h}(\mathbf{S})\mathbf{x}, \quad (1)$$

where $\mathbf{y}$ is the filter's output and $\mathbf{h}(\mathbf{S}) = \sum_{k=0}^K h_k\mathbf{S}^k$ is a shift-invariant graph filter with K taps, and denotes the weight of local information after K-hop data exchanges. The graph filter is then combined with the nonlinear activation function, forming the primary component of GCNN and contributing to its expressivity.

Graph Perceptron and GCNN: A Graph Perceptron [4] is a simple unit of transformation in the GCNN. The functionality of a graph perceptron can be seamlessly extended to accommodate graph signals with multiple features. Specifically, a multi-feature graph signal can be denoted by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_d] \in \mathbb{R}^{N \times d}$, where d signifies the number of features. The architecture of an L-layer GCNN is built upon cascading multiple graph perceptrons. It operates such that the output of a graph perceptron in a preceding layer serves as the input to the graph perceptron at the subsequent layer ℓ, where ℓ spans from 1 to L. We denote the feature fed to the first layer as $\mathbf{X}_0 = \mathbf{X}$. For an L-layer GCNN, the graph perceptron at layer ℓ can be represented as

$$\mathbf{Y}_\ell = \sum_{k=1}^{K} \mathbf{S}^k\mathbf{X}_{\ell-1}\mathbf{H}_{\ell k}, \quad \mathbf{X}_\ell = \sigma_\ell\left(\mathbf{Y}_\ell\right). \quad (2)$$

Here, $\mathbf{Y}_\ell$ signifies the intermediate graph filter output, $\sigma_\ell(\cdot)$ denotes the nonlinear activation function at layer ℓ, and graph signals at each layer are $\mathbf{X}_\ell$ and $\mathbf{X}_{\ell-1}$ with sizes of $\mathbb{R}^{N \times F_\ell}$ and $\mathbb{R}^{N \times F_{\ell-1}}$, respectively, where F_ℓ denotes the number of features at the ℓ-th layer. The bank of filter coefficients is represented by $\mathbf{H} = \{\mathbf{H}_{\ell k}\}_{\ell=1,\ldots,L;k=1,\ldots,K}$. By recursively using (2) until $\ell = L$, a general GCNN can be formulated as

$$\mathbf{\Phi}(\mathbf{X}; \mathbf{H}, \mathbf{S}) = \mathbf{X}_L = \sigma\left(\sum_{k=1}^{K} \mathbf{S}\mathbf{X}_{L-1}\mathbf{H}_{Lk}\right). \quad (3)$$

This representation captures the nature of GCNN operations, going through each layer and applying the corresponding transformation defined by the graph signal, filter coefficients, and the non-linearity function. This hierarchical arrangement facilitates the flow of information through successive layers, thus enabling effective learning from graph-structured data.

III. Problem Formulation

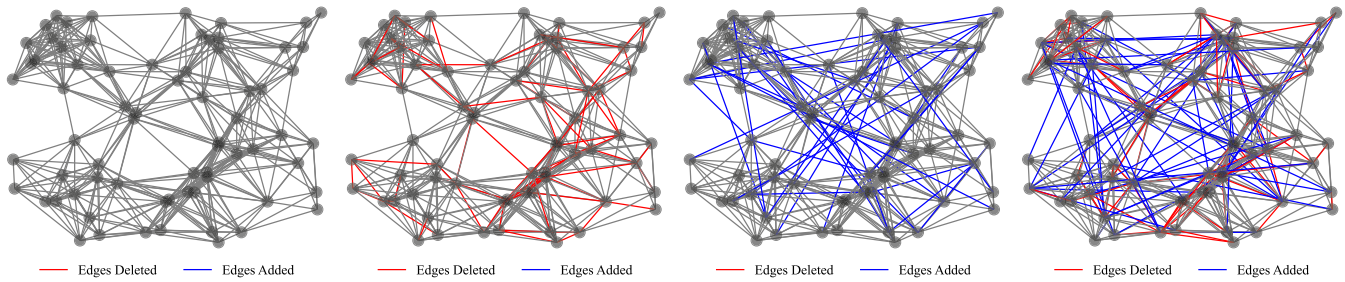
A pivotal aspect of understanding the sensitivity of a GCNN is the considerations of potential alterations in the underlying graph structure. These alterations can be broadly construed as perturbations to the GSO, intrinsically linking to changes in the graph topology. In the simplest form, any perturbation to the GSO can be depicted as

$$\hat{\mathbf{S}} = \mathbf{S} + \mathbf{E}, \quad (4)$$

where $\hat{\mathbf{S}}$ signifies the perturbed GSO, $\mathbf{S}$ is the original GSO, and $\mathbf{E}$ represents the error term. The spectral norm of this error term is denoted by

$$d(\hat{\mathbf{S}}, \mathbf{S}) = \|\hat{\mathbf{S}} - \mathbf{S}\| = \|\mathbf{E}\|. \quad (5)$$

Inspired by a previous work [18], we utilize a probabilistic error model to represent graph perturbations, where each edge of the graph is subject to perturbation independently. In this context, we primarily focus on the alterations occurring within the neighborhood of a particular node $u \in \mathcal{V}$. More specifically, the perturbed neighborhood may encompass added nodes ($\mathcal{A}_u$), deleted nodes ($\mathcal{D}_u$), and remaining nodes ($\mathcal{R}_u$), which ultimately leads to changes in node degree and modifications to the adjacency matrix. We aim to quantify the sensitivity of GSO in



(a) $\epsilon_1 = 0, \epsilon_2 = 0$ (b) $\epsilon_1 = 0.3, \epsilon_2 = 0$ (c) $\epsilon_1 = 0, \epsilon_2 = 0.1$ (d) $\epsilon_1 = 0.3, \epsilon_2 = 0.1$

Fig. 1. Visual representation of the probabilistic graph error model applied to a random geometric graph. From left to right: (a) Original graph; (b) Graph after edge deletions ($\epsilon_1 = 0.3, \epsilon_2 = 0$); (c) Graph after edge additions ($\epsilon_1 = 0, \epsilon_2 = 0.1$); (d) Graph after both edge deletions and additions ($\epsilon_1 = 0.3, \epsilon_2 = 0.1$). Deleted edges are marked in red and added edges are marked in blue. The transformations effectively illustrate the impact of perturbations modeled by (6).

relation to these perturbations. To this end, we adopt and expand upon the notation used in [22], [23] for clarity and consistency.

When the graph undergoes perturbations, it transforms into $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}}, \hat{\mathcal{W}})$, with the node set remaining unaffected. We express degrees of node $u \in \mathcal{V}$ in original and perturbed graphs as $d_u = \sum_j |[\mathbf{A}]_{u,j}|$ and $\hat{d}_u = \sum_j |[\hat{\mathbf{A}}]_{u,j}| = d_u + \delta_u$, respectively. Here, $\hat{\mathbf{A}}$ denotes the adjacency matrix of the perturbed graph $\hat{\mathcal{G}}$, and $\delta_u = \delta_u^+ - \delta_u^-$ is the degree change at node u, with $\delta_u^+ = |\mathcal{A}_u|$ and $\delta_u^- = |\mathcal{D}_u|$ corresponding to the number of edges added and deleted, respectively. We will further delve into the assumptions for the error model and its effects on the GCNN's performance in the following discussion.

A. Probabilistic Graph Error Model

In this work, we utilize an Erdös-Rényi (ER) graph-based model for perturbations on a graph adjacency matrix, following the approach proposed in [18]. The adjacency matrix of an ER graph is characterized by a random $N \times N$ matrix $\mathbf{\Delta}_\epsilon$, where each element of the matrix is generated independently, satisfying $\Pr([\mathbf{\Delta}_\epsilon]_{i,j} = 1) = \epsilon$ and $\Pr([\mathbf{\Delta}_\epsilon]_{i,j} = 0) = 1 - \epsilon$ for all $i \neq j$. The diagonal elements are zero, i.e., $[\mathbf{\Delta}_\epsilon]_{i,i} = 0$ for $i = 1, \ldots, N$, eliminating the possibility of self-loops. For the sake of our analysis, we also assume that the perturbed graph $\hat{\mathcal{G}}$ does not contain any isolated nodes, meaning that for all $u \in \mathcal{V}$, $\hat{d}_u \geq 1$. The model can be adapted by employing the lower triangular matrix $\mathbf{\Delta}_\epsilon^l$, and then defining $\mathbf{\Delta}_\epsilon = \mathbf{\Delta}_\epsilon^l + (\mathbf{\Delta}_\epsilon^l)^\top$. Consequently, by specifying the error term in (4), the perturbed adjacency matrix of a graph signal can be expressed as

$$\hat{\mathbf{A}} = \mathbf{A} - \mathbf{\Delta}_{\epsilon_1} \circ \mathbf{A} + \mathbf{\Delta}_{\epsilon_2} \circ (\mathbf{1}_{N \times N} - \mathbf{A}), \tag{6}$$

where the first term is responsible for edge deletion with probability ϵ_1, and the second term accounts for edge addition with probability ϵ_2. This error model can be conceptualized as superimposing two ER graphs on top of the original graph. To better illustrate this model, we utilize visual aids based on a random geometric graph [33], [34]. Fig. 1 visually represents the transition from the original graph to perturbed versions, which include the graph with only edge deletions ($\epsilon_1 = 0.3, \epsilon_2 = 0$), the graph with only edge additions ($\epsilon_1 = 0, \epsilon_2 = 0.1$), and the graph with both edge deletions and additions ($\epsilon_1 = 0.3, \epsilon_2 = 0.1$). Each state depicts the progressive impacts of the perturbations.

In this context, the impact of the perturbation on the degree of a given node $u \in \mathcal{V}$ can be computed as follows. The effect of edge deletion is represented by $(-\mathbf{\Delta}_{\epsilon_1} \circ \mathbf{A})_u$, where each non-zero element in $\mathbf{A}_u$ has a probability of ϵ_1 being deleted. Thus, the total number of deleted edges δ_u^- is the sum of d_u independent and identically distributed (i.i.d.) Bernoulli random variables, each with a probability of ϵ_1. Similarly, the effect of edge addition is denoted by $(\mathbf{\Delta}_{\epsilon_2} \circ (\mathbf{1}_{N \times N} - \mathbf{A}))_u$, and the total number of added edges δ_u^+ is the sum of d_u^* i.i.d. Bernoulli random variables, each with a probability of ϵ_2, where $d_u^* = N - d_u - 1$. Hence, we can express the number of deleted edges δ_u^- and the number of added edges δ_u^+ as following binomial distributions:

$$\delta_u^- \sim \text{Bin}(d_u, \epsilon_1), \ \delta_u^+ \sim \text{Bin}(d_u^*, \epsilon_2), \tag{7}$$

where $\text{Bin}(n, p)$ represents a binomial distribution with parameters n and p.

IV. Expected Bound for GSO Error

A. Error Bound for Unnormalized GSO Using ℓ_1 Norm

Building on the foundation laid by the discussion of graph structure perturbations and the proposed error model, we now outline the primary theoretical contributions of this study. Our focus here is to detail the probabilistic bounds that help quantify the sensitivity of the GSO to graph structure perturbations. We examine the case where the adjacency matrix serves as the GSO, implying $\hat{\mathbf{S}} = \hat{\mathbf{A}}$ and $\mathbf{S} = \mathbf{A}$. The error model derived in (6) can be expressed as

$$\mathbf{E} = \hat{\mathbf{A}} - \mathbf{A} = -\mathbf{\Delta}_{\epsilon_1} \circ \mathbf{A} + \mathbf{\Delta}_{\epsilon_2} \circ (\mathbf{1}_{N \times N} - \mathbf{A}). \tag{8}$$

We can link the change in degree with the ℓ_1 norm of error term in (8) as

$$\|\mathbf{E}\|_1 = \max_{u \in \mathcal{V}} \|\mathbf{E}_u\|_1, \tag{9}$$

where

$$Y_u \triangleq \|\mathbf{E}_u\|_1 = |\mathcal{D}_u| + |\mathcal{A}_u| = \delta_u^- + \delta_u^+. \tag{10}$$

Let $Y \triangleq \max_{u\in\mathcal{V}} Y_u$. Since δ_u^- and δ_u^+ are independent random variables, it is not appropriate to give deterministic upper bounds. Instead, we present expected value bounds, which are better suited for analyzing the degree changes of nodes given the probabilistic nature of the model. Our goal is to derive a closed-form expression for the expectation of the maximum node degree error, i.e.,

$$\mathbb{E}[\|\mathbf{E}\|_1] = \mathbb{E}[\max_{u\in\mathcal{V}} \|\mathbf{E}_u\|_1]. \tag{11}$$

The probability mass function (PMF) of Y_u can be found by convolving the PMFs of δ_u^- and δ_u^+, which are independent random variables. Following binomial distributions in (7), we can obtain the following PMFs

$$\mathrm{Pr}_{\delta_u^-}(k) = \binom{d_u}{k} \epsilon_1^k (1-\epsilon_1)^{d_u-k},\ k = 0,\dots,d_u, \tag{12}$$

$$\mathrm{Pr}_{\delta_u^+}(k) = \binom{d_u^*}{k} \epsilon_2^k (1-\epsilon_2)^{d_u^*-k},\ k = 0,\dots,d_u^*, \tag{13}$$

where $d_u^* = N - d_u - 1$, $\mathrm{Pr}_{\delta_u^-}(k)$ and $\mathrm{Pr}_{\delta_u^+}(k)$ represent the probabilities of δ_u^- and δ_u^+ taking the value k, respectively. Then, the PMF of Y_u can be computed as

$$\begin{aligned}\mathrm{Pr}_{Y_u}(k) &= \sum_{i=\max\{0,k-d_u^*\}}^{\min\{k,d_u\}} \mathrm{Pr}_{\delta_u^-,\delta_u^+}(i,k-i) \\ &= \sum_{i=\max\{0,k-d_u^*\}}^{\min\{k,d_u\}} \mathrm{Pr}_{\delta_u^-}(i)\mathrm{Pr}_{\delta_u^+}(k-i),\end{aligned} \tag{14}$$

where $k = 0,\dots,N-1$. Using (14), the cumulative distribution function (CDF) of Y is computed as

$$\begin{aligned}\mathrm{F}_Y(k) &= \Pr(Y \le k) = \Pr(\max(Y_1,\dots,Y_N) \le k) \\ &= \Pr(Y_1 \le k,\dots,Y_N \le k) = \prod_{u=1}^{N} \Pr(Y_u \le k).\end{aligned} \tag{15}$$

Given that Y_u for $u \in \mathcal{V}$ are i.i.d. and for $k = 1,\dots,N-1$, the CDFs for Y and Y_u are as follows

$$\mathrm{F}_Y(k) = \prod_{u=1}^{N} \mathrm{F}_{Y_u}(k), \quad \mathrm{F}_{Y_u}(k) = \sum_{j=0}^{k} \mathrm{Pr}_{Y_u}(j). \tag{16}$$

With the PMF of Y taking on a specific value k being $\mathrm{Pr}_Y(k) = \mathrm{F}_Y(k) - \mathrm{F}_Y(k-1)$, the expectation of Y can be represented as

$$\mathbb{E}[Y] = \sum_{k=1}^{N-1} k\mathrm{Pr}_Y(k) = \sum_{k=1}^{N-1} k\left[\mathrm{F}_Y(k) - \mathrm{F}_Y(k-1)\right], \tag{17}$$

which provides a closed-form expression for $\mathbb{E}[Y] = \mathbb{E}[\|\mathbf{E}\|_1]$. The variance of Y can also be given as

$$\mathrm{Var}[Y] = \mathrm{Var}[\|\mathbf{E}\|_1] = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2, \tag{18}$$

where $\mathbb{E}[Y^2] = \sum_{k=1}^{N-1} k^2 \mathrm{Pr}_Y(k)$.

B. Bridging ℓ_1 and ℓ_2 Norms in GSO Analysis

In the analysis of graph-structured data, the spectral norm (ℓ_2 norm), is often employed to quantify the graph spectral error. While [31] did furnish a spectral error bound for the GSO, the need for a more refined and interpretable bound persists to enable more comprehensive analyses. Following the approach of [23], this study uses the ℓ_1 norm and assumes that the error matrix $\mathbf{E}$ is fixed. The proposed approach of bounding $\|\mathbf{E}\|$ is based on assumptions of an undirected graph and perturbation $\mathbf{E} = \mathbf{E}^\top$. Using inequalities $\|\mathbf{E}\|^2 \le \|\mathbf{E}\|_1\|\mathbf{E}\|_\infty$ [35, Sec. 2.3.3] and the fact that in our case $\|\mathbf{E}\|_1 = \|\mathbf{E}\|_\infty$, the ℓ_2 norm can be bounded by the ℓ_1 norm

$$\|\mathbf{E}\| \le \|\mathbf{E}\|_1 = \max_{u\in\mathcal{V}} \|\mathbf{E}_u\|_1. \tag{19}$$

The entries in the error matrix $\mathbf{E}$ of (8) are random variables. As such, it is challenging to derive a deterministic bound for (19) that is both tight and generalizable. In contrast, an expected bound

$$\mathbb{E}[\|\mathbf{E}\|] \le \mathbb{E}[\|\mathbf{E}\|_1] = \mathbb{E}[\max_{u\in\mathcal{V}} \|\mathbf{E}_u\|_1], \tag{20}$$

provides a more reasonable estimate of the true behavior of the error matrix, as it takes into account the distribution of the random variables, as well as the structural changes of the perturbed graph. Thus, we have the following theorem.

Theorem 1: In the context of the probabilistic error model (8), let GSO be adjacency matrix $\mathbf{S} = \mathbf{A}$, and perturbed GSO be $\hat{\mathbf{S}} = \hat{\mathbf{A}}$, then, a closed-form expression for the upper bound on the expectation of the GSO distance is given by

$$\mathbb{E}\left[d(\hat{\mathbf{S}},\mathbf{S})\right] \le \mathbb{E}[Y], \tag{21}$$

where $\mathbb{E}[Y]$ is computed using (17), (16), and (14).

Theorem 1 provides a closed-form expression for the upper bound, which are explicitly dependent on the parameters (ϵ_1,ϵ_2) of the probabilistic error model in (8). Using a loose upper bound proposed in [36], we can bound (21) as

$$\begin{aligned}\mathbb{E}[Y] \le\ & \max_{1\le u\le N}(d_u\epsilon_1 + d_u^*\epsilon_2) \\ & + \sqrt{\frac{N-1}{N}\sum_{u=1}^{N}(d_u\epsilon_1(1-\epsilon_1) + d_u^*\epsilon_2(1-\epsilon_2))}.\end{aligned} \tag{22}$$

We note that (22) showcases how our bound in Theorem 1 is parameterized by the probabilities of adding and deleting edges. Thus, Theorem 1 precisely captures the resulting structural changes induced by the probabilistic error model, unlike the generic spectral bound in [31], which overlooks specific structural changes on the perturbed GSO.

Remark 1 (Why not use ℓ_2 norm?): The spectral bounds derived using the ℓ_2 norm, as presented in [31], cannot fully capture the specific structural changes to the GSO from perturbations, especially in graphs with unique properties like degree distribution or sparsity. Focused on worst-case scenarios, these bounds lead to overestimations, rendering them looser and less applicable to particular graph types. The ℓ_1 norm is preferred

over the ℓ_2 norm for providing an upper bound because it reveals the impact of structural changes denoted by $\mathbf{\Delta}_{\epsilon_1}$ and $\mathbf{\Delta}_{\epsilon_2}$ in (8), whereas the ℓ_2 norm absorbs these structural changes into the overall spectral change, making it more challenging to derive a tight bound.

C. Error Bound for Normalized GSO

In this context, the GSO is considered as the normalized version of the adjacency matrix, i.e., $\mathbf{S} = \mathbf{A}_{\text{n}}$. The entries of the normalized adjacency matrix are as follows, $[\mathbf{A}_{\text{n}}]_{u,v} = \frac{1}{\sqrt{d_u d_v}}$ if $(u, v) \in \mathcal{E}$, and $[\mathbf{A}_{\text{n}}]_{u,v} = 0$ if $(u, v) \notin \mathcal{E}$. In [23], a closed form for $\|\mathbf{E}_u\|_1$ is proposed

$$\|\mathbf{E}_u\|_1 = \sum_{v\in\mathcal{D}_u} \frac{1}{\sqrt{d_u d_v}} + \sum_{v\in\mathcal{A}_u} \frac{1}{\sqrt{\hat{d}_u \hat{d}_v}} + \sum_{v\in\mathcal{R}_u} \left| \frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{\hat{d}_u \hat{d}_v}} \right|, \tag{23}$$

where $\hat{d}_u$ and $\hat{d}_v$ denote the degrees of node u and v after perturbation. However, the assumption in [23] states that the degree alteration $\hat{d}_v$ should not exceed twice the initial degree, i.e., $\hat{d}_v \leq 2d_v, v \in \{\mathcal{N}_u \cup u\}$. This restriction is not needed in our work. Following the error model in (6), this limitation could easily be breached with an increased probability of edge addition ϵ_2. We start with the following lemma.

Lemma 1: Let $\mathbf{E}_u$ be defined as in (23), then its ℓ_1 norm is bounded by a random variable Z_u

$$\|\mathbf{E}_u\|_1 \leq Z_u = Z_{u,1} + Z_{u,2}, \tag{24}$$

where Z_u is defined as the sum of $Z_{u,1} = \sqrt{d_u/\tau_u}$ and $Z_{u,2} = \sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u} \frac{1}{\sqrt{(d_u+\delta_u^+-\delta_u^-)(d_v+\delta_v^+-\delta_v^-)}}$, d_u is the degree of node u, τ_u is the minimum degree of neighboring nodes of u, and $\delta_u^-, \delta_u^+, \delta_v^-, \delta_v^+$ are random variables with binomial distributions as $\delta_u^- \sim \text{Bin}(d_u, \epsilon_1), \delta_u^+ \sim \text{Bin}(d_u^*, \epsilon_2), \delta_v^- \sim \text{Bin}(d_v, \epsilon_1), \delta_v^+ \sim \text{Bin}(d_v^*, \epsilon_2)$ for $u \in \mathcal{V}$ and $v \in \mathcal{A}_u \cup \mathcal{R}_u$, where $d_u^* = N - d_u - 1$ and $d_v^* = N - d_v - 1$.

Proof: See Appendix A.

Let

$$Z \triangleq \max_{u\in\mathcal{V}} Z_u, \tag{25}$$

and note that Z_u and Z are discrete random variables. While the binomial random variables and degrees in the expression for Z are assumed to be i.i.d., the inherent nonlinearity and high-dimensionality in the function, along with the complexity introduced by the maximization operation over all nodes, pose challenges for deriving an analytical expression for $\mathbb{E}[Z]$. Furthermore, the expectation of a maximum of random variables often lacks a simple closed form with only bounds often being derivable, not the exact value. On the other hand, Monte Carlo simulations provide an efficient alternative for estimating $\mathbb{E}[Z]$, which is given as

$$\mu_Z \triangleq \mathbb{E}[Z] \approx \frac{1}{N_{\text{samp}}} \sum_{i=1}^{N_{\text{samp}}} Z_{(i)} = \hat{\mu}_Z, \tag{26}$$

where $Z_{(i)}$ represents the outcome from the i-th Monte Carlo trial. Thus, for the normalized GSO, we have the following proposition as the counterpart of Theorem 1.

Proposition 1: In the context of the probabilistic error model (8), let GSO be normalized adjacency matrix $\mathbf{S} = \mathbf{A}_{\text{n}}$, and perturbed GSO being $\hat{\mathbf{S}} = \hat{\mathbf{A}}_{\text{n}}$. Then, an upper bound on the expectation of the GSO distance is given by

$$\mathbb{E}\left[d(\hat{\mathbf{S}}, \mathbf{S})\right] \leq \mathbb{E}[Z], \tag{27}$$

where $\mathbb{E}[Z]$ is computed using (26), (25), and Lemma 1.

The upperbound provided in Proposition 1 focuses specifically on normalized adjacency matrices. This result complements the analysis for the unnormalized case. We note that the bound for normalized GSO is not an approximation or an empirical estimation; it presents a theoretical upperbound. The only difference between the bound in Proposition 1 and the bound in Theorem 1 is the computation. As for the bound in Theorem 1 (unnormalized case), $\mathbb{E}[Y]$ has a closed-form expression; while for computing the bound in Proposition 1 (normalized case) $\mathbb{E}[Z]$, we use Monte Carlo simulations.

V. GCNN Sensitivity

A. Graph Filter Sensitivity Analysis

The sensitivity of graph filters is a critical aspect that follows logically from the preceding discussion on the expected bounds of GSO errors. Having extensively delved into the properties of GSO perturbations, we now turn our attention to the graph filters. Graph filters, being polynomials of GSOs, inherit the perturbations in the graph structure, manifesting as variations in filter responses.

The sensitivity of a graph filter to perturbations in the GSO is captured by the theorem below, which establishes a bound on the error in the graph filter response due to perturbations in the GSO and the filter coefficients.

Theorem 2 (Graph filter sensitivity): Let $\mathbf{S}$ and $\hat{\mathbf{S}}$ be the GSO for the true graph $\mathcal{G}$ and the perturbed graph $\hat{\mathcal{G}}$, respectively. The distance between polynomial graph filters $\mathbf{h}(\mathbf{S}) = \sum_{k=0}^{K} h_k \mathbf{S}^k$ and $\mathbf{h}(\hat{\mathbf{S}}) = \sum_{k=0}^{K} h_k \hat{\mathbf{S}}^k$ is defined as

$$d\left(\mathbf{h}(\hat{\mathbf{S}}), \mathbf{h}(\mathbf{S})\right) = \|\mathbf{h}(\hat{\mathbf{S}}) - \mathbf{h}(\mathbf{S})\|. \tag{28}$$

The expectation of filter distance (28) is bounded as

$$\mathbb{E}\left[d\left(\mathbf{h}(\hat{\mathbf{S}}), \mathbf{h}(\mathbf{S})\right)\right] \leq \sum_{k=1}^{K} k|h_k| \left(\lambda_k \mathbb{E}[\|\mathbf{E}\|] + \zeta_k\right), \tag{29}$$

where $\lambda_k \triangleq \mathbb{E}[\lambda^{k-1}]$, $\zeta_k \triangleq \text{Cov}[\|\mathbf{E}\|, \lambda^{k-1}]$, and $\lambda = \max\{\|\hat{\mathbf{S}}\|, \|\mathbf{S}\|\}$ denotes the largest of the maximum singular values of two GSOs.

Proof: See Appendix B.

Theorem 2 reveals that the expected graph filter distance is linearly bounded by the expected GSO distance, $\mathbb{E}[\|\mathbf{E}\|]$, if the sufficient condition $\lambda = \|\mathbf{S}\|$ is met. This bound is influenced by: the filter degree K, the maximum singular value λ of GSOs, and the filter coefficients $\{h_k\}_{k=1}^K$. The theorem indicates that higher order graph filters are likely to exhibit greater instability. In

Section VI-B, we present a supporting experiment, specifically for low-pass graph filters with the unnormalized GSO, $\mathbf{S} = \mathbf{A}$.

B. GCNN Sensitivity Analysis

Based on the sensitivity analysis of graph filter, we extend this study to the sensitivity analysis of the general GCNN. Instead of meticulously quantifying the specifics of each perturbed graph, we propose a probabilistic boundary that captures the potential magnitude of graph perturbations and more insightful assessment of the system's sensitivity to graph perturbations. We present the following theorem to exemplify this approach, encapsulating the sensitivity of a general GCNN to GSO perturbations.

Theorem 3 (GCNN Sensitivity): For a general GCNN under the probabilistic error model (8), the expected difference of outputs at the final layer L is given as

$$\mathbb{E}\left[\left\|\hat{\mathbf{X}}_L - \mathbf{X}_L\right\|\right] \leq C_{\sigma_L} B_L \mathbb{E}\left[\|\mathbf{E}\|\right] + C_{\sigma_L} D_L, \tag{30}$$

where C_{σ_ℓ} represents the Lipschitz constant for the nonlinear activation function used at layer ℓ, for $\ell = 1, \ldots, L$, B_ℓ and D_ℓ for $\ell = 1$ and then for $\ell = 2, \ldots, L$ are defined as follows

$$\begin{aligned} B_1 &= \sum_{k=1}^{K} k\lambda_k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\|, D_1 = \sum_{k=1}^{K} k\zeta_k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\|, \\ B_\ell &= \sum_{k=1}^{K} \left(\lambda_{k+1} C_{\sigma_{\ell-1}} B_{\ell-1} + k\lambda_k \|\mathbf{X}_{\ell-1}\|\right) \|\mathbf{H}_{\ell k}\|, \\ D_\ell &= \sum_{k=1}^{K} \left(\mu_{k,\ell-1} + \lambda_k C_{\sigma_{\ell-1}} D_{\ell-1} + k\zeta_k \|\mathbf{X}_{\ell-1}\|\right) \|\mathbf{H}_{\ell k}\|, \end{aligned} \tag{31}$$

with constant $\mu_{k,\ell-1} \triangleq \sqrt{\text{Var}[\|\hat{\mathbf{X}}_{\ell-1} - \mathbf{X}_{\ell-1}\|]\text{Var}[\lambda^k]}$, and λ_k and ζ_k in Theorem 2, for $k = 1, \ldots, K$.

Proof: See Appendix C.

In Theorem 3, we use recursive bounds containing inter-layer features to simplify the formulation. Note that these inter-layer features $\{\mathbf{X}_{\ell-1}, \hat{\mathbf{X}}_{\ell-1}\}_{\ell=2}^{L}$ can be explicitly computed by the initial input feature $\mathbf{X}_0$, both original and perturbed GSOs $(\mathbf{S}, \hat{\mathbf{S}})$, GCNN parameters (number of layers L and graph shift K, network's learned weights $\{\mathbf{H}_{\ell k}\}$, and activation functions $\sigma(\cdot)$). The derivation process employs induction. For the first layer $\ell = 1$, we have $\mathbf{X}_1 = \sigma_1(\sum_{k=1}^{K} \mathbf{S}^k \mathbf{X}_0 \mathbf{H}_{1k})$ and $\hat{\mathbf{X}}_1 = \sigma_1(\sum_{k=1}^{K} \hat{\mathbf{S}}^k \mathbf{X}_0 \mathbf{H}_{1k})$; for the second layer $\ell = 2$, the features are $\mathbf{X}_2 = \sigma_2(\sum_{k=1}^{K} \mathbf{S}^k \mathbf{X}_1 \mathbf{H}_{2k})$ and $\hat{\mathbf{X}}_2 = \sigma_2(\sum_{k=1}^{K} \hat{\mathbf{S}}^k \hat{\mathbf{X}}_1 \mathbf{H}_{2k})$; by induction, for the $\ell - 1$th layer, we have

$$\begin{aligned} \mathbf{X}_{\ell-1} &= \sigma_\ell \left(\sum_{k=1}^{K} \mathbf{S}^k \mathbf{X}_{\ell-2} \mathbf{H}_{\ell-1,k} \right), \\ \hat{\mathbf{X}}_{\ell-1} &= \sigma_\ell \left(\sum_{k=1}^{K} \hat{\mathbf{S}}^k \hat{\mathbf{X}}_{\ell-2} \mathbf{H}_{\ell-1,k} \right). \end{aligned} \tag{32}$$

Theorem 3 forms the bedrock of our analysis, quantifying how GCNNs respond to graph perturbations, which is described by a linear relationship at each layer. The sensitivity of multilayer GCNN to perturbations can be represented by a recursion of linearity. For multilayer GCNN, its expected output difference is controlled by: *(i)* the input feature, *(ii)* the GSO, error model parameters, *(iii)* Lipschitz constants of activation functions, and *(iv)* GCNN weights. We note that, choosing activation functions with more conservative Lipschitz constants can possibly improve the stability of GCNNs by imposing more constraints on the recursion. However, this may suppress the performance of a neural network, as noted in [37]. Our sensitivity analysis framework is generic, allowing for simplifications such as assuming a unit Lipschitz constant and normalized input features, as suggested in [22]. However, these simplifications do not indicate that the GCNN sensitivity is unaffected by the Lipschitz constant or input features. This layered analysis also enables an understanding of how perturbations propagate through GCNN layers, impacting the overall performance. Additionally, Theorem 3 does not restrict the scale of graph perturbations, which is a typical restriction in the existing literature.

Within the evasion attack context, where the focus is on learned representations, we demonstrate the following property: given that the GSO error is bounded as in Theorem 1 and Proposition 1, the linear bound of each layer of GCNN (illustrated in Subsection VI-C1) permits the network's stability against perturbation as long as the graph error remains within the bound. In Subsection VI-C2, we show that multilayer GCNN is stable by showing its finite responses to large scale perturbations, even under notable declines in accuracy.

C. Specifications for GCNN Variants

Building upon sensitivity analysis Theorem 3, our discussion now evolves towards two specific GCNN variants - GIN [6] and SGCN [7], [8]. They apply different GSOs for feature propagation. In GIN, the GSO for each layer is chosen as a partially augmented unnormalized adjacency matrix; in SGCN, the GSO is chosen as a normalized augmented adjacency matrix. This choice is made to align with the discussions on tight GSO bounds in Section IV. By focusing on GIN and SGCN, we are essentially extending our theoretical understanding to practical and real-world applications.

1) Specification for GIN: The GIN is designed to capture the node features and the graph structure simultaneously. The primary intuition behind GIN is to learn a function of the feature information from both the target node and its neighbors, which is related to the Weisfeiler-Lehman (WL) graph isomorphism test [38]. The chosen GSO for GIN is $\mathbf{S} = \mathbf{A} + (1 + \varepsilon)\mathbf{I}$, where the learnable parameter ε preserves the distinction between nodes in the graph that are connected differently, and prevents GIN from reducing to a WL isomorphism test.

Given the GSO above, only the first order term with $K = 1$ in (1) is kept, and the intermediate output of such graph filter is $\mathbf{y} = \mathbf{S}\mathbf{x}$. A node Multilayer Perceptron (MLP) $\mathbf{h}_{\boldsymbol{\Theta}}$ is then applied to the filter's output as $\mathbf{h}_{\boldsymbol{\Theta}}(\mathbf{y})$. Assuming the inner MLP has two layers in each GIN layer, a single-layer GIN ($L = 1$)

can be represented as

$$\mathbf{X}_L = \sigma_{L2}(\sigma_{L1}(\mathbf{S}\mathbf{X}_{L-1}\mathbf{W}_{L1} + \mathbf{B}_{L1})\mathbf{W}_{L2} + \mathbf{B}_{L2}), \quad (33)$$

where $(\mathbf{W}_{L1}, \mathbf{B}_{L1}, \sigma_{L1}(\cdot))$ are weight matrix, bias matrix, and nonlinearity function in the first layer of the MLP, and $(\mathbf{W}_{L2}, \mathbf{B}_{L2}, \sigma_{L2}(\cdot))$ are weight matrix, bias matrix, and nonlinearity function in the second layer of the MLP. Then, we provide the following corollary.

Corollary 1 (The sensitivity of single-layer GIN): For the single-layer GIN ($L = 1$) in (33) under the probabilistic error model (8), the expected difference of outputs because of GSO perturbations is given as

$$\mathbb{E}\left[\|\hat{\mathbf{X}}_L - \mathbf{X}_L\|\right] \leq \xi \mathbb{E}\left[\|\mathbf{E}\|\right], \quad (34)$$

with constant

$$\xi = C_{\sigma_{L2}} C_{\sigma_{L1}} \|\mathbf{W}_{L2}\| \|\mathbf{W}_{L1}\| \|\mathbf{X}_{L-1}\|, \quad (35)$$

where $\mathbf{X}_{L-1} = \mathbf{X}_0$ is the input feature.

Proof: See Appendix D.

Corollary 1 shows a linear dependency between the output difference of a single-layer GIN and GSO perturbations. In GIN, node vector transformations by MLP contribute significantly to network's expressivity. Under evasion attacks, with Corollary 1, the analysis of these transformed node representations is straightforward.

2) Specification for SGCN: The SGCN is a streamlined model, developed by aiming to simplify a multilayered GCNN through the utilization of an affine approximation of graph convolution filter and the elimination of intermediate layer activation functions. The GSO chosen for SGCN is $\mathbf{S} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the augmented adjacency matrix and $\tilde{\mathbf{D}}$ is the corresponding degree matrix of the augmented graph.

Given the normalized augmented GSO, the node degrees $d_u, u = 1, \ldots, N$ are redefined based on the augmented GSO, specifically, they are incremented by 1 compared to their values in the non-augmented version. This streamlined model simplifies the structure of a vanilla GCN [5] by retaining a single layer and the Kth order GSO in (1), so the output of the filter is $\mathbf{y} = h_K \mathbf{S}^K \mathbf{x}$. Note that for a SGCN, the maximum number of layers is $L = 1$. Consequently, the output of a single-layer SGCN using a linear logistic regression layer is represented as

$$\mathbf{X}_L = \sigma_L(\mathbf{S}^K \mathbf{X} \mathbf{H}_K), \quad (36)$$

and thus, we can easily give the following corollary.

Corollary 2 (The sensitivity of SGCN): For the SGCN in (36) under the probabilistic error model (8), the expected difference of outputs because of GSO perturbations is given as

$$\mathbb{E}\left[\|\hat{\mathbf{X}}_L - \mathbf{X}_L\|\right] \leq C_{\sigma_L} B_L \mathbb{E}\left[\|\mathbf{E}\|\right] + C_{\sigma_L} D_L, \quad (37)$$

where $B_L = \lambda_K \|\mathbf{X}\| \|\mathbf{H}_K\|$, $D_L = K\zeta_K \|\mathbf{X}\| \|\mathbf{H}_K\|$, λ_K and ζ_K are defined in Theorem 3.

With Corollary 2, we conclude that the sensitivity analysis for SGCN is a specification for the general form of a multilayer GCNN.

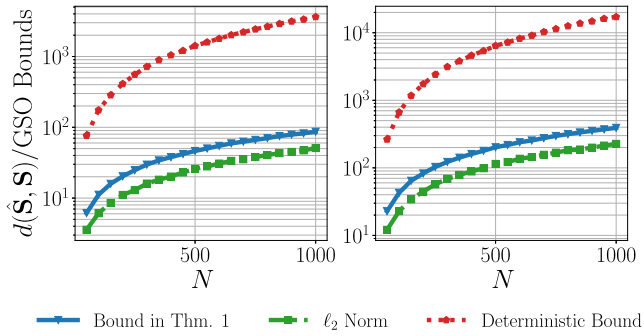


Fig. 2. Comparative analysis of our bound in Theorem 1, the deterministic bound in Theorem 2 of [31], and the empirical GSO distance in ℓ_2 norm.

VI. Numerical Experiments

A. Theoretical GSO Bound Corroboration

1) Synthetic Graph: We consider a two-group planted partition model (PPM), which is a special case of the stochastic block model. Parameters are set with in-group probability to $p_{\text{in}} = 0.8$, and between-group probability to $p_{\text{bet}} = 0.5$. The GSO is set as the unnormalized adjacency matrix $\mathbf{S} = \mathbf{A}$. We perturb the PPM graph using the probabilistic error model (6) with two scales of perturbation budgets:

- *Small-scale perturbation (see Fig. 2, left panel):* With $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.01$, the graph is slightly altered, preserving its fundamental structure.
- *Large-scale perturbation (see Fig. 2, right panel):* With $\epsilon_1 = 0.5$ and $\epsilon_2 = 0.1$, the graph is under significant structural changes.

We carry out 101 Monte Carlo trials for varying graph sizes (ranging from 50 to 1000, in 50-node increments). These simulations evaluate the expected bound from Theorem 1 and the deterministic bound from [31, Theorem 2] in relation to graph size. Comparisons with empirical GSO distances (5), calculated using the ℓ_2 norm, reveal that our expectation bound is consistently tighter than the deterministic counterpart from [31]. This difference arises due to the consideration of degree changes and the probabilistic nature of our bound, as opposed to the worst-case scenario focus of the deterministic bound. Another observation is the increased bound magnitude correlating with higher perturbation budgets, as depicted in Fig. 2. Both bounds remain valid, even in high perturbation scenarios, underscoring the robustness of our theoretical frameworks.

2) Real-Life Graph: We utilize the undirected Cora citation graph [39], which comprises $N = 2708$ nodes, and $C = 7$ classes. Assuming the undirected nature of the underlying graph, we modify the original Cora graph from a directed to an undirected one. The undirected Cora graph has $|\mathcal{E}| = 5278$ edges. We ascertain the evolution of our theoretical bounds against an increase in edge deletion probability ϵ_1 and edge addition probability ϵ_2. These alterations are systematically tracked along with using the ℓ_1 and ℓ_2 norms of the discrepancy between the original and perturbed graphs.

The range of ϵ_1 and ϵ_2 is set within $[3 \times 10^{-2}, 3 \times 10^{-1}]$, increasing in steps of 3×10^{-2}. In each step, we compute

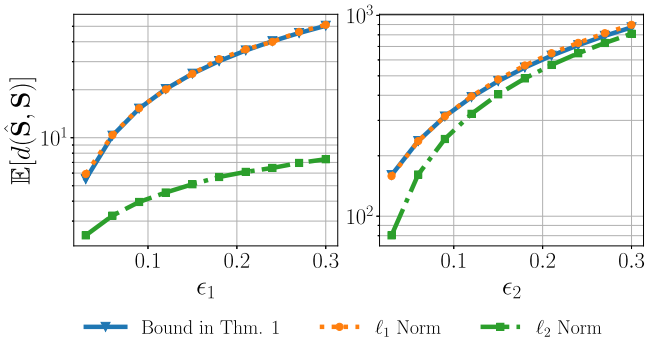


Fig. 3. Theoretical (bound in Thm. 1) and empirical bounds (ℓ_1 and ℓ_2 norms) for the perturbed Cora graph with $\mathbf{S} = \mathbf{A}$. Left panel: varying ϵ_1 with fixed $\epsilon_2 = 0$. Right panel: varying ϵ_2 with fixed $\epsilon_1 = 0.5$.

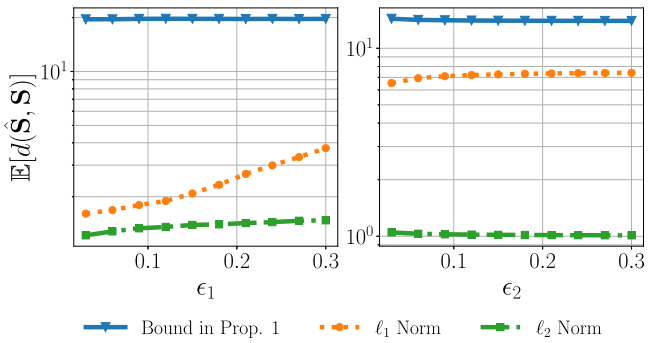


Fig. 4. Theoretical (bound in Prop. 1) and empirical bounds (ℓ_1 and ℓ_2 norms) for the perturbed Cora graph with $\mathbf{S} = \mathbf{A}_{\text{n}}$, under identical (ϵ_1, ϵ_2) settings as Fig. 3.

the ℓ_1 and ℓ_2 norms of the difference between the original and perturbed adjacency matrices. We then compare these empirical results with the theoretical bounds provided in Theorem 1 and Proposition 1. In Fig. 3, with the GSO as the unnormalized adjacency matrix $\mathbf{S} = \mathbf{A}$, two distinct scenarios are presented: varying ϵ_1 with $\epsilon_2 = 0$ (left panel), and varying ϵ_2 with $\epsilon_1 = 0.5$ (right panel). Through 101 Monte Carlo trials, the theoretical bound closely aligns with the empirical ℓ_1 norm, particularly in scenarios where increased ϵ_2 leads to denser graphs. This trend suggests that enhanced precision of the bounds as graph densities shift from sparse to dense.

In Fig. 4, employing the normalized adjacency matrix $\mathbf{S} = \mathbf{A}_{\text{n}}$ as the GSO, a similar analysis is conducted. In the left panel, an increase in ℓ_1 and ℓ_2 norm bounds is observed under rising error, and Proposition 1 gives a stable upper bound. However, the accuracy of the bound is comparatively less satisfactory in the normalized case. The right-hand case illustrates a stable empirical ℓ_2 norm with an increasing number of edges, while the ℓ_1 norm and our bound present slight increases and decreases, respectively. These observations can be attributed to the following factors: *(i)* the normalization operation keeps the adjacency matrix operator norm around 1; *(ii)* an increased number of edges raises the ℓ_1 norm; *(iii)* increases in the denominator in Lemma 1 result in a general decrease in the bound.

B. GF Sensitivity Test

In this experiment, we evaluate the sensitivity of GF to the probabilistic error model. We employ an ER graph with $N = 100$ nodes and a connection probability of 0.1 as the baseline graph. The GSO is set as the unnormalized adjacency matrix $\mathbf{S} = \mathbf{A}$. Our focus is on the relationship between filter distance and the bound in Theorem 2 for low pass GFs of orders $K = 1, 2, 3$. The findings are presented in Figs. 5 and 6.

In Fig. 5, the edge addition probability is fixed as $\epsilon_2 = 0.05$ and the edge deletion probability ϵ_1 varies among [0.1,0.2,0.3]. Over 101 Monte Carlo trials, we plot the empirical GF distances $d(\mathbf{h}(\hat{\mathbf{S}}, \mathbf{S}))$ alongside the corresponding GSO distances $d(\hat{\mathbf{S}}, \mathbf{S}) = \|\mathbf{E}\|$ as scatter plots. These empirical GF distances demonstrate the linear scaling with the bounds in Theorem 2, depicted as solid lines. It is noted that the tightness of these bounds decreases with an increase in the GF order. The primary aim of this analysis is to confirm the linear relationship in Theorem 2.

In Fig. 6, the expected output differences of GFs $\mathbb{E}[d(\mathbf{h}(\hat{\mathbf{S}}), \mathbf{h}(\mathbf{S}))]$ with orders $K = 1, 2, 3$ are plotted against the expected GSO differences $\mathbb{E}[d(\hat{\mathbf{S}}, \mathbf{S})]$ and the bound in Theorem 1. Over 101 Monte Carlo trials with perturbation probabilities $\epsilon_1 \in [0, 0.3]$ and $\epsilon_2 \in [0, 0.05]$, the left panel shows that output differences increase with the GF order. The right panel confirms that the bound $\mathbb{E}[Y]$ captures trends similar to the empirical expectation of GSO distance, corroborating Theorem 1. This suggests that for small, sparsely connected graphs, the sensitivity of a low pass GF to perturbations intensifies as its order increases.

C. GCNN Sensitivity Test

1) Linearity Corroboration: The experimental validation of Theorem 3 is conducted using GIN (Corollary 1) and SGCN (Corollary 2). We note that Corollary 1 is only applicable for the single-layer GIN ($L = 1$). For the multi-layer GIN, our experiments show the recursion of linearity indicated in Theorem 3 empirically (see left panel of Fig. 7). These experiments are carried out on the Cora citation dataset, as discussed in Section VI-A, to assess the sensitivity of GIN and SGCN to perturbed GSOs under evasion attacks.

In Fig. 7, for GIN (left panel), each layer comprises 16 hidden features. GIN variants with 1, 2, and 3 layers differ only in the number of cascaded graph filters with MLPs. We investigate the correlation between empirical GIN output differences and GSO distances. The edge deletion probability, ϵ_1, is varied within $[5 \times 10^{-2}, 3 \times 10^{-1}]$ in increments of 5×10^{-2}, while the edge addition probability is fixed as $\epsilon_2 = 1 \times 10^{-4}$. The results, categorized by edge deletion probability ϵ_1, are obtained from 101 Monte Carlo trials, computing pairs of bounds and GIN output differences. For SGCN (right panel), we examine networks of orders $K = [1, 2, 3]$ using a similar approach. Empirical observations for $L = 1, 2, 3$ and $K = 1, 2, 3$ in GIN and SGCN demonstrate a linear correlation between output differences and GSO distances, corroborating the theoretical frameworks in Corollary 1 and Corollary 2.

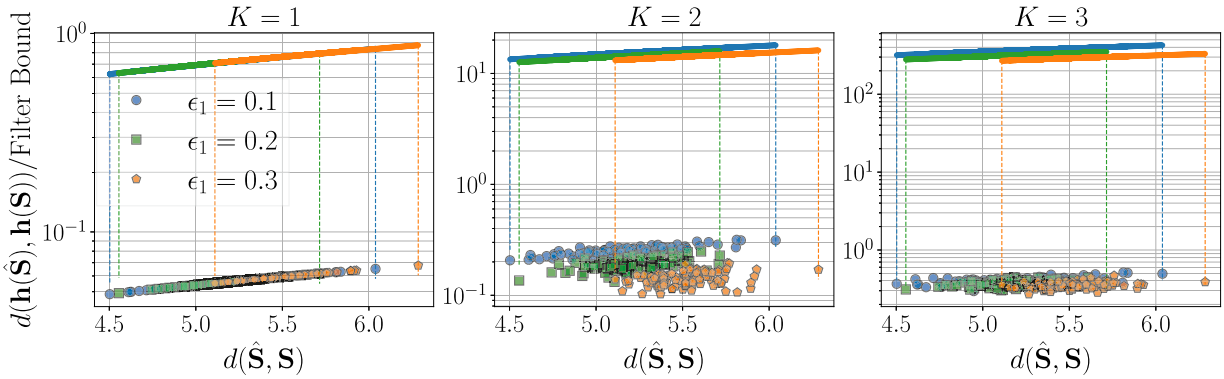


Fig. 5. Comparison of Theorem 2 bounds (solid lines) and empirical GF distances (scatter points) with fixed $\epsilon_2 = 0.05$ and varying ϵ_1 in [0.1,0.2,0.3].

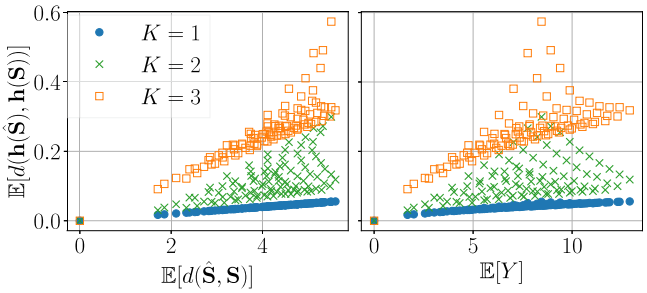


Fig. 6. Expected GF output differences under increasing GF order and perturbation budget, illustrating intensified sensitivity along increased GF order and the alignment of Theorem 1 bound with empirical GSO distance trends.

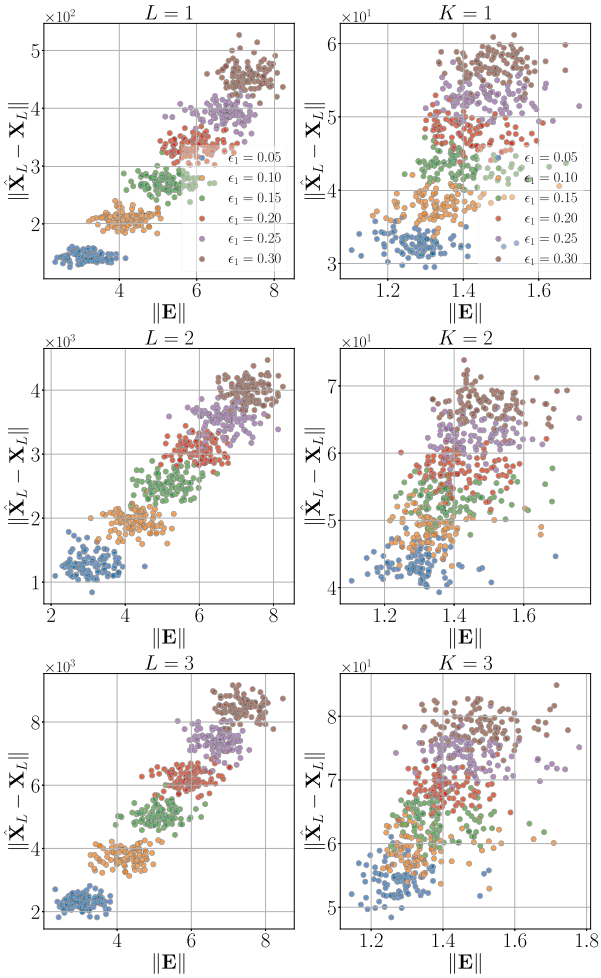


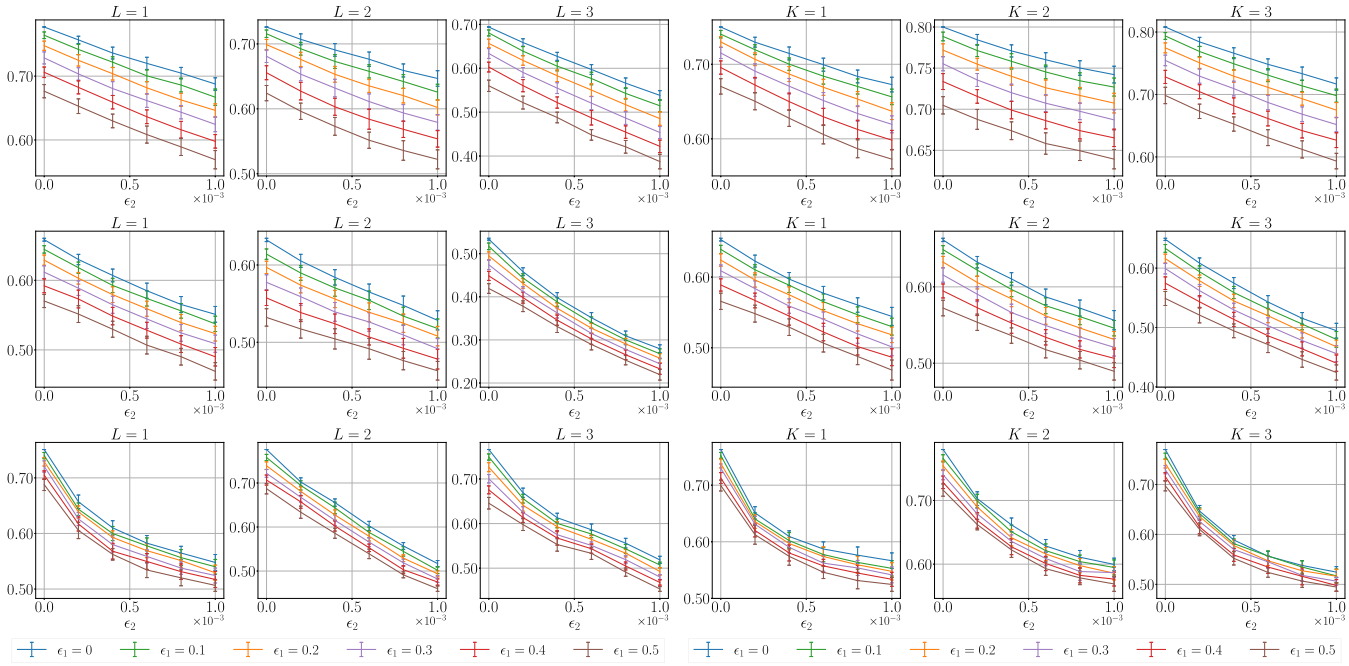
Fig. 7. Correlation between GIN (left panel) and SGCN (right panel) output differences and GSO distances. Analysis is conducted with varying edge deletion probabilities ϵ_1, and a fixed edge addition probability $\epsilon_2 = 1 \times 10^{-4}$.

Notably, the output differences observed in the two cases operate on different scales. For the SGCN with normalized GSO (right panel), the variation in output differences with increasing perturbation probability is more gradual compared to the unnormalized GSO used in GIN (left panel), which shows a steeper change. This discrepancy is likely due to the influence of the estimated GSO spectral norm λ.

2) Accuracy Drop Under Perturbation: After affirming the linear sensitivity in Theorem 3, we also examine the stability of GCNN under significant graph perturbations by observing the accuracy changes of same GCNN candidates as in Section VI-C1.

These experiments are conducted on three citation datasets: Cora, CiteSeer and PubMed [39]. The objective is to assess the impact of different perturbation budgets on the accuracy of GIN and SGCN models. The perturbation budget parameters are set as follows: edge deletion probability ϵ_1 varies within [0,0.5] in increments of 0.1, and edge addition probability ϵ_2 varies within $[0, 1 \times 10^{-3}]$ in increments of 2×10^{-4}. Consistent with the experimental settings in Section VI-C1, the same GCNN candidates are utilized. The averaged accuracy results are shown in Fig. 8, where the bar indicates the standard variance of accuracy results. The first, second and third rows correspond to datasets Cora, CiteSeer and PubMed, respectively.

A consistent pattern of accuracy decrease across all datasets and GCNN models is observed in Fig. 8, where the accuracy gradually decreases with increasing perturbation budgets. Notably, larger graphs (e.g., PubMed) exhibit a faster accuracy drop compared to smaller graphs (e.g., Cora and CiteSeer). This can be attributed to the alteration of more edges under the same perturbation budget in larger graphs. When fixing edge deletion probability ϵ_1, accuracy drops by approximately 10% (as in Fig. 8(a), 1st row with $L = 1$), and up to 20% (as in Fig. 8(a), 3rd row with $L = 3$). With a fixed edge addition probability



(a) GIN (b) SGCN

Fig. 8. Accuracy changes for GIN ($L = 1, 2, 3$) and SGCN ($K = 1, 2, 3$) under perturbations, with $\epsilon_1 \in [0, 0.5]$ and $\epsilon_2 \in [0, 1 \times 10^{-3}]$. The 1st, 2nd and 3rd rows correspond to Cora, CiteSeer, and PubMed datasets, respectively.

ϵ_2, the accuracy drop is around 10% (as in Fig. 8(a), 1st row with $L = 1$), and approximately 5% (as in Fig. 8(a), 3rd row with $L = 1$). This is likely because that, for sparse graphs, the same edge addition probability results in the addition of more edges than the number influenced by the same edge deletion probability.

The maximum of edge perturbation budget ϵ_1 and ϵ_2 is set to 0.5 and 1×10^{-3}, respectively. Consequently, up to 50% of the edges are deleted, and 70% are added relative to the original edge count. In this case, the graph structure is significantly perturbed. This significant graph perturbation makes the accuracy drop by up to 20%. Under such large perturbations, GCNN gives finite responses. Thus, the GCNN is stable in our context even when the downstream task performance is significantly impacted, which is due to large-scale edge perturbations. This also verifies Theorem 3, where it is stated that as long as the GSO perturbation is bounded/finite, the GCNN output difference is also bounded/finite.

VII. Conclusion and Discussion

This paper has presented an analytical framework for investigating the sensitivity of GCNNs to GSO perturbations, employing a probabilistic graph perturbation model. We have established tighter error bounds than those previously available. We have theoretically demonstrated that the expected output variation for a single layer of GCNN is linearly bounded by the GSO error, ensuring the stability (bounded output difference) of single-layer GCNN under bounded GSO errors. For multilayer GCNN, our analysis has shown that the dependency of GCNN output difference on GSO error can be described through a recursion of linearity. Specifically, this dependency is explicitly controlled by: the input feature, the GSO, error model parameters, Lipschitz constants of activation functions in GCNN, and GCNN weights. Through numerical experiments, we have validated our theoretical findings and confirmed that GCNNs (exemplified with GIN and SGCN) maintain stability under large-scale graph edge perturbations, despite significant performance reductions.

In this work, our primary focus is on edge perturbations in graphs, while potential modifications to the graph signal and node injections are not considered. Any alterations to the graph signal could be subsumed within the spectral norm when performing sensitivity analysis. However, node injection presents a challenge that cannot be addressed using the current definition of graph distance. This is due to the discrepancy in sizes between the unperturbed and perturbed graphs as the number of nodes increases. A potential solution to this issue could involve redefining the GSO distance using a different metric. In this context, Optimal Transport (OT) and its variants emerge as viable candidates for this task [40], [41], [42]. These methods allow for the augmentation of a smaller graph, facilitating the establishment of a meaningful graph distance metric [43]. Consequently, future research could explore an encompassing approach that considers all of the aforementioned types of graph perturbations. Such an investigation has the potential to yield more comprehensive insights into the stability of GCNNs under perturbations.

Graph regularization methods are commonly used to achieve robust graph learning and estimation [44]. Research on adversarial training of GCNNs typically uses specifically designed loss

functions to strengthen GCNNs against structural and feature perturbations, thus improving their performance stability against certain graph disturbances [45], [46], [47], [48], [49]. In graph learning, several techniques have been developed to regulate graphs and signals based on specific graph signal assumptions to perform graph estimation [15], [16], [50], [51]. With the inclusion of effective graph regularization, our sensitivity analysis offers insight that can contribute to the development of a uniform metric, paving the way for a more transferable and robust GCNN.

Appendix A
Upper Bound of $\|\mathbf{E}_u\|_1$

Proof of Lemma 1: We start with the first term in (23), which is bounded by $\tau_u \leq d_v$

$$\sum_{v\in\mathcal{D}_u}\frac{1}{\sqrt{d_u d_v}} \leq \sum_{v\in\mathcal{D}_u}\frac{1}{\sqrt{d_u \tau_u}} = \frac{\delta_u^-}{\sqrt{d_u\tau_u}}. \tag{38}$$

The second and third terms in (23) can be bounded using triangle inequality as follows

$$\begin{aligned}
&\sum_{v\in\mathcal{A}_u}\frac{1}{\sqrt{\hat{d}_u\hat{d}_v}} + \sum_{v\in\mathcal{R}_u}\left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{\hat{d}_u\hat{d}_v}}\right| \\
&\leq \sum_{v\in\mathcal{A}_u}\frac{1}{\sqrt{\hat{d}_u\hat{d}_v}} + \sum_{v\in\mathcal{R}_u}\left(\frac{1}{\sqrt{d_u d_v}} + \frac{1}{\sqrt{\hat{d}_u\hat{d}_v}}\right) \\
&= \sum_{v\in\mathcal{R}_u}\frac{1}{\sqrt{d_u d_v}} + \sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u}\frac{1}{\sqrt{\hat{d}_u\hat{d}_v}}.
\end{aligned} \tag{39}$$

For the first term in (39), we have

$$\sum_{v\in\mathcal{R}_u}\frac{1}{\sqrt{d_u d_v}} \leq \sum_{v\in\mathcal{R}_u}\frac{1}{\sqrt{d_u\tau_u}} \leq \frac{d_u-\delta_u^-}{\sqrt{d_u\tau_u}}. \tag{40}$$

For the second term in (39), we have

$$\begin{aligned}
&\sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u}\frac{1}{\sqrt{\hat{d}_u\hat{d}_v}} \\
&= \sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u}\frac{1}{\sqrt{(d_u+\delta_u^+-\delta_u^-)(d_v+\delta_v^+-\delta_v^-)}}
\end{aligned} \tag{41}$$

Thus, we have a new bound, which is more suited to our error model, that is

$$\begin{aligned}
\|\mathbf{E}_u\|_1 &\leq \frac{\delta_u^-}{\sqrt{d_u\tau_u}} + \frac{d_u-\delta_u^-}{\sqrt{d_u\tau_u}} \\
&\quad + \sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u}\frac{1}{\sqrt{(d_u+\delta_u^+-\delta_u^-)(d_v+\delta_v^+-\delta_v^-)}} \\
&= \sqrt{d_u/\tau_u} \\
&\quad + \sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u}\frac{1}{\sqrt{(d_u+\delta_u^+-\delta_u^-)(d_v+\delta_v^+-\delta_v^-)}}.
\end{aligned} \tag{42}$$

We will adapt the general bound (42) to the probabilistic error model presented in (8). In (42), we let

$$\begin{aligned}
Z_{u,1} &= \sqrt{d_u/\tau_u}, \\
Z_{u,2} &= \sum_{v\in\mathcal{A}_u\cup\mathcal{R}_u}\frac{1}{\sqrt{(d_u+\delta_u^+-\delta_u^-)(d_v+\delta_v^+-\delta_v^-)}},
\end{aligned} \tag{43}$$

where $\delta_u^- \sim \text{Bin}(d_u,\epsilon_1)$, $\delta_u^+ \sim \text{Bin}(d_u^*,\epsilon_2)$, $\delta_v^- \sim \text{Bin}(d_v,\epsilon_1)$, $\delta_v^+ \sim \text{Bin}(d_v^*,\epsilon_2)$, $d_u^* = N-d_u-1$ and $d_v^* = N-d_v-1$. Finally, we obtain

$$\|\mathbf{E}_u\|_1 \leq Z_{u,1} + Z_{u,2}. \tag{44}$$

This completes the proof.

Appendix B
Graph Filter Sensitivity

Proof of Theorem 2: First, we recall the following result.

Lemma 2: (Lemma 3, [52]): Suppose that $\hat{\mathbf{S}},\mathbf{S},\mathbf{E}\in\mathbb{R}^{N\times N}$ are Hermitian matrices satisfying $\hat{\mathbf{S}}=\mathbf{S}+\mathbf{E}$, and $\lambda = \max\{\|\hat{\mathbf{S}}\|,\|\mathbf{S}\|\}$. Then for every $k\geq 0$

$$\|\hat{\mathbf{S}}^k-\mathbf{S}^k\| = \|(\mathbf{S}+\mathbf{E})^k-\mathbf{S}^k\| \leq k\lambda^{k-1}\|\mathbf{E}\|. \tag{45}$$

Expand the filter representation in $\|\mathbf{h}(\hat{\mathbf{S}})-\mathbf{h}(\mathbf{S})\|$, as

$$\left\|\mathbf{h}(\hat{\mathbf{S}})-\mathbf{h}(\mathbf{S})\right\| = \left\|\sum_{k=0}^{K}\left(h_k\hat{\mathbf{S}}^k - h_k\mathbf{S}^k\right)\right\|. \tag{46}$$

By Lemma 2 and repeatably using triangle inequality, (46) is bounded by

$$\begin{aligned}
\left\|\sum_{k=0}^{K}\left(h_k\hat{\mathbf{S}}^k - h_k\mathbf{S}^k\right)\right\| &\leq \sum_{k=0}^{K}|h_k|\|\hat{\mathbf{S}}^k-\mathbf{S}^k\| \\
&\leq \sum_{k=0}^{K}|h_k|k\lambda^{k-1}\|\mathbf{E}\| = \sum_{k=1}^{K}|h_k|k\lambda^{k-1}\|\mathbf{E}\|.
\end{aligned} \tag{47}$$

The correlation between λ and $\|\mathbf{E}\|$ has two cases:

1) If $\lambda = \|\mathbf{S}\|$,

$$\mathbb{E}[\lambda^{k-1}\|\mathbf{E}\|] = \mathbb{E}[\lambda^{k-1}]\mathbb{E}[\|\mathbf{E}\|]; \tag{48}$$

2) If $\lambda = \|\hat{\mathbf{S}}\|$,

$$\mathbb{E}[\lambda^{k-1}\|\mathbf{E}\|] = \mathbb{E}[\lambda^{k-1}]\mathbb{E}[\|\mathbf{E}\|] + \text{Cov}[\|\mathbf{E}\|,\lambda^{k-1}]. \tag{49}$$

The following proof is based on the second case (49) because the covariance term can be set to zero to include the first case. By using (46) and taking the expectation of (47), we obtain

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{h}(\hat{\mathbf{S}})-\mathbf{h}(\mathbf{S})\right\|\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K}|h_k|k\lambda^{k-1}\|\mathbf{E}\|\right] \\
&\leq \sum_{k=1}^{K}k|h_k|\mathbb{E}\left[\lambda^{k-1}\|\mathbf{E}\|\right] \\
&= \sum_{k=1}^{K}k|h_k|\left(\mathbb{E}[\lambda^{k-1}]\mathbb{E}[\|\mathbf{E}\|] + \text{Cov}[\|\mathbf{E}\|,\lambda^{k-1}]\right).
\end{aligned} \tag{50}$$

In (50), let

$$\lambda_k = \mathbb{E}[\lambda^{k-1}], \tag{51}$$

$$\zeta_k = \text{Cov}[\|\mathbf{E}\|, \lambda^{k-1}]. \tag{52}$$

Then, we have

$$\mathbb{E}\left[\left\|\mathbf{h}(\hat{\mathbf{S}}) - \mathbf{h}(\mathbf{S})\right\|\right] \leq \sum_{k=1}^{K} k|h_k| \left(\lambda_k \mathbb{E}[\|\mathbf{E}\|] + \zeta_k\right). \tag{53}$$

This completes the proof.

Appendix C
GCNN Sensitivity

Proof of Theorem 3: **First Layer.** At the first layer $\ell = 1$, the graph convolution is performed as follows

$$\mathbf{Y}_1 = \sum_{k=1}^{K} \mathbf{S}^k \mathbf{X}_0 \mathbf{H}_{1k}, \quad \mathbf{X}_1 = \sigma_1(\mathbf{Y}_1). \tag{54}$$

For a perturbed GSO $\hat{\mathbf{S}}$, the difference between the perturbed and clean graph convolutions is

$$\hat{\mathbf{Y}}_1 - \mathbf{Y}_1 = \sum_{k=1}^{K} (\hat{\mathbf{S}}^k - \mathbf{S}^k) \mathbf{X}_0 \mathbf{H}_{1k}. \tag{55}$$

Using Lemma 2, we can bound (55) as follows

$$\left\|\hat{\mathbf{Y}}_1 - \mathbf{Y}_1\right\| \leq \sum_{k=1}^{K} k\lambda^{k-1} \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\| \|\mathbf{E}\|. \tag{56}$$

Similar to giving the upper bound for the expectation of graph filter distance from (47) to (53), given the constants $\lambda_k = \mathbb{E}[\lambda^{k-1}]$ and $\zeta_k = \text{Cov}[\|\mathbf{E}\|, \lambda^{k-1}]$, we take the expectation of (56) and obtain

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\mathbf{Y}}_1 - \mathbf{Y}_1\right\|\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K} k\lambda^{k-1} \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\| \|\mathbf{E}\|\right] \\
&= \sum_{k=1}^{K} k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\| \mathbb{E}\left[\lambda^{k-1} \|\mathbf{E}\|\right] \\
&= \sum_{k=1}^{K} k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\| \left(\mathbb{E}[\lambda^{k-1}] \mathbb{E}\left[\|\mathbf{E}\|\right] + \text{Cov}[\|\mathbf{E}\|, \lambda^{k-1}]\right) \\
&\leq \sum_{k=1}^{K} k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\| \left(\lambda_k \mathbb{E}\left[\|\mathbf{E}\|\right] + \zeta_k\right).
\end{aligned} \tag{57}$$

For simplicity, let $B_1 = \sum_{k=1}^{K} k\lambda_k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\|$, and $D_1 = \sum_{k=1}^{K} k\zeta_k \|\mathbf{X}_0\| \|\mathbf{H}_{1k}\|$. Thus, (57) illustrates that the expectation of the graph filter distance at the first layer is bounded by a polynomial of $\mathbb{E}[\|\mathbf{E}\|]$ as

$$\mathbb{E}\left[\left\|\hat{\mathbf{Y}}_1 - \mathbf{Y}_1\right\|\right] \leq B_1 \mathbb{E}[\|\mathbf{E}\|] + D_1. \tag{58}$$

Consider the nonlinearity function $\sigma_1(\cdot)$ at the first layer, which satisfies the Lipschitz condition

$$\|\sigma_1(\hat{\mathbf{Y}}) - \sigma_1(\mathbf{Y})\| \leq C_{\sigma_1} \|\hat{\mathbf{Y}} - \mathbf{Y}\|. \tag{59}$$

Applying this Lipschitz condition to (56), we have

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\mathbf{X}}_1 - \mathbf{X}_1\|\right] &= \mathbb{E}\left[\left\|\sigma_1(\hat{\mathbf{Y}}) - \sigma_1(\mathbf{Y})\right\|\right] \\
&\leq C_{\sigma_1} \mathbb{E}\left[\left\|\hat{\mathbf{Y}} - \mathbf{Y}\right\|\right] \leq C_{\sigma_1} B_1 \mathbb{E}\left[\|\mathbf{E}\|\right] + C_{\sigma_1} D_1.
\end{aligned} \tag{60}$$

Second Layer. At the second layer $\ell = 2$, the graph convolution is performed as

$$\mathbf{Y}_2 = \sum_{k=1}^{K} \mathbf{S}^k \mathbf{X}_1 \mathbf{H}_{2k}, \quad \mathbf{X}_2 = \sigma(\mathbf{Y}_2). \tag{61}$$

The difference between the perturbed and clean graph convolutions is given by

$$\begin{aligned}
\hat{\mathbf{Y}}_2 - \mathbf{Y}_2 &= \sum_{k=1}^{K} \hat{\mathbf{S}}^k \hat{\mathbf{X}}_1 \mathbf{H}_{2k} - \sum_{k=1}^{K} \mathbf{S}^k \mathbf{X}_1 \mathbf{H}_{2k} \\
&= \sum_{k=1}^{K} (\hat{\mathbf{S}}^k \hat{\mathbf{X}}_1 - \hat{\mathbf{S}}^k \mathbf{X}_1 + \hat{\mathbf{S}}^k \mathbf{X}_1 - \mathbf{S}^k \mathbf{X}_1) \mathbf{H}_{2k} \\
&= \sum_{k=1}^{K} \left(\hat{\mathbf{S}}^k (\hat{\mathbf{X}}_1 - \mathbf{X}_1) + (\hat{\mathbf{S}}^k - \mathbf{S}^k) \mathbf{X}_1\right) \mathbf{H}_{2k}.
\end{aligned} \tag{62}$$

Taking the expectation of (62) and using (49), Lemma 2 as well as the submultiplicativity of the spectral norm, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{\mathbf{Y}}_2 - \mathbf{Y}_2\right\|\right] \\
&\leq \mathbb{E}\left[\left\|\sum_{k=1}^{K} \left(\hat{\mathbf{S}}^k (\hat{\mathbf{X}}_1 - \mathbf{X}_1) + (\hat{\mathbf{S}}^k - \mathbf{S}^k) \mathbf{X}_1\right) \mathbf{H}_{2k}\right\|\right] \\
&\leq \sum_{k=1}^{K} \|\mathbf{H}_{2k}\| \mathbb{E}\left[\left\|\hat{\mathbf{S}}^k (\hat{\mathbf{X}}_1 - \mathbf{X}_1)\right\| + \left\|(\hat{\mathbf{S}}^k - \mathbf{S}^k) \mathbf{X}_1\right\|\right] \\
&\leq \sum_{k=1}^{K} \|\mathbf{H}_{2k}\| \left(\mathbb{E}[\lambda^k] \mathbb{E}\left[\|\hat{\mathbf{X}}_1 - \mathbf{X}_1\|\right] + \text{Cov}\left[\|\hat{\mathbf{X}}_1 - \mathbf{X}_1\|, \lambda^k\right]\right. \\
&\quad \left. + k \|\mathbf{X}_1\| \left(\mathbb{E}[\lambda^{k-1}] \mathbb{E}\left[\|\mathbf{E}\|\right] + \text{Cov}[\|\mathbf{E}\|, \lambda^{k-1}]\right)\right).
\end{aligned} \tag{63}$$

Let

$$\mu_{k,\ell-1} = \text{Cov}[\|\hat{\mathbf{X}}_{\ell-1} - \mathbf{X}_{\ell-1}\|, \lambda^k], \tag{64}$$

where $k = 1, \ldots, K$, and $\ell = 2, \ldots, L$. Thus, in (63), we have $\mu_{k,1} = \text{Cov}\|\hat{\mathbf{X}}_1 - \mathbf{X}_1\|$. Then, we can express (63) as a function controlled by $\mathbb{E}[\|\mathbf{E}\|]$

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\mathbf{Y}}_2 - \mathbf{Y}_2\right\|\right] &\leq \sum_{k=1}^{K} \|\mathbf{H}_{2k}\| \left(\lambda_{k+1} \mathbb{E}\left[\|\hat{\mathbf{X}}_1 - \mathbf{X}_1\|\right]\right. \\
&\quad \left. + \mu_{k,1} + k\lambda_k \|\mathbf{X}_1\| \mathbb{E}[\|\mathbf{E}\|] + k\zeta_k \|\mathbf{X}_1\|\right) \\
&\leq \sum_{k=1}^{K} \|\mathbf{H}_{2k}\| \Big((\lambda_{k+1} C_{\sigma_1} B_1 + k\lambda_k \|\mathbf{X}_1\|) \mathbb{E} \\
&\quad \times [\|\mathbf{E}\|] + \mu_{k,1} + \lambda_k C_{\sigma_1} D_1 + k\zeta_k \|\mathbf{X}_1\|\Big) \\
&\leq B_2 \mathbb{E}[\|\mathbf{E}\|] + D_2,
\end{aligned} \tag{65}$$

where $B_2 = \sum_{k=1}^{K}(\lambda_{k+1}C_{\sigma_1}B_1 + k\lambda_k\|\mathbf{X}_1\|)\|\mathbf{H}_{2k}\|$ and $D_2 = \sum_{k=1}^{K}(\mu_{k,1} + \lambda_k C_{\sigma_1}D_1 + k\zeta_k\|\mathbf{X}_1\|)\|\mathbf{H}_{2k}\|$. Consider the second layer's nonlinearity function $\sigma_2(\cdot)$, we have

$$\mathbb{E}\left[\|\hat{\mathbf{X}}_2 - \mathbf{X}_2\|\right] \leq C_{\sigma_2}B_2\mathbb{E}[\|\mathbf{E}\|] + C_{\sigma_2}D_2. \tag{66}$$

Generalization to Layer $\ell \geq 1$. By induction, we can generalize the result to the output difference at any layer $\ell \geq 1$

$$\mathbb{E}\left[\left\|\hat{\mathbf{X}}_\ell - \mathbf{X}_\ell\right\|\right] \leq C_{\sigma_\ell}B_\ell\mathbb{E}\left[\|\mathbf{E}\|\right] + C_{\sigma_\ell}D_\ell, \tag{67}$$

where

$$B_\ell = \sum_{k=1}^{K}\left(\lambda_{k+1}C_{\sigma_{\ell-1}}B_{\ell-1} + k\lambda_k\|\mathbf{X}_{\ell-1}\|\right)\|\mathbf{H}_{\ell k}\|,$$

$$D_\ell = \sum_{k=1}^{K}\left(\mu_{k,\ell-1} + \lambda_k C_{\sigma_{\ell-1}}D_{\ell-1} + k\zeta_k\|\mathbf{X}_{\ell-1}\|\right)\|\mathbf{H}_{\ell k}\|. \tag{68}$$

This completes the proof. □

Appendix D
Single-Layer GIN Sensitivity

Proof: In a single-layer GIN, we assume that the inner MLP has two layers as earlier introduced in the paper. The outputs of a single-layer GIN ($L = 1$) with original and perturbed GSOs are given as

$$\mathbf{X}_L = \mathbf{h}_{\mathbf{\Theta}_L}\left(\mathbf{S}\mathbf{X}_{L-1}\right), \tag{69}$$

$$\hat{\mathbf{X}}_L = \mathbf{h}_{\mathbf{\Theta}_L}\left(\hat{\mathbf{S}}\hat{\mathbf{X}}_{L-1}\right). \tag{70}$$

Expanding (69) and (70) with full matrix transformations, we have

$$\mathbf{X}_L = \sigma_{L2}(\sigma_{L1}(\mathbf{S}\mathbf{X}_{L-1}\mathbf{W}_{L1} + \mathbf{B}_{L1})\mathbf{W}_{L2} + \mathbf{B}_{L2}), \tag{71}$$

$$\hat{\mathbf{X}}_L = \sigma_{L2}(\sigma_{L1}(\hat{\mathbf{S}}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} + \mathbf{B}_{L1})\mathbf{W}_{L2} + \mathbf{B}_{L2}). \tag{72}$$

We can split (71) as

$$\mathbf{Y}_{L1} = \mathbf{S}\mathbf{X}_{L-1}\mathbf{W}_{L1} + \mathbf{B}_{L1}, \tag{73a}$$

$$\mathbf{X}_{L1} = \sigma_{L1}(\mathbf{Y}_{L1}), \tag{73b}$$

$$\mathbf{Y}_{L2} = \mathbf{X}_{L1}\mathbf{W}_{L2} + \mathbf{B}_{L2}, \tag{73c}$$

$$\mathbf{X}_L = \sigma_{L2}(\mathbf{Y}_{L2}), \tag{73d}$$

where $\mathbf{X}_{L1}$ denotes the intermediate output of the first layer, and $\mathbf{X}_L$ represents the output of the second layer. For simplicity of notation, we use $\mathbf{X}_L$ instead of $\mathbf{X}_{L2}$. Similarly, we split (72) as

$$\hat{\mathbf{Y}}_{L1} = \hat{\mathbf{S}}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} + \mathbf{B}_{L1}, \tag{74a}$$

$$\hat{\mathbf{X}}_{L1} = \sigma_{L1}(\hat{\mathbf{Y}}_{L1}), \tag{74b}$$

$$\hat{\mathbf{Y}}_{L2} = \hat{\mathbf{X}}_{L1}\mathbf{W}_{L2} + \mathbf{B}_{L2}, \tag{74c}$$

$$\hat{\mathbf{X}}_L = \sigma_{L2}(\hat{\mathbf{Y}}_{L2}). \tag{74d}$$

Then, the ℓ_2 norm of difference between the perturbed (74d) and clean outputs (73d) is

$$\|\hat{\mathbf{X}}_L - \mathbf{X}_L\| = \|\sigma_{L2}(\hat{\mathbf{Y}}_{L2}) - \sigma_{L2}(\mathbf{Y}_{L2})\|. \tag{75}$$

Using the Lipschitz condition of the nonlinearity function $\sigma_{L2}(\cdot)$ in (75), we have

$$\|\hat{\mathbf{X}}_L - \mathbf{X}_L\| \leq C_{\sigma_{L2}}\|\hat{\mathbf{Y}}_{L2} - \mathbf{Y}_{L2}\|. \tag{76}$$

Representing $\hat{\mathbf{Y}}_{L2}$ by (74c) and $\mathbf{Y}_{L2}$ by (73c), we have

$$\begin{aligned}\|\hat{\mathbf{Y}}_{L2} - \mathbf{Y}_{L2}\| &= \|\hat{\mathbf{X}}_{L1}\mathbf{W}_{L2} - \mathbf{X}_{L1}\mathbf{W}_{L2}\| \\ &\leq \|\hat{\mathbf{X}}_{L1} - \mathbf{X}_{L1}\|\|\mathbf{W}_{L2}\|. \end{aligned}\tag{77}$$

Representing $\hat{\mathbf{X}}_{L1}$ by (74b) and $\mathbf{X}_{L1}$ by (73b), we obtain

$$\|\hat{\mathbf{X}}_{L1} - \mathbf{X}_{L1}\| = \|\sigma_{L1}(\hat{\mathbf{Y}}_{L1}) - \sigma_{L1}(\mathbf{Y}_{L1})\|. \tag{78}$$

Using the Lipschitz condition of the nonlinearity function $\sigma_{L1}(\cdot)$ in (78), we have

$$\|\hat{\mathbf{X}}_{L1} - \mathbf{X}_{L1}\| \leq C_{\sigma_{L1}}\|\hat{\mathbf{Y}}_{L1} - \mathbf{Y}_{L1}\|. \tag{79}$$

Representing $\hat{\mathbf{Y}}_{L1}$ by (74a) and $\mathbf{Y}_{L1}$ by (73a), we have

$$\|\hat{\mathbf{Y}}_{L1} - \mathbf{Y}_{L1}\| = \|\hat{\mathbf{S}}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} - \mathbf{S}\mathbf{X}_{L-1}\mathbf{W}_{L1}\|. \tag{80}$$

We can rewrite (80) by deleting and adding $\mathbf{S}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1}$ as

$$\begin{aligned}&\hat{\mathbf{S}}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} - \mathbf{S}\mathbf{X}_{L-1}\mathbf{W}_{L1} \\ &= \hat{\mathbf{S}}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} - \mathbf{S}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} + \mathbf{S}\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} \\ &\quad - \mathbf{S}\mathbf{X}_{L-1}\mathbf{W}_{L1} \\ &= (\hat{\mathbf{S}} - \mathbf{S})\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1} + \mathbf{S}(\hat{\mathbf{X}}_{L-1} - \mathbf{X}_{L-1})\mathbf{W}_{L1}. \end{aligned}\tag{81}$$

Substituting (81) into (80), and using the triangular inequality, we have

$$\begin{aligned}&\|\mathbf{Y}_{L1} - \hat{\mathbf{Y}}_{L1}\| \\ &\leq \|(\hat{\mathbf{S}} - \mathbf{S})\hat{\mathbf{X}}_{L-1}\mathbf{W}_{L1}\| + \|\mathbf{S}(\hat{\mathbf{X}}_{L-1} - \mathbf{X}_{L-1})\mathbf{W}_{L1}\| \\ &\leq \|\hat{\mathbf{S}} - \mathbf{S}\|\|\hat{\mathbf{X}}_{L-1}\|\|\mathbf{W}_{L1}\| + \|\mathbf{S}\|\|\hat{\mathbf{X}}_{L-1} - \mathbf{X}_{L-1}\|\|\mathbf{W}_{L1}\|. \end{aligned}\tag{82}$$

For the second term in (82), we have $\hat{\mathbf{X}}_{L-1} = \mathbf{X}_{L-1} = \mathbf{X}_0$ for $L = 1$. Then, with the definition of GSO error (5), (82) becomes

$$\|\hat{\mathbf{Y}}_{L1} - \mathbf{Y}_{L1}\| \leq \|\mathbf{E}\|\|\mathbf{X}_{L-1}\|\|\mathbf{W}_{L1}\|. \tag{83}$$

By connecting (83), (79), (77), (76) together, we can bound the one-layer GIN output difference as

$$\|\hat{\mathbf{X}}_L - \mathbf{X}_L\| \leq C_{\sigma_{L2}}C_{\sigma_{L1}}\|\mathbf{W}_{L2}\|\|\mathbf{W}_{L1}\|\|\mathbf{X}_{L-1}\|\|\mathbf{E}\|. \tag{84}$$

Taking the expectation of (84), we have

$$\mathbb{E}\left[\|\hat{\mathbf{X}}_L - \mathbf{X}_L\|\right] \leq C_{\sigma_{L2}}C_{\sigma_{L1}}\|\mathbf{W}_{L2}\|\|\mathbf{W}_{L1}\|\|\mathbf{X}_{L-1}\|\mathbb{E}\left[\|\mathbf{E}\|\right]. \tag{85}$$

Finally, let $\xi = C_{\sigma_{L2}}C_{\sigma_{L1}}\|\mathbf{W}_{L2}\|\|\mathbf{W}_{L1}\|\|\mathbf{X}_{L-1}\|$, then, we have

$$\mathbb{E}\left[\|\hat{\mathbf{X}}_L - \mathbf{X}_L\|\right] \leq \xi\mathbb{E}\left[\|\mathbf{E}\|\right]. \tag{86}$$

This completes the proof. □

REFERENCES

[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[2] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 117–127, Nov. 2020.

[3] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[4] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," *IEEE Trans. Signal Process.*, vol. 72, pp. 4745–4781, 2024.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, 2017, pp. 1–14.

[6] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. 7th Int. Conf. Learn. Representations*, New Orleans, LA, USA, 2019, pp. 1–17.

[7] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan, "Label efficient semi-supervised learning via graph filtering," in *Proc. 32nd Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 9574–9583.

[8] F. Wu, T. Zhang, A. H. d. Souza Jr., C Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 6861–6871.

[9] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, Jan. 2019.

[10] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, 2018, pp. 1–12.

[11] E. Isufi, F. Gama, and A. Ribeiro, "EdgeNets: Edge varying graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7457–7473, Nov. 2022.

[12] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2320–2333, May 2019.

[13] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 3844–3858.

[15] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[16] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.

[17] A. Buciulea, S. Rey, and A. G. Marques, "Learning graphs from smooth and graph-stationary signals with hidden variables," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 273–287, 2022.

[18] J. Miettinen, S. A. Vorobyov, and E. Ollila, "Modelling and studying the effect of graph errors in graph signal processing," *Signal Process.*, vol. 189, Dec. 2021, Art. no. 108256.

[19] Z. Gao, E. Isufi, and A. Ribeiro, "Stability of graph convolutional neural networks to stochastic perturbations," *Signal Process.*, vol. 188, Nov. 2021, Art. no. 108216.

[20] K. Xu et al., "Topology attack and defense for graph neural networks: An optimization perspective," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, 2019, pp. 3961–3967.

[21] E. Ceci and S. Barbarossa, "Graph signal processing in the presence of topology uncertainties," *IEEE Trans. Signal Process.*, vol. 68, pp. 1558–1573, 2020.

[22] H. Kenlay, D. Thanou, and X. Dong, "On the stability of graph convolutional neural networks under edge rewiring," in *Proc. 46th IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toronto, Canada, 2021, pp. 8513–8517.

[23] H. Kenlay, D. Thanou, and X. Dong, "Interpretable stability bounds for spectral graph filters," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, vol. 139, pp. 5388–5397.

[24] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.

[25] R. Levie, W. Huang, L. Bucci, M. Bronstein, and G. Kutyniok, "Transferability of spectral graph convolutional neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 12462–12520, Nov. 2021.

[26] H. Dai et al., "Adversarial attack on graph structured data," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, vol. 80, pp. 1115–1124.

[27] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, London, U.K., 2018, pp. 2847–2856.

[28] 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 *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, 2019, pp. 4816–4823.

[29] B. Wang, J. Jia, X. Cao, and N. Z. Gong, "Certified robustness of graph neural networks against adversarial structural perturbation," in *Proc. 27th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1645–1653.

[30] L. Lin, E. Blaser, and H. Wang, "Graph structural attack by perturbing spectral distance," in *Proc. 28th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Washington DC, USA, 2022, pp. 989–998.

[31] X. Wang, E. Ollila, and S. A. Vorobyov, "Graph neural network sensitivity under probabilistic error model," in *Proc. 30th Eur. Signal Process. Conf.*, Belgrade, Serbia, 2022, pp. 2146–2150.

[32] 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[33] M. Penrose, "Random geometric graphs," in *Oxford Studies in Probability*, vol. 5, London, U.K.: Oxford Univ. Press, 2003.

[34] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos Nat. Lab., Los Alamos, NM, USA, Tech. Rep. LA-UR-08-05495; LA-UR-08-5495, 2008.

[35] G. Golub and C. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012.

[36] T. Aven, "Upper (lower) bounds on the mean of the maximum (minimum) of a number of random variables," *J. Appl. Probability*, vol. 22, no. 3, pp. 723–728, Sep. 1985.

[37] G. Ohayon, T. Michaeli, and M. Elad, "The perception-robustness tradeoff in deterministic image restoration," in *Proc. 41st Int. Conf. Mach. Learning*, Vienna, Austria, Jul. 21–27, 2024, vol. 235, pp. 38599–38638.

[38] B. Weisfeiler and A. Lehman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsia*, vol. 2, no. 9, pp. 12–16, 1968.

[39] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, Sep. 2008, Art. no. 93.

[40] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Unbalanced optimal transport: Dynamic and kantorovich formulations," *J. Funct. Anal.*, vol. 274, no. 11, pp. 3090–3123, Jun. 2018.

[41] L. Chapel, M. Z. Alaya, and G. Gasso, "Partial optimal tranport with applications on positive-unlabeled learning," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 2903–2913.

[42] H. P. Maretic, M. El Gheche, M. Minder, G. Chierchia, and P. Frossard, "Wasserstein-based graph alignment," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 353–363, 2022.

[43] C.-Y. Chuang and S. Jegelka, "Tree mover's distance: Bridging graph metrics and stability of graph neural networks," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, New Orleans, USA, 2022, vol. 35, pp. 2944–2957.

[44] L. Sun et al., "Adversarial attack and defense on graph data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7693–7711, Aug. 2023.

[45] H. Jin and X. Zhang, "Latent adversarial training of graph convolution networks," in *Proc. 36th Int. Conf. Mach. Learn. Workshop Learn. Reasoning Graph-Structured Representations*, Long Beach, CA, USA, 2019, pp. 1–7.

[46] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2493–2504, Jun. 2021.

[47] Q. Dai, X. Shen, L. Zhang, Q. Li, and D. Wang, "Adversarial training methods for network embedding," in *Proc. 30th World Wide Web Conf.*, San Francisco, CA, USA, 2019, pp. 329–339.

[48] J. Ren et al., "Integrated defense for resilient graph matching," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, vol. 139, pp. 8982–8997.

[49] X. Zhao et al., "Expressive 1-Lipschitz neural networks for robust multiple graph learning against adversarial attacks," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, vol. 139, pp. 12719–12735.

[50] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from filtered signals: Graph system and diffusion kernel identification," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 360–374, Jun. 2019.

[51] X. Pu, S. L. Chau, X. Dong, and D. Sejdinovic, "Kernel-based graph learning from smooth signals: A functional viewpoint," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 7, pp. 192–207, 2021.
[52] R. Levie, E. Isufi, and G. Kutyniok, "On the transferability of spectral graph filters," in *Proc. 13th Int. Conf. Sampling Theory Appl.*, Bordeaux, France, 2019, pp. 1–5.

Xinjue Wang (Graduate Student Member, IEEE) received the B.Eng. degree in electronic and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2020, and the M.Sc. degree in computer, communication and information sciences in 2022 from Aalto University, Espoo, Finland, where he is currently working toward the Ph.D. degree. His research interests include graph signal processing and machine learning.

Esa Ollila (Senior Member, IEEE) received the M.Sc. degree in mathematics from the University of Oulu, Oulu, Finland, in 1998, the Ph.D. degree (with Hons.) in statistics from the University of Jyväskylä, Jyväskylä, Finland, in 2002, and the D.Sc. (Tech.) degree (with Hons.) in signal processing from Aalto University, Espoo, Finland, in 2010. Since June 2015, he has been an Associate Professor of signal processing with Aalto University. From 2004 to 2007, he was a Postdoctoral Fellow and an Academy Research Fellow with the Academy of Finland, from August 2010 to May 2015. From 2010 to 2011, he was a Visiting Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, NJ, USA. He has coauthored a textbook *Robust Statistics for Signal Processing* published by Cambridge University Press, in 2018. His research interests include statistical signal processing, high-dimensional statistics, and machine learning. He is also on the Board of Directors of European Association for Signal Processing (EURASIP) and an Elected Member of IEEE SPS Signal Processing Theory and Methods (SPTM) Technical Committee (2022-). He was the General Co-Chair for EUSIPCO-2023 and has been an Associate Editor of *Scandinavian Journal of Statistics* since 2021.

Sergiy A. Vorobyov (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in systems and control from the National University of Radio Electronics, Kharkiv, Ukraine, in 1994 and 1997, respectively. He is currently a Professor with the Department of Information and Communications Engineering, Aalto University, Espoo, Finland. He has held various Faculty and Research positions with the University of Alberta, Edmonton, AB, Canada, the Joint Research Institute between Heriot-Watt University, Edinburgh, U.K., and Edinburgh University, Edinburgh, U.K., Duisburg-Essen University, Duisburg, Germany, and Darmstadt University of Technology, Darmstadt, Germany, McMaster University, Hamilton, ON, Canada, the Institute of Physical and Chemical Research, Japan, and the National University of Radio Electronics, Kharkiv, Ukraine. His research interests include optimization and multi-liner algebra methods in signal processing and data analysis, statistical, array, and graph signal processing, estimation, detection and learning theory and methods, computational imaging, and multi-antenna, very large, cooperative, and cognitive systems. Dr. Vorobyov was the recipient of the 2004 IEEE Signal Processing Society Best Paper Award, 2007 Alberta Ingenuity New Faculty Award, 2011 Carl Zeiss Award (Germany), 2012 NSERC Discovery Accelerator Award, and IEEE ICASSP 2023 Top 3% paper recognition, and other awards. From 2016 to 2020, he was the Senior Area Editor of the IEEE SIGNAL PROCESSING LETTERS an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2006 to 2010, and IEEE SIGNAL PROCESSING LETTERS from 2007 to 2009. He was also a Member with the Sensor Array and Multi-Channel Signal Processing and Signal Processing for Communications and Networking Technical Committees of the IEEE Signal Processing Society from 2007 to 2012 and 2010 to 2016, respectively. He was the Track Chair for Asilomar 2011, Pacific Grove, CA, USA, Technical Co-Chair for IEEE CAMSAP 2011, Puerto Rico, Tutorial Chair for ISWCS 2013, Ilmenau, Germany, Technical Co-Chair for IEEE SAM 2018, Sheffield, U.K., Technical Co-Chair for IEEE CAMSAP 2023, Costa Rica, and the General Co-Chair for EUSIPCO 2023, Helsinki, Finland.