A APPENDIX

A.1 METRIC FOR ASSESSING GNN STABILITY

In the context of single-label classification in graph nodes, consider an output vector $\mathbf{y} = [y_1, y_2, ..., y_k]$ corresponding to an input $\mathbf{x}$, where k represents the total number of classes. The model's predicted class is denoted as $\hat{y} = \text{argmax}_i(y_i)$. Now, let's assume that the output vector transforms to $\mathbf{y}' = [y'_1, y'_2, ..., y'_k]$, while preserving the ordinality of the elements, i.e., if $y_i > y_j$, then $y'_i > y'_j$. This condition ensures that $\hat{y'} = \text{argmax}_i(y'_i) = \hat{y}$.

Relying solely on model accuracy can be deceptive, as it is contingent upon the preservation of the ordinality of the output vector elements, even when the vector itself undergoes significant transformations. This implies that the model's accuracy remains ostensibly unaffected as long as the ranking of the elements within the output vector is conserved. However, this perspective neglects potential alterations in the model's prediction confidence levels.

In contrast, the cosine similarity provides a more holistic measure as it quantifies the angle between two vectors, thereby indicating the extent of modification in the output direction. This method offers a more granular insight into the impact of adversarial attacks on the model's predictions.

Moreover, it is crucial to consider the nature of the output space, Y. In situations where Y forms a probability distribution, a common occurrence in classification problems, the application of a distribution distance measure such as the Kullback-Leibler (KL) divergence is typically more suitable. Unlike the oversimplified perspective of accuracy, these measures can provide a nuanced understanding of the degree of perturbation introduced in the predicted probability distribution by an adversarial attack. This additional granularity can expose subtle modifications in the model's output that might be missed when solely relying on accuracy as a performance metric.

A.2 DIMENSIONALITY REDUCTION AND WEIGHTED SPECTRAL EMBEDDING MATRIX

In the exploration of dimensionality reduction techniques, principal component analysis (PCA) stands as a well-established method. The essential concept in PCA is to retain the most salient information from the input data while reducing the dimensionality. This is accomplished by transforming the original dataset into a new set of uncorrelated variables, known as principal components, which are ordered by the amount of variance each holds from the original data. The first few components usually contain the bulk of the variance, and thus, keep the key information.

Similar to PCA, our approach aims to reduce the dimension of graphs while preserving the key structural information. Here, the structural information of a graph is measured by resistance distance, an effective metric for assessing the relationship between nodes.

In the realm of graph-based manifold problems, both geodesic and effective-resistance distances serve as valid metrics for measuring the distance between nodes on a graph. Notably, effective-resistance distances have been the subject of extensive study in contemporary spectral graph theory, revealing their relevance to a multitude of significant problems. These include cover and commute times of random walks (Chandra et al., 1996), the enumeration of spanning trees in a graph, and more.

We present Lemma 2 to elucidate the relationship between effective-resistance distance $d^{eff}(p, q)$ and geodesic distance $d^{geo}(p, q)$:

Lemma 2. *The effective-resistance distance $d^{eff}(p, q)$ and the geodesic distance $d^{geo}(p, q)$ between any two nodes p and q within an N-node connected, undirected graph have the following relationship (Cheng et al., 2021):*

1. $d^{eff}(p, q) = d^{geo}(p, q)$ *if only one path exists between nodes p and q;*
2. $d^{eff}(p, q) < d^{geo}(p, q)$ *otherwise.*

This Lemma 2 asserts that the equality $d^{eff}(p, q) = d^{geo}(p, q)$ consistently holds for tree structures, given that a single path exists between any pair of nodes in a tree. For general graphs, the resistance distance $d^{eff}(p, q)$ is upper-bounded by the geodesic distance $d^{geo}(p, q)$. The effective-resistance distance $d^{eff}(p, q)$ between any two nodes p and q for an undirected and connected graph G is given by (Feng, 2021):

$$d^{\text{eff}}(p,q) = e_{p,q}^{\top} L_G^{+} e_{p,q} = |U_N^{\top} e_{p,q}|_2^2 = \sum_{i=2}^{|V|} \frac{(u_i^T e_{p,q})^2}{\lambda_i} \tag{9}$$

In this equation, $e_{p,q} = e_p - e_q$, $e_p \in \mathbb{R}^N$ is the standard basis vector with the p-th element being 1 and the rest being 0, $L_G^{+} \in \mathbb{R}^{N\times N}$ is the Moore–Penrose pseudoinverse of the graph Laplacian matrix $L_G \in \mathbb{R}^{N\times N}$, and U_N is the eigensubspace matrix comprising $N-1$ nontrivial weighted Laplacian eigenvectors $U_N = \left[\frac{u_2}{\sqrt{\lambda_2}}, \ldots, \frac{u_N}{\sqrt{\lambda_N}}\right] \in \mathbb{R}^{N\times(N-1)}$, $0 = \lambda_1 < \lambda_2, \ldots, \leq \lambda_N$ are the ascending eigenvalues corresponding to their respective eigenvectors $u_1, \ldots, u_N$.

A.3 DIMENSION REDUCTION VIA SPECTRAL EMBEDDING

Spectral Embedding is a potent technique for dimensionality reduction in graph-based data. It is designed to preserve the data's inherent structure while representing it in a lower-dimensional space. This is achieved by computing the eigenvectors and eigenvalues of the Laplacian matrix, which is derived from the input data graph. The primary objective of Spectral Embedding is to minimize the following function (Belkin & Niyogi, 2003):

$$\min_{Y\in\mathbb{R}^{n\times k}} \operatorname{Tr}(Y^{\top}LY), \quad \text{subject to } Y^{\top}DY = I \tag{10}$$

In this equation, Tr denotes the trace of a matrix, and I is the identity matrix. The constraint $Y^{\top}DY = I$ ensures that the columns of Y are orthogonal with respect to D, which promotes a unique and informative low-dimensional embedding.

According to (Belkin & Niyogi, 2003), the optimal solution to the optimization problem in Equation 10 is achieved by selecting the k smallest eigenvectors associated with the k smallest eigenvalues of L. This underlines the pivotal role of eigenvalues and their corresponding eigenvectors in deriving a lower-dimensional representation of the graph. By leveraging these properties, Spectral Embedding effectively condenses the complexity of the graph while preserving its essential structure, thereby facilitating more efficient data analysis and processing.

A.4 ALGORITHM FLOW OF SAGMAN

The Algorithm 1 shows the key steps in SAGMAN. The process of obtaining low-dimensional graphs G_X and G_Y involves several steps, starting with the reduction of the graph dimension of G_X using the weighted spectral embedding matrix. This matrix is denoted as $U_N = \left[\frac{u_2}{\sqrt{\lambda_2}}, \ldots, \frac{u_N}{\sqrt{\lambda_N}}\right] \in \mathbb{R}^{N\times(N-1)}$, where $0 = \lambda_1 < \lambda_2, \ldots, \leq \lambda_N$ are the eigenvalues in ascending order. Following the approach suggested by Deng et al. (2022), we enhance the graph construction by concatenating node features with dominant eigenvectors. The next step involves constructing the input PGM. This is achieved by first constructing a kNN dense graph using the embedding matrix E, and then sparsifying this dense graph through short-cycle decomposition. The structure of the low-dimensional graph G_X is obtained with the assistance of the input PGM. This graph, along with the node feature X, is then fed into a GNN to generate the output matrix Y. To construct the output PGM, we follow a similar process as with the input PGM: we construct a kNN dense graph with Y and sparsify the dense graph via short-cycle decomposition. This results in the low-dimensional graph G_Y, which is obtained with the aid of the output PGM. From the graphs G_X and G_Y, we derive the Laplacian matrices L_X and L_Y, respectively. The metric $\delta^M(p, q_i)$ can then be calculated by evaluating $V_N^{\top} e_{p,q_i}$. In Appendix A.8, we provide a more detailed explanation of the relationship between $V_N^{\top} e_{p,q_i}$ and $\delta^M(p, q_i)$. Finally, the DMD score is obtained by averaging the DMD scores of node p and its neighbors.

A.5 DMD CALCULATION WITHOUT SAGMAN

In this study, we present the outcomes of stability quantification using original input and output graphs, as depicted in Figure 4. We also demonstrate the Nettack result using the original input graph in Table 3. Our experimental findings underscore a key observation: DMD calculations, when applied to the original inputs of GNNs, do not adequately quantify the stability of samples.

Algorithm 1 The algorithm flow of SAGMAN

```
Input: Input graph $G$, Node features $X$, GNN
Output: Average DMD for all nodes.
$U_N \leftarrow$ compute_weighted_spectral_embedding$(G)$
$E \leftarrow$ concatenate$(X, U_N)$
$kNN\_dense\_graph \leftarrow$ construct_kNN_graph$(E)$
$input\_PGM \leftarrow$ sparse_graph$(kNN\_dense\_graph)$
$G_X \leftarrow$ get_low_dimensional_graph$(input\_PGM)$
$Y \leftarrow$ GNN$(G_X, X)$
$kNN\_dense\_graph\_Y \leftarrow$ construct_kNN_graph$(Y)$
$output\_PGM \leftarrow$ sparse_graph$(kNN\_dense\_graph\_Y)$
$G_Y \leftarrow$ get_low_dimensional_graph$(output\_PGM)$
$L_X \leftarrow$ compute_laplacian$(G_X)$
$L_Y \leftarrow$ compute_laplacian$(G_Y)$
$V_N \leftarrow$ calculate_generalized_eigenvectors$(L_Y, L_X)$
for each node $p$ in $G_X$ do
    Compute the average DMD for node $p$ and its neighbors in $M_X$:
        $\frac{1}{|\mathbb{N}_X(p)|} \sum_{q_i \in \mathbb{N}_X(p)} \left( \|V_N^\top e_{p,q}\|_2^2 \right)$
end for
```

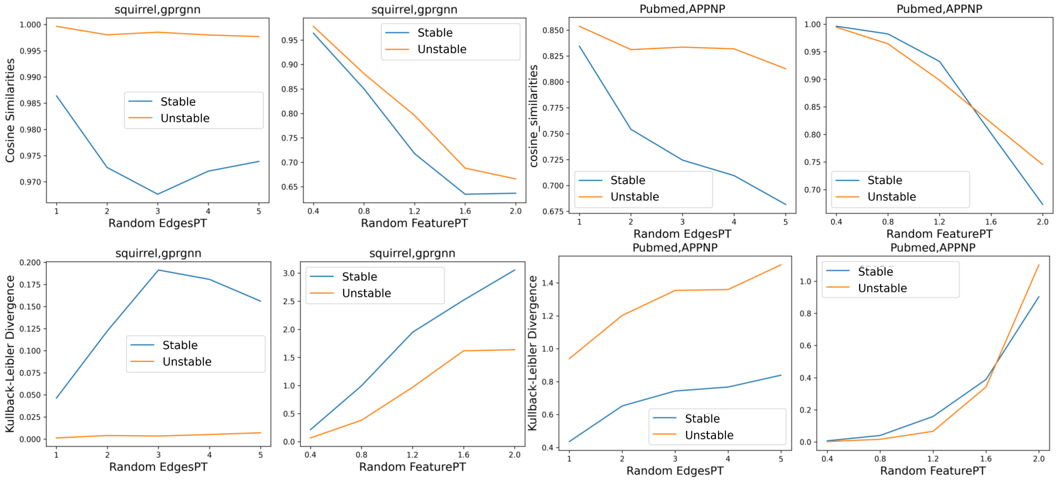

Figure 4: The upper quartet of figures represents cosine similarities, while the lower quartet illustrates KL divergence. "Random EdgesPT" corresponds to the DICE edge evasion attack. "Random FeaturePT" refers to Gaussian noise evasion perturbation $X + \xi\eta$, where X is feature matrix, η is Gaussian noise, ξ is noise level controls

A.6 EFFECTIVE-RESISTANCE ESTIMATION

The effective-resistance between nodes $(p, q) \in |V|$ can be computed using the following equation:

$$R_{eff}(p, q) = \sum_{i=2}^{N} \frac{(u_i^\top e_{p,q})^2}{u_i^\top L_G u_i}, \tag{11}$$

where u_i represents the eigenvector corresponding to σ_i eigenvalue of L_G and $e_{p,q} = e_p - e_q$. To avoid the computational complexity associated with computing eigenvalues/eigenvectors, we leverage a scalable algorithm that approximates the eigenvectors by exploiting the Krylov subspace. In this context, given a nonsingular matrix $A_{N \times N}$ and a vector $c \neq 0 \in \mathbb{R}^N$, the order-$(m)$ Krylov subspace

Table 3: Nettack adversarial attack targeting at selected Cora samples in GCN, we bold better results

Nettack Level	Cosine Similarities: Stable/Unstable	
	DMD with Original Input and Output	SAGMAN
1	0.90/0.96	**0.99/0.90**
2	0.84/0.93	**0.98/0.84**
3	0.81/0.91	**0.97/0.81**

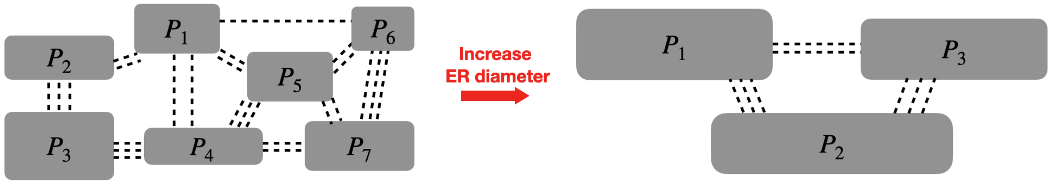

Figure 5: Graph decomposition results with respect to effective-resistance (ER) diameter

generated by A from c is defined as:

$$\kappa_m(A,c) := span(c, Ac, A^2c, ..., A^{m-1}c), \tag{12}$$

where c denotes a random vector, and A denotes the adjacency matrix of graph G. We compute a new set of vectors denoted as $x^{(1)}$, $x^{(2)}$, ..., $x^{(m)}$ by ensuring that the Krylov subspace vectors are mutually orthogonal with unit length. We estimate the effective-resistance between node p and q using Equation 11 by exploring the eigenspace of L_G and selecting the vectors that capture various spectral properties of G:

$$R_{eff}(p,q) \approx \sum_{i=1}^{m} \frac{(x^{(i)^\top} e_{p,q})^2}{x^{(i)\top} L_G x^{(i)}}, \tag{13}$$

We control the diameter of each cycle by propagating effective resistances across multiple levels. Let $G = (V, E)$ represent the graph at the δ-th level, and let the edge $(p, q) \in E$ be a contracted edge that creates a supernode $\vartheta \in V^{(\delta+1)}$ at level $\delta + 1$. We denote the vector of node weights as $\eta^{(\delta)} \in \mathbb{R}^{V^{(\delta)}}_{\geq 0}$, which is initially set to all zeros for the original graph. The update of η at level $\delta + 1$ is defined as follows:

$$\eta_\vartheta := \eta(p^{(\delta)}) + \eta(q^{(\delta)}) + R^{(\delta)}_{eff}(p,q). \tag{14}$$

Consequently, the effective-resistance diameter of each cycle is influenced not only by the computed effective-resistance ($R^{(\delta)}_e$) at the current level but also by the clustering information acquired from previous levels.

The graph decomposition results with respect to effective-resistance (ER) diameter are illustrated in Figure 5. The figure demonstrates that selecting a larger ER diameter leads to the decomposition of the graph into a smaller number of partitions, with more nodes included in each cluster. On the left side of the figure, the graph is decomposed into seven partitions: $P_1, ..., P_7$, by choosing a smaller ER diameter. Conversely, increasing the ER diameter on the right side of the figure results in the graph being partitioned into three clusters: P_1, P_2, and P_3.

A.7 ADDITIONAL RESULTS

We present the additional results in Figure 6, Figure 7, Figure 8, and Table 5. Table 4 summarizes the datasets utilized.

The horizontal axes denote the level of perturbation. 'Random FeaturePT' corresponds to the DICE adversarial attack. In this scenario, the horizontal axes represent deleting X edges within the same class and adding X edges between different classes. On the other hand, 'Random FeaturePT' represents Gaussian noise perturbation, and the horizontal axes indicate the Gaussian noise level ξ. The subtitle of each graph presents the names of corresponding datasets and GNNs. Post-perturbation, 'stable' samples should exhibit greater cosine similarities than 'unstable' ones. Additionally, the KLD of 'stable' samples should be lower than that of 'unstable' samples.

Table 4: Nettack adversarial attack targeting at selected Cora samples in GCN

Nettack Level	Cosine Similarities: Stable/Unstable SAGMAN
1	0.99/0.90
2	0.98/0.84
3	0.97/0.81

Table 5: Summary of datasets used in our experiments

Dataset	Type	Nodes	Edges	Classes	Features
Cora	Homophily	2,485	5,069	7	1,433
Cora-ML	Homophily	2,810	7,981	7	2,879
Pubmed	Homophily	19,717	44,324	3	500
Citeseer	Homophily	2,110	3,668	6	3703
Chameleon	Heterophily	2,277	62,792	5	2,325
Squirrel	Heterophily	5,201	396,846	5	2,089
ogbn-arxiv	Homophily	169,343	1,166,243	40	128

A.8 WHY GENERALIZED EIGENPAIRS ASSOCIATE WITH DMD

Cheng et al. (2021) propose a method to estimate the maximum distance mapping distortion (DMD), denoted as δ^{F}_{max}, by solving the following combinatorial optimization problem:

$$\max \delta^{F} = \max_{\substack{\forall p,q\in V\\ p\neq q}} \frac{e_{p,q}^{\top} L_{Y}^{+} e_{p,q}}{e_{p,q}^{\top} L_{X}^{+} e_{p,q}} \tag{15}$$

However, due to the discrete nature of $e_{p,q}$, this problem has a super-linear complexity. To avoid this, the Spectral Perturbation Analysis of Distortions score has been introduced, which can be computed in nearly-linear time using fast Laplacian solvers. Given the Laplacian matrices L_X and L_Y of the input and output graphs G_X and G_Y respectively, the stability score of a function (model) $Y = F(X)$ is defined as $\lambda_{max}(L_Y^{+} L_X)$, where λ_{max} is the largest generalized eigenvalue.

Lemma 3. *When computing δ^{F}_{max} via effective-resistance distance, the stability score is an upper bound of δ^{F}_{max}.*

A function $Y = M(X)$ is called Lipschitz continuous if there exists a real constant $K \geq 0$ such that for all $x_i, x_j \in X$:

$$dist_Y(M(x_i), M(x_j)) \leq K dist_X(x_i, x_j), \tag{16}$$

where K is the Lipschitz constant for the function M. The smallest Lipschitz constant, denoted by K^{*}, is called the best Lipschitz constant.

Lemma 4. *Let the resistance distance be the distance metric, we have:*

$$\lambda_{max}(L_Y^{+} L_X) \geq K^{*} \geq \delta^{M}_{max} \tag{17}$$

Lemma 4 indicates that the $\lambda_{max}(L_Y^{+} L_X)$ is also an upper bound of the best Lipschitz constant K^{*} under the low dimensional manifold setting. A greater $\lambda_{max}(L_Y^{+} L_X)$ of a function (model) implies worse stability since the output will be more sensitive to small input perturbations.

Lemma 5. *A node pair (p, q) is deemed non-robust if it exhibits a large DMD, i.e., $\delta^{M}(p, q) \approx \delta^{M}_{\max}$.*

This lemma suggests that a non-robust node pair consists of nodes that are adjacent in the input graph G_X but distant in the output graph G_Y. To effectively identify such non-robust node pairs, the Cut Mapping Distortion (CMD) metric was introduced:

Lemma 6. *For two graphs G_X and G_Y sharing the same node set V, let $S \subset V$ denote a node subset and $\bar{S}$ denote its complement. Also, let $cut_G(S, \bar{S})$ denote the number of edges crossing S and $\bar{S}$ in graph G. The CMD $\zeta(S)$ of node subset S is defined as:*

$$\zeta(S) \overset{\text{def}}{=} \frac{cut_{G_Y}(S, \bar{S})}{cut_{G_X}(S, \bar{S})} \tag{18}$$

A small CMD score indicates that node pairs crossing the boundary of S are likely to have small distances in G_X but large distances in G_Y.

Lemma 7. *Given the Laplacian matrices L_X and L_Y of input and output graphs, respectively, the minimum CMD $\zeta_{\min}$ satisfies the following inequality:*

$$\zeta_{\min} = \min_{\forall S \subset V} \zeta(S) \geq \frac{1}{\sigma_{\max}(L_Y^+ L_X)} \tag{19}$$

Lemma 7 establishes a connection between the maximum generalized eigenvalue $\sigma_{\max}(L_Y^+ L_X)$ and $\zeta_{\min}$, indicating the ability to exploit the largest generalized eigenvalues and their corresponding eigenvectors to measure the stability of node pairs. Embedding G_X with generalized eigenpairs. We first compute the weighted eigensubspace matrix $V_r \in \mathbb{R}^{N \times r}$ for spectral embedding on G_X with N nodes:

$$V_r \overset{\text{def}}{=} \left[v_1\sqrt{\sigma_1}, ..., v_r\sqrt{\sigma_r}\right], \tag{20}$$

where $\sigma_1, \sigma_2, ..., \sigma_r$ represent the first r largest eigenvalues of $L_Y^+ L_X$ and $v_1, v_2, ..., v_r$ are the corresponding eigenvectors. Consequently, the input graph G_X can be embedded using V_r, so each node is associated with an r-dimensional embedding vector. We can then quantify the stability of an edge $(p, q) \in E_X$ by measuring the spectral embedding distance of its two end nodes p and q. Formally, we have the edge stability score defined for any edge $(p, q) \in E_X$ as $stability^M(p, q) \overset{\text{def}}{=} \|V_r^\top e_{p,q}\|_2^2$

Lemma 8. *Let $u_1, u_2, ..., u_r$ denote the first r dominant generalized eigenvectors of $L_X L_Y^+$. If an edge (p, q) is dominantly aligned with one dominant eigenvector u_k, where $1 \leq k \leq r$, the following holds:*

$$(u_i^\top e_{p,q})^2 \approx \begin{cases} \alpha_k^2 \gg 0 & \textit{if } (i = k) \\ 0 & \textit{if } (i \neq k). \end{cases} \tag{21}$$

Then its edge stability score has the following connection with its DMD computed using effective-resistance distances:

$$\|V_r^\top e_{p,q}\|_2^2 \propto \left(\delta^M(p, q)\right)^3. \tag{22}$$

The stability score of an edge $(p, q) \in E_X$ can be regarded as a surrogate for the directional derivative $\|\nabla_v F(x)\|$ under the manifold setting, where $v = \pm(x_p - x_q)$. An edge with a larger stability score is considered more non-robust and can be more vulnerable to attacks along the directions formed by its end nodes.

Last, The node DMD score can be calculated for any node (data sample) $p \in V$ as follows:

$$\frac{1}{|\mathbb{N}_X(p)|} \sum_{q_i \in \mathbb{N}_X(p)} \left(\|V_N^\top e_{p,q}\|_2^2)\right) \propto \frac{1}{|\mathbb{N}_X(p)|} \sum_{q_i \in \mathbb{N}_X(p)} \left(\delta^M(p, q_i)\right)^3 \tag{23}$$

where $q_i \in \mathbb{N}_X(p)$ denotes the i-th neighbor of node p in graph G_X, and $\mathbb{N}_X(p) \in V$ denotes the node set including all the neighbors of p. The DMD score of a node (data sample) p can be regarded as a surrogate for the function gradient $\|\nabla F(x)\|$ where x is near p under the manifold setting. A node with a larger stability score implies it is likely more vulnerable to adversarial attacks.

In this study, we restrict our calculations of DMD scores to the two largest generalized eigenpairs. We also show results of the third and fourth largest generalized eigenpairs, as well as the fifth and sixth largest generalized eigenpairs in Table 6.

Table 6: Cosine Similarity and KL Divergence for DICE Edges Perturbation and Random Feature Perturbation. We bold the better results.

		Largest Generalized Eigenpairs: 1st,2nd/3rd,4th/5th,6th			
Method	PT Level	Cos Stable	Cos Unstable	KL Stable	KL Unstable
EdgesPT	1	**0.99**/0.98/**0.99**	0.97/0.96/**0.91**	**0.04**/0.25/0.12	0.46/0.73/**1.83**
EdgesPT	2	**0.96**/0.87/**0.96**	0.92/0.89/**0.86**	**0.82**/1.17/1.11	2.43/2.01/**3.47**
EdgesPT	3	0.88/0.81/**0.90**	0.92/0.87/**0.84**	2.18/2.29/**2.04**	2.71/3.79/**5.34**
EdgesPT	4	0.83/0.80/**0.85**	0.88/**0.85**/**0.85**	4.57/**3.29**/3.83	4.16/4.23/**5.25**
EdgesPT	5	0.81/0.78/**0.82**	0.88/0.83/**0.76**	**3.88**/4.56/5.28	4.92/4.07/**6.02**
FeaturePT	0.4	**1.00**/0.84/0.81	1.00/0.75/**0.68**	**0.00**/2.16/5.70	0.00/4.57/**8.24**
FeaturePT	0.8	**1.00**/0.79/0.75	1.00/0.73/**0.71**	**0.00**/4.00/6.49	0.01/4.87/**8.29**
FeaturePT	1.2	**0.98**/0.78/0.76	0.98/0.70/**0.69**	**0.02**/6.11/6.65	0.08/5.47/**9.33**
FeaturePT	1.6	**0.97**/0.69/0.71	0.93/0.70/**0.64**	**0.01**/8.94/8.85	1.09/7.07/**8.41**
FeaturePT	2.0	**0.80**/0.67/0.70	0.82/0.72/**0.66**	**1.72**/8.57/9.53	4.02/5.45/**9.40**

A.9 BROADER IMPACT

Our research into the stability analysis of Graph Neural Networks (GNNs) holds profound implications for various applications utilizing GNNs, ranging from social network analytics to bioinformatics. By investigating the stability of these models, we aim to boost their reliability in real-world scenarios. Stability is paramount to ensure that minor perturbations in input do not lead to drastic changes in output. In scenarios where GNNs are used to make critical decisions, such as disease diagnosis or financial fraud detection, instability could lead to severe negative consequences. Therefore, enhancing the stability of GNNs could have a positive societal impact by increasing the reliability of such systems and reducing erroneous decisions. Our work in identifying and addressing instability issues in GNNs could also contribute to the development of more robust defense mechanisms against adversarial attacks. These attacks, which introduce small, intentionally designed perturbations to mislead models, pose a significant threat to machine learning systems, including GNNs. Hence, our stability analysis can help improve the security of GNN-based systems. On the other hand, we acknowledge that our research might also be used inappropriately. For instance, adversaries could potentially exploit our analysis to devise more sophisticated attacks that specifically target the identified instability issues. Therefore, while our work is geared towards improving the robustness of GNNs, it is crucial to consider and guard against potential misuse.

Overall, our research serves as an essential step towards more robust and reliable GNN models, though vigilance and continuous research are necessary to mitigate potential negative consequences.

A.10 CONSTRUCTED GRAPH AND THE ORIGINAL GRAPH

As shown in Appendix A.2, Equation 9 suggests that effective-resistance distances computed with the first k smallest eigenpairs can serve as a good approximation of the original distances, particularly when there is a large eigengap $\sigma_k \ll \sigma_{k+1}$.

To validate this, we compare the exact resistance distances computed using all eigenpairs with the approximate ones obtained using the first few eigenpairs. The comprehensive results of resistance correlation coefficients for 100 randomly selected node pairs in the Cora graph are reported in Table 7. As observed, even a small number of eigenpairs can provide a satisfactory approximation of the original effective-resistance distances. In this work, spectral embedding utilizes the smallest 50 eigenpairs.

We also show the comparison of the resistance correlation coefficient based on principal component analysis (PCA) in Table 8, which indicates PCA fails to estimate the resistance distances of the original graph.

Table 7: Resistance correlation coefficients for Cora graph with various numbers of eigenvectors

Number of Eigenvectors	Resistance Correlation Coefficient
20	0.69
30	0.78
40	0.79
50	0.82
100	0.87
200	0.93
300	0.96
400	0.97
500	0.99

Table 8: Resistance correlation coefficient based on PCA. We selected 100 randomly selected node pairs in the Cora graph. Larger resistance correlation coefficients indicate better estimation.

Number of Resistance Principal Components	Correlation Coefficient
20	-0.03
30	-0.01
40	0.09
50	0.15
100	0.14
200	0.15
300	0.14
400	0.13
500	0.12

A.11 EVALUATE THE ENTIRE DATASET

Previous works (Cheng et al., 2021; Hua et al., 2021; Chang et al., 2017) highlighted that only part of the dataset plays a crucial role in model stability, so we want to focus on the difference between the most "stable" and "unstable" parts. However, it is certainly feasible to evaluate the entire graph. Table 9 shows the result regarding the Pubmed dataset in GPRGNN under Gaussian noise perturbation. Samples were segmented based on SAGMAN ranking, with the bottom 20% being the most "stable", the middle 60% as intermediate, and the top 20% representing the most "unstable". As anticipated, the "stable" category (representing the bottom 20%) should exhibit the lowest average KL divergences. This is followed by the intermediate category (covering the mid 60%), and finally, the "unstable" category (comprising the top 20%) should display the highest divergences.

Table 9: KL Divergence for different FeaturePT values.

FeaturePT	KL divergence (bottom 20%)	KL divergence (mid 60%)	KL divergence (top 20%)
0.4	0.01	0.03	0.03
0.8	0.09	0.16	0.19
1.2	0.43	0.56	0.59

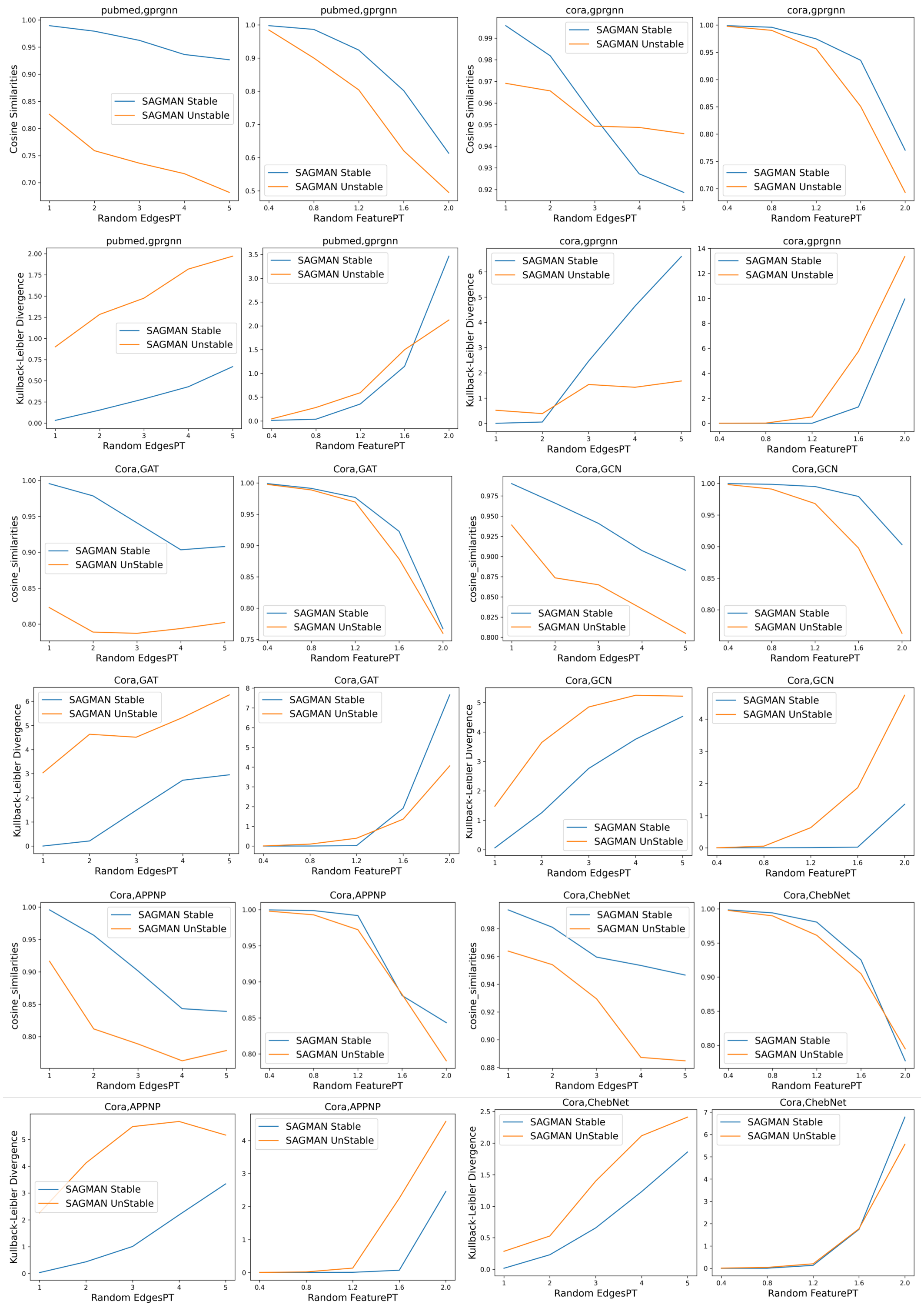

Figure 6: Figures represent cosine similarities and KL divergence. "Random EdgesPT" corresponds to the DICE edge evasion attack. "Random FeaturePT" refers to Gaussian noise evasion perturbation $X + \xi\eta$, where X is feature matrix, η is Gaussian noise, ξ is noise level controls

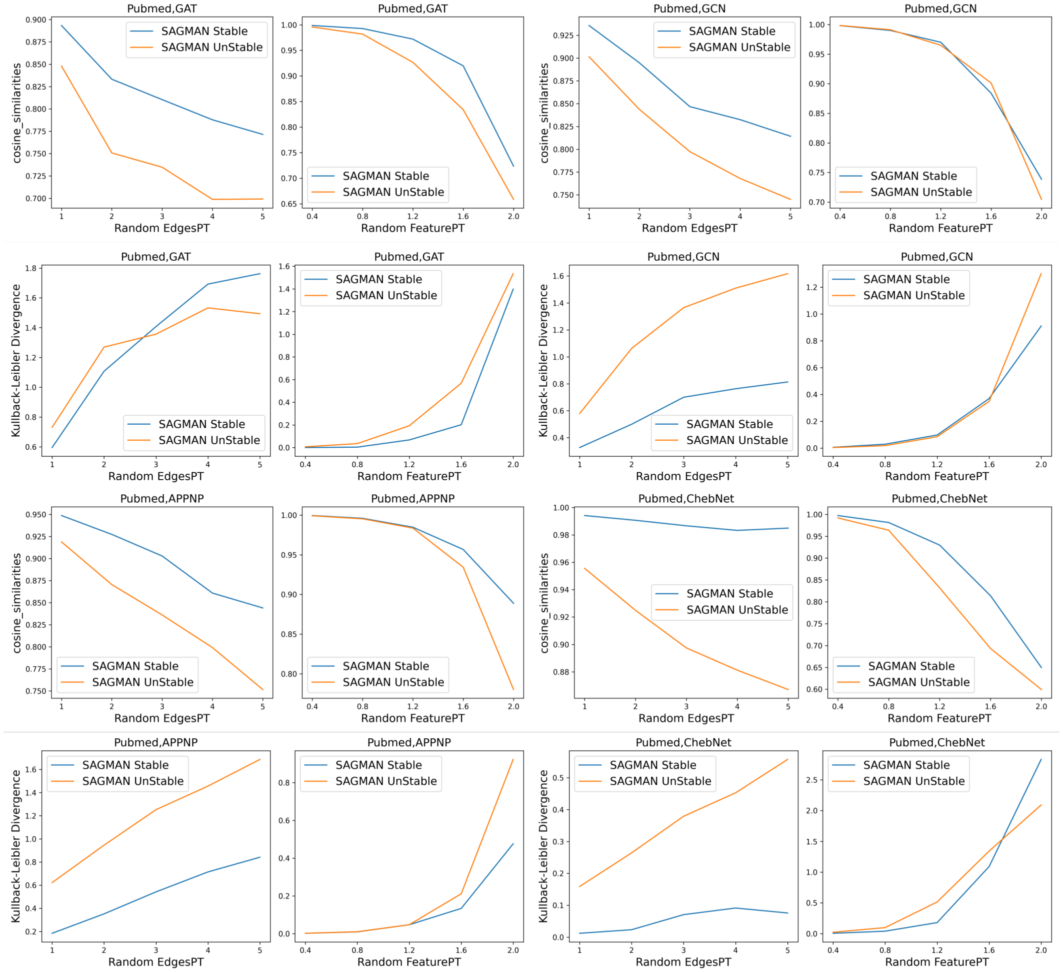

Figure 7: Figures represent cosine similarities and KL divergence. "Random EdgesPT" corresponds to the DICE edge evasion attack. "Random FeaturePT" refers to Gaussian noise evasion perturbation $X + \xi\eta$, where X is feature matrix, η is Gaussian noise, ξ is noise level controls

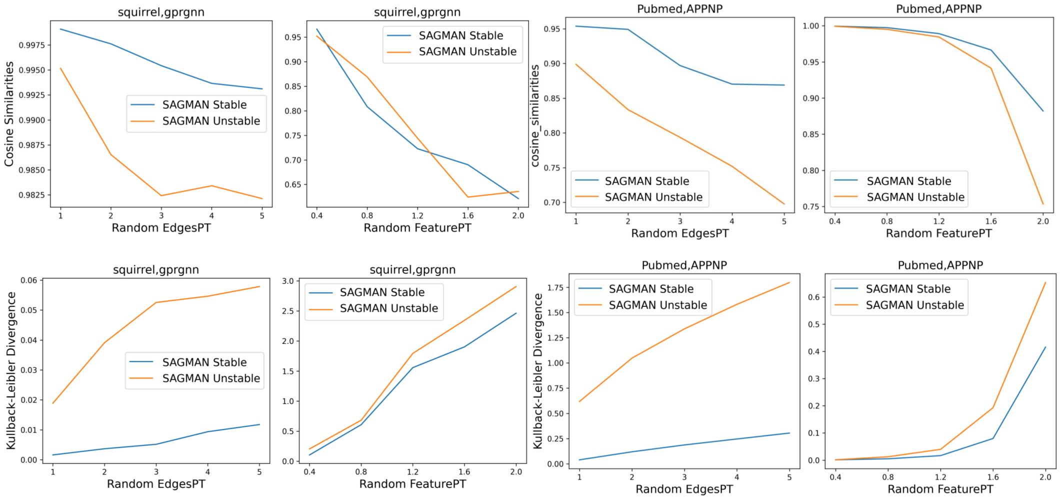

Figure 8: The top four plots show cosine similarities, and the bottom four depict KLD. "Random EdgesPT" stands for DICE edge evasion, where the x-axis marks the number of same-class edges deleted and inter-class edges added. "Random FeaturePT" signifies Gaussian noise evasion, with the x-axis denoting the added noise level $X\eta$, scaled by the data's standard deviation η.