Robust Graph Neural Networks via Unbiased Aggregation

Zhichao Hou[1*] **Ruiqi Feng**[1*] **Tyler Derr**[2] **Xiaorui Liu**[1†]

[1]North Carolina State University, [2]Vanderbilt University

{zhou4,xliu96}@ncsu.edu ruiqifeng.2024@gmail.com tyler.derr@vanderbilt.edu

Abstract

The adversarial robustness of Graph Neural Networks (GNNs) has been questioned due to the false sense of security uncovered by strong adaptive attacks despite the existence of numerous defenses. In this work, we delve into the robustness analysis of representative robust GNNs and provide a unified robust estimation point of view to understand their robustness and limitations. Our novel analysis of estimation bias motivates the design of a robust and unbiased graph signal estimator. We then develop an efficient Quasi-Newton Iterative Reweighted Least Squares algorithm to solve the estimation problem, which is unfolded as robust unbiased aggregation layers in GNNs with theoretical guarantees. Our comprehensive experiments confirm the strong robustness of our proposed model under various scenarios, and the ablation study provides a deep understanding of its advantages. Our code is available at https://github.com/chris-hzc/RUNG.

1 Introduction

Graph neural networks (GNNs) have gained tremendous popularity in recent years due to their ability to capture topological relationships in graph-structured data [1]. However, most GNNs are vulnerable to adversarial attacks, which can lead to a substantial decline in predictive performance [2, 3, 4]. Despite the numerous defense strategies proposed to robustify GNNs, a recent study has revealed that most of these defenses are not as robust as initially claimed [5]. Specifically, under adaptive attacks, they easily underperform the multi-layer perceptrons (MLPs) which do not utilize the graph topology information at all [5]. Therefore, it is imperative to thoroughly investigate the limitations of existing defenses and develop innovative robust GNNs to securely harness the topology information in graphs.

Existing defenses attempt to bolster the resilience of GNNs using diverse approaches. For instance, Jaccard-GCN [6] and SVD-GCN [7] aim to denoise the graph by removing potential adversarial edges during the pre-processing procedure, while ProGNN [3] learns the clean graph structure during the training process. GRAND [8] and robust training [9, 10] also improve the training procedure through data augmentation. GNNGuard [2] and RGCN [11] reinforce their GNN architectures by heuristically reweighting edges in the graph. Additionally, there emerge some ODEs-inspired architectures including the GraphCON [12] and HANG [13] that demonstrate decent robustness. Although most of these defenses exhibit decent robustness against transfer attacks, i.e., the attack is generated through surrogate models, they encounter catastrophic performance drops when confronted with adaptive adversarial attacks that directly attack the victim model [5].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

[*]Equal contribution.

[†]Corresponding author.

Concerned by the false sense of security, we provide a comprehensive study on existing defenses under adaptive attacks. Our preliminary study in Section 2 indicates that SoftMedian [4], TWIRLS [14], and ElasticGNN [15] exhibit closely aligned performance and notably outperform other defenses despite their apparent architectural differences. However, as attack budgets increase, these defenses still experience a severe performance decrease and underperform the graph-agnostic MLPs. These observations are intriguing, but the underlying reasons are still unclear.

To unravel the aligned robustness and performance degradation of SoftMedian, TWIRLS, and ElasticGNN, we delve into their theoretical understanding and unveil their inherent connections and limitations in the underlying principles. Specifically, their improved robustness can be understood from a unified view of ℓ_1-based robust graph smoothing. Moreover, we unearth the problematic estimation bias of ℓ_1-based graph smoothing that allows the adversarial impact to accumulate as the attack budget escalates, which provides a plausible explanation of their declining robustness. Motivated by these understandings, we propose a robust and unbiased graph signal estimator to reduce the estimation bias in GNNs. We design an efficient Quasi-Newton IRLS algorithm that unrolls as robust unbiased aggregation layers to safeguard GNNs against adversarial attacks. Our contributions can be summarized as follows:

- We provide a unified view of ℓ_1-based robust graph signal smoothing to justify the improved and closely aligned robustness of representative robust GNNs. Moreover, we reveal their estimation bias, which explains their severe performance degradation as the attack budgets increase.
- We propose a robust and unbiased graph signal estimator to mitigate the estimation bias in ℓ_1-based graph signal smoothing and design an efficient Quasi-Newton IRLS algorithm to solve the non-smooth and non-convex estimation problem with theoretical guarantees.
- The proposed algorithm can be readily unfolded as feature aggregation building blocks in GNNs, which not only provides clear interpretability but also covers many classic GNNs as special cases.
- Comprehensive experiments demonstrate that our proposed GNN significantly improves the robustness while maintaining clean accuracy. We also provide comprehensive ablation studies to validate its working mechanism.

2 Estimation Bias Analysis of Robust GNNs

In Section 2.1, we conduct a preliminary study to evaluate the robustness of several representative robust GNNs. In Section 2.2, we establish a unified view as ℓ_1-based models to uncover the inherent connections of three well-performing GNNs, including SoftMedian, TWIRLS and ElasticGNN. In Section 2.3, we leverage the bias of ℓ_1-based estimation to explain the catastrophic performance degradation in the preliminary experiments.

Notation. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph with node set $\mathcal{V} = \{v_1, \ldots, v_n\}$ and edge set $\mathcal{E} = \{e_1, \ldots, e_m\}$. The adjacency matrix of $\mathcal{G}$ is denoted as $\boldsymbol{A} \in \{0,1\}^{n\times n}$ and the graph Laplacian matrix is $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. $\boldsymbol{D} = \text{diag}(d_1, \ldots, d_n)$ is the degree matrix where $d_i = |\mathcal{N}(i)|$ and $\mathcal{N}(i)$ is the neighborhood set of v_i. The node feature matrix is denoted as $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n]^\top \in \mathbb{R}^{n\times d}$, and $\boldsymbol{f}^{(0)}$ ($\boldsymbol{F}^{(0)}$) denotes the node feature vector (matrix) before graph smoothing in decoupled GNN models. Let $\Delta \in \{-1, 0, 1\}^{m\times n}$ be the incidence matrix whose l-th row denotes the l-th edge $e_l = (i, j)$ such that $\Delta_{li} = -1, \Delta_{lj} = 1, \Delta_{lk} = 0\ \forall k \notin \{i, j\}$. $\tilde{\Delta}$ is its normalized version : $\tilde{\Delta}_{lj} = \Delta_{lj}/\sqrt{d_j}$. For a vector $\boldsymbol{x} \in \mathbb{R}^d$, we use ℓ_1-based gragh smoothing penalty to denote either $\|\boldsymbol{x}\|_1 = \sum_i |\boldsymbol{x}_i|$ or $\|\boldsymbol{x}\|_2 = \sqrt{\sum_i \boldsymbol{x}_i^2}$. Note that we use ℓ_2-based gragh smoothing penalty to denote $\|\boldsymbol{x}\|_2^2 = \sum_i \boldsymbol{x}_i^2$.

2.1 Robustness Analysis

To test the robustness of existing GNNs without the false sense of security, we perform a preliminary evaluation of existing robust GNNs against adaptive attacks. We choose various baselines including the undefended MLP, GCN [16], some of the most representative defenses in [5], and two additional robust models TWIRLS [14] and ElasticGNN [15]. We execute adaptive local evasion topological attacks and test the node classification accuracy on the Cora ML and Citeseer datasets. The detailed settings follow Section 4.1. From Figure 1, it can be observed that:

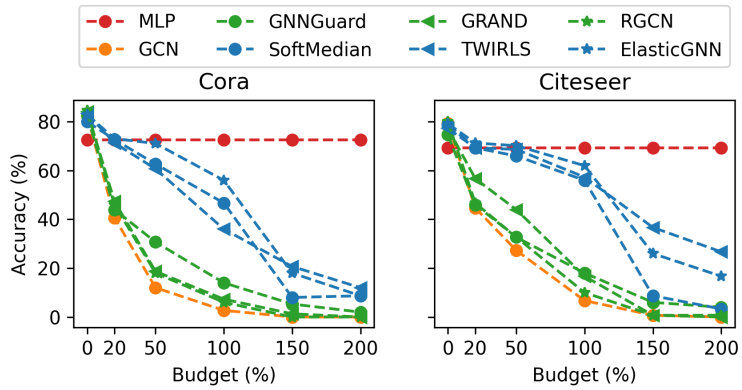

Figure 1: Robustness analysis under adaptive local attack. The perturbation budget (x-axis) is the number of edges allowed to be perturbed relative to the target node's degree. SoftMedian, TWIRLS, and ElasticGNN (blue curves) exhibit similarly aligned competitive robustness among all the selected robust GNNs, but all models experience catastrophic performance degradation as the attack budget increases.

- Among all the selected robust GNNs, only SoftMedian, TWIRLS, and ElasticGNN exhibit notable and closely aligned improvements in robustness whereas other GNNs do not show obvious improvement over undefended GCN.
- SoftMedian, TWIRLS, and ElasticGNN encounter a similar catastrophic performance degradation as the attack budget scales up. Their accuracy easily drops below that of the graph-unware MLP, indicating their failure in safely exploiting the topology of the data.

2.2 A Unified View of Robust Estimation

Our preliminary study provides intriguing observations in Section 2.1, but the underlying reasons behind these phenomena remain obscure. This motivates us to delve into their theoretical understanding and explanation. In this section, we will compare the architectures of three well-performing GNNs, aiming to reveal their intrinsic connections.

SoftMedian [4] substitutes the GCN aggregation for enhanced robustness with the dimension-wise median $\boldsymbol{m}_i \in \mathbb{R}^d$ for all neighbors of each node $i \in \mathcal{V}$. However, computing the median involves operations like ranking and selection, which is not compatible with the back-propagation training of GNNs. Therefore, the median is approximated as a differentiable weighted sum $\tilde{\boldsymbol{m}}_i = \frac{1}{Z}\sum_{j\in\mathcal{N}(i)} w(\boldsymbol{f}_j, \boldsymbol{m}_i)\boldsymbol{f}_j, \forall i \in \mathcal{V}$, where $\boldsymbol{m}_i$ is the exact non-differentiable dimension-wise median, $\boldsymbol{f}_j$ is the feature vector of the j-th neighbor, $w(\boldsymbol{x}, \boldsymbol{y}) = e^{-\beta\|\boldsymbol{x}-\boldsymbol{y}\|_2}$, and $Z = \sum_k w(\boldsymbol{f}_k, \boldsymbol{m}_k)$ is a normalization factor. In this way, the aggregation assigns the largest weights to the neighbors closest to the actual median.

TWIRLS [14] utilizes the iteratively reweighted least squares (IRLS) algorithm to optimize the objective with parameter λ, and $\rho(y) = y$ is the default:

$$2\lambda \sum_{(i,j)\in\mathcal{E}} \rho(\|\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j\|_2) + \sum_{i\in\mathcal{V}} \|\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}^{(0)}\|_2^2, \tilde{\boldsymbol{f}}_i = (1+\lambda d_i)^{-\frac{1}{2}}\boldsymbol{f}_i. \tag{1}$$

ElasticGNN [15] proposes the elastic message passing which unfolds the proximal alternating predictor-corrector (PAPC) algorithm to minimize the objective with parameter $\lambda_{\{1,2\}}$:

$$\frac{1}{2}\sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2 + \lambda_1 \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}} \right\|_p + \lambda_2 \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}} \right\|_2^2, \text{where } p \in \{1,2\}. \tag{2}$$

A Unified View of Robust Estimation. While these three approaches have seemingly different architectures, we provide a unified view of robust estimation to illuminate their inherent connections. First, the objective of TWIRLS in Eq. (1) can be considered as a particular case of ElasticGNN

with $\lambda_2 = 0$ and $p = 2$ when neglecting the difference in the node degree normalization. However, TWIRLS and ElasticGNN leverage different optimization solvers, i.e., IRLS and PAPC, which leads to vastly different GNN layers. Second, SoftMedian approximates the computation of medians in a soft way of weighted sums, which can be regarded as approximately solving the dimension-wise median estimation problem [17]: $\arg\min_{\boldsymbol{f}_i} \sum_{j\in\mathcal{N}(i)} \|\boldsymbol{f}_i - \boldsymbol{f}_j\|_1$. Therefore, SoftMedian can be regarded as the ElasticGNN with $\lambda_2 = 0$ and $p = 1$. We also note that the SoftMedoid [18] approach also resembles ElasticGNN with $\lambda_2 = 0$ and $p = 2$, and the Total Variation GNN [19] also utilizes an ℓ_1 estimator in spectral clustering.

The above analyses suggest that SoftMedian, TWIRLS, and ElasticGNN share the same underlying design principle of ℓ_1-based robust graph signal estimation, i.e. a similar graph smoothing objective with edge difference penalties $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_1$ or $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2$. However, they adopt different approximation solutions that result in distinct architecture designs. This unified view of robust estimation clearly explains their closely aligned performance. Besides, the superiority ℓ_1-based models over the ℓ_2-based models such as GCN [16], whose graph smoothing objective is essentially $\sum_{(i,j)\in\mathcal{E}} \|\boldsymbol{f}_i/\sqrt{d_i} - \boldsymbol{f}_j/\sqrt{d_j}\|_2^2$ [20], can be explained since ℓ_1-based graph smoothing mitigates the impact of the outliers [15].

2.3 Bias Analysis and Performance Degradation

The unified view of ℓ_1-based graph smoothing we established in Section 2.2 not only explains their aligned robustness improvement but also provides a perspective to understand their failure as attack budgets scale up through an estimation bias analysis.

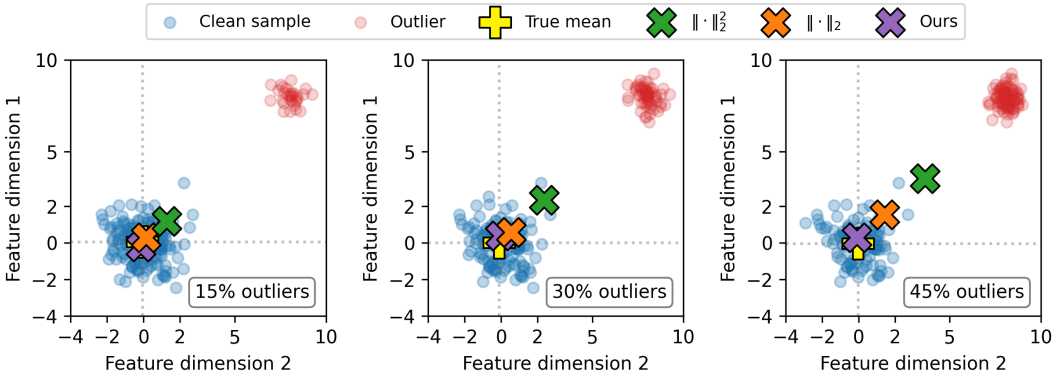

Figure 2: Different mean estimators in the presence of outliers. The clean samples are the majority of data points following the Gaussian distribution $\mathcal{N}((0,0), 1\cdot I)$, while the outliers are data points that deviate significantly from the main data pattern, following $\mathcal{N}((8,8), 0.5\cdot I)$. ℓ_2-estimator deviates far from the true mean, while the ℓ_1-based estimator is more resistant to outliers. However, as the ratio of outliers escalates, the ℓ_1-based estimator encounters a greater shift from the true mean, but our estimator still maintains a position close to the ground truth.

Bias of ℓ_1-based Estimation. In the literature of high-dimensional statistics, it has been well understood that the ℓ_1 regularization will induce an estimation bias. In the context of denoising [21] or variable selection [22], small coefficients β are undesirable. To exclude small β in the estimation, a soft-thresholding operator can be derived as $\mathbf{S}_\lambda(\beta) = \text{sign}(\beta)\max(|\beta| - \lambda, 0)$. As a result, large β are also shrunk by a constant, so the ℓ_1 estimation is biased towards zero.

A similar bias effect also occurs in graph signal estimation in the presence of adversarial attacks. For example, in TWIRLS (Eq. (1)), after the graph aggregation $\tilde{\boldsymbol{f}}_i^{(k+1)} = \sum_{j\in\mathcal{N}(i)} w_{ij}\tilde{\boldsymbol{f}}_j^{(k)}$ where $w_{ij} = \|\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j\|_2^{-1}$, the edge difference $\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j$ will shrink towards zero. Consequently, every adversarial edge the attacker adds will induce a bias that can be accumulated and amplified when the attack budget scales up.

Numerical Simulation. To provide a more intuitive illustration of the estimation bias of ℓ_1-based models, we simulate a mean estimation problem on synthetic data since most message passing schemes in GNNs essentially estimate the mean of neighboring node features. The results in Figure 2

shows that ℓ_1-based estimator is more resistant than ℓ_2-based estimator. However, as the ratio of outliers escalates, the ℓ_1-based estimator encounters a greater shift from the true mean due to the accumulated bias caused by outliers. This observation explains why ℓ_1-based graph smoothing models suffer from catastrophic degradation under large attack budgets. The detailed simulation settings and results are available in Appendix A.

3 Robust GNNs with Unbiased Aggregation

In this section, we first design a robust unbiased estimator to reduce the bias in graph signal estimation in Section 3.1 and propose an efficient second-order IRLS algorithm to compute the robust estimator with theoretical convergence guarantees in Section 3.2. Finally, we unroll the proposed algorithm as the robust unbiased feature aggregation layers in GNNs in Section 3.3.

3.1 Robust and Unbiased Graph Signal Estimator

Our study and analysis in Section 2 have shown that while ℓ_1-based methods outperform ℓ_2-based methods in robustness, they still suffer from the accumulated estimation bias, leading to severe performance degradation under large perturbation budgets. This motivates us to design a robust and unbiased graph signal estimator that derives unbiased robust aggregation for GNNs with stronger resilience to attacks.

Theoretically, the estimation bias in Lasso regression has been discovered and analyzed in high-dimensional statistics [23]. Statisticians have proposed adaptive Lasso [23] and many non-convex penalties such as Smoothly Clipped Absolute Deviation (SCAD) [24] and Minimax Concave Penalty (MCP) [25] to alleviate this bias. Motivated by these advancements, we propose a Robust and Unbiased Graph signal Estimator (RUGE) as follows:

$$\arg\min_{\boldsymbol{F}} \mathcal{H}(\boldsymbol{F}) = \sum_{(i,j)\in\mathcal{E}} \rho_\gamma\left(\left\|\frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}}\right\|_2\right) + \lambda \sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2, \tag{3}$$

where $\rho_\gamma(y)$ denotes the function that penalizes the feature differences on edges by MCP:

$$\rho_\gamma(y) = \begin{cases} y - \frac{y^2}{2\gamma} & \text{if } y < \gamma \\ \frac{\gamma}{2} & \text{if } y \geq \gamma \end{cases}. \tag{4}$$

As shown in Figure 3, MCP closely approximates the ℓ_1 norm when y is small since the quadratic term $\frac{y^2}{2\gamma}$ is negligible, and it becomes a constant value when y is large. This transition can be adjusted by the thresholding parameter γ. When γ approaches infinity, the penalty $\rho_\gamma(y)$ reduces to the ℓ_1 norm. Conversely, when γ is very small, the "valley" of ρ_γ near zero is exceptionally sharp, so $\rho_\gamma(y)$ approaches the ℓ_0 norm and becomes a constant for a slightly larger y. This enables RUGE to suppress smoothing on edges whose node differences exceeding the threshold γ. This not only mitigates the estimation bias against outliers but also maintains the estimation accuracy in the absence of outliers. The simulation in Figure 2 verifies that our proposed estimator ($\eta(\boldsymbol{x}) := \rho_\gamma(\|\boldsymbol{x}\|_2)$) can recover the true mean despite the increasing outlier ratio when the outlier ratio is below the theoretical optimal breakdown point.

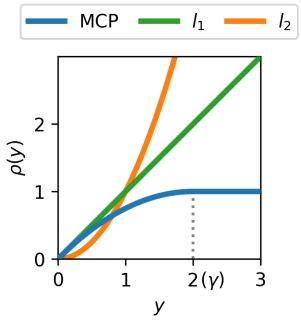

Figure 3: Penalties.

3.2 Quasi-Newton IRLS

Despite the advantages discussed above, the proposed RUGE in Eq. (3) is non-smooth and non-convex, which results in challenges for deriving efficient numerical solutions that can be readily unfolded as neural network layers. In the literature, researchers have developed optimization algorithms for MCP-related problems, such as the Alternating Direction Multiplier Method (ADMM) and Newton-type algorithms [24, 25, 26]. However, due to their excessive computation and memory requirements as well as the incompatibility with back-propagation training, these algorithms are not well-suited

for the construction of feature aggregation layers in GNNs. To solve these challenges, we propose an efficient Quasi-Newton Iteratively Reweighted Least Squares algorithm (QN-IRLS) to solve the estimation problem in Eq. (3).

IRLS. Before stepping into our QN-IRLS, we first introduce the main idea of iteratively reweighted least squares (IRLS) [27] and analyze its weakness in convergence. IRLS aims to circumvent the non-smooth $\mathcal{H}(\boldsymbol{F})$ in Eq. (3) by computing its quadratic upper bound $\hat{\mathcal{H}}$ based on $\boldsymbol{F}^{(k)}$ in each iteration k and optimizing $\hat{\mathcal{H}}(\mathcal{F})$:

$$\hat{\mathcal{H}}(\boldsymbol{F}) = \sum_{(i,j)\in\mathcal{E}} \boldsymbol{W}_{ij}^{(k)} \left\| \frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}} \right\|_2^2 + \lambda \sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2, \tag{5}$$

where $\boldsymbol{W}_{ij}^{(k)} = \mathbf{1}_{i\neq j} \frac{d\rho_\gamma(y_{ij})}{dy_{ij}^2}\big|_{y_{ij}=y_{ij}^{(k)}}$[3] , where $y_{ij}^{(k)} = \left\|\boldsymbol{f}_i^{(k)}/\sqrt{d_i} - \boldsymbol{f}_j^{(k)}/\sqrt{d_j}\right\|_2$ and $\rho_\gamma(\cdot)$ is the MCP function. For the detailed proof of the upper bound, please refer to Lemma 1 in Appendix B. Then, each iterative step of IRLS can be formulated as the first-order gradient descent for $\hat{\mathcal{H}}(\boldsymbol{F})$:

$$\boldsymbol{F}^{(k+1)} = \boldsymbol{F}^{(k)} - \eta\nabla\hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) = \boldsymbol{F}^{(k)} - \eta\left((\hat{\boldsymbol{Q}}^{(k)} - 2\boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}})\boldsymbol{F}^{(k)} - 2\lambda\boldsymbol{F}^{(0)}\right), \tag{6}$$

where η is the step size, $\hat{\boldsymbol{Q}}^{(k)} = 2(\mathrm{diag}(\boldsymbol{q}^{(k)}) + \lambda\boldsymbol{I})$, and $\boldsymbol{q}_m^{(k)} = \sum_j \boldsymbol{W}_{mj}^{(k)}\boldsymbol{A}_{mj}/d_m$. Its convergence condition is given in Theorem 1, with a proof in Appendix B.

Theorem 1. *If $\boldsymbol{F}^{(k)}$ follows the update rule in Eq.* (6) *where ρ defining $\boldsymbol{W}$ satisfies that $\frac{d\rho(y)}{dy^2}$ is non-decreasing $\forall y \in (0,\infty)$, then a sufficient condition for $\mathcal{H}(\boldsymbol{F}^{(k+1)}) \leq \mathcal{H}(\boldsymbol{F}^{(k)})$ is that the step size η satisfies $0 < \eta \leq \|diag(\boldsymbol{q}^{(k)}) - \boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}} + \lambda\boldsymbol{I}\|_2^{-1}$.*

Quasi-Newton IRLS. Theorem 1 suggests the difficulty in the proper selection of stepsize for (first-order) IRLS due to its non-trivial dependency on the graph ($\tilde{\boldsymbol{A}}$) and the dynamic terms ($\boldsymbol{q}^{(k)}$ and $\boldsymbol{W}^{(k)}$) [4]. The dilemma is that a small stepsize will lead to slow convergence but a large step easily causes divergence and instability as verified by our experiments in Section 4.3 (Figrue 5), which reveals its critical shortcoming for the construction of GNN layers.

To overcome this limitation, we aim to propose a second-order Newton method, $\boldsymbol{F}^{(k+1)} = \boldsymbol{F}^{(k)} - (\nabla^2\hat{\mathcal{H}}(\boldsymbol{F}^{(k)}))^{-1}\nabla\hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$, to achieve faster convergence and stepsize-free hyperparameter tuning by better capturing the geometry of the optimization landscape. However, obtaining the analytic expression for the inverse Hessian matrix $(\nabla^2\hat{\mathcal{H}}(\boldsymbol{F}^{(k)}))^{-1} \in \mathbb{R}^{n\times n}$ is intractable, and the numerical solution requires expensive computation for large graphs. Therefore, we propose a novel Quasi-Newton IRLS algorithm (QN-IRLS) that approximates the Hessian matrix $\nabla^2\hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) = 2(\mathrm{diag}(\boldsymbol{q}^{(k)}) - \boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}} + \lambda\boldsymbol{I})$ by the diagonal matrix $\hat{\boldsymbol{Q}}^{(k)} = 2(\mathrm{diag}(\boldsymbol{q}^{(k)}) + \lambda\boldsymbol{I})$ such that the inverse is trivial. The proposed QN-IRLS works as follows:

$$\boldsymbol{F}^{(k+1)} = \boldsymbol{F}^{(k)} - \left(\hat{\boldsymbol{Q}}^{(k)}\right)^{-1}\nabla\hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) = (\mathrm{diag}(\boldsymbol{q}^{(k)}) + \lambda\boldsymbol{I})^{-1}\left((\boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}})\boldsymbol{F}^{(k)} + \lambda\boldsymbol{F}^{(0)}\right), \tag{7}$$

where $(\hat{\boldsymbol{Q}}^{(k)})^{-1}$ automatically adjusts the per-coordinate stepsize according to the local geometry of the optimization landscape, $\boldsymbol{q}^{(k)}$ and $\boldsymbol{W}^{(k)}$ are defined as in Eq. (5) and (6). In this way, QN-IRLS provides faster convergence without needing to select a stepsize. The convergence is guaranteed by Theorem 2 with the proof in Appendix B.

Theorem 2. *If $\boldsymbol{F}^{(k+1)}$ follows update rule in Eq.* (7)*, where ρ satisfies that $\frac{d\rho(y)}{dy^2}$ is non-decreasing $\forall y \in (0,\infty)$, it is guaranteed that $\mathcal{H}(\boldsymbol{F}^{(k+1)}) \leq \mathcal{H}(\boldsymbol{F}^{(k)})$.*

3.3 GNN with Robust Unbiased Aggregation

[3] $\boldsymbol{W}_{ij}$ is defined as $\frac{d\rho(y)}{dy^2}\big|_{y=y_{ij}^{(k)}}$ so that the quadratic upper bound $\hat{\mathcal{H}}$ is tight at $\boldsymbol{F}^{(k)}$ according to Lemma 3. The diagonal terms of $\boldsymbol{W}$ are set to zero to avoid undefined derivative of $\frac{d\rho(y)}{dy^2}\big|_{y=0}$ as discussed in Remark 2.

[4] A related work [14] adopts IRLS algorithm to optimize the problem in Eq. (1). A preconditioned version is proposed to handle the unnormalized graph Laplacian, but its step size needs to satisfy $\eta \leq \|\Delta^\top\boldsymbol{\Gamma}^{(k)}\Delta + \lambda\boldsymbol{I}\|_2^{-1}$ as shown in Lemma 3.3 of [14], which is expensive to estimate.

The proposed QN-IRLS provides an efficient algorithm to optimize the RUGE in Eq. (3) with a theoretical convergence guarantee. Instantiated with $\rho = \rho_\gamma$, each iteration in QN-IRLS in Eq. (7) can be used as one layer in GNNs, which yields the Robust Unbiased Aggregation (RUNG):

$$\boldsymbol{F}^{(k+1)} = (\mathrm{diag}(\boldsymbol{q}^{(k)}) + \lambda \boldsymbol{I})^{-1} \left((\boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}}) \boldsymbol{F}^{(k)} + \lambda \boldsymbol{F}^{(0)} \right), \quad (8)$$

where $\boldsymbol{q}_m^{(k)} = \sum_j \boldsymbol{W}_{mj}^{(k)} \boldsymbol{A}_{mj} / d_m$ as in Eq. (6), $\boldsymbol{W}_{ij}^{(k)} = \mathbf{1}_{i \neq j} \max(0, \frac{1}{y_{ij}^{(k)}} - \frac{1}{\gamma})$ and $y_{ij}^{(k)} = \left\| \frac{\boldsymbol{f}_i^{(k)}}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j^{(k)}}{\sqrt{d_j}} \right\|_2$.

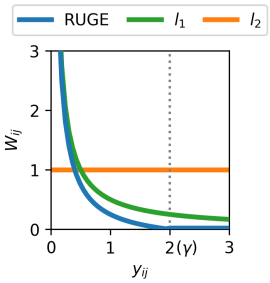

Figure 4: $\frac{d\rho(y)}{dy^2}$.

Interpretability. The proposed RUNG can be interpreted intuitively with edge reweighting. In Eq. (8), the normalized adjacency matrix $\tilde{\boldsymbol{A}}$ is reweighted by $\boldsymbol{W}^{(k)}$, where $\boldsymbol{W}_{ij}^{(k)} = \frac{d\rho(y)}{dy^2}|_{y=y_{ij}^{(k)}}$. It is shown in Figure 4 that $\boldsymbol{W}_{ij}$ becomes zero for any edge $e_k = (i, j)$ with a node difference $y_{ij}^{(k)} \geq \gamma$, thus pruning suspicious edges. This implies RUNG's strong robustness under large-budget adversarial attacks. With the inclusion of the skip connection $\boldsymbol{F}^{(0)}$, $\mathrm{diag}(\boldsymbol{q}^{(k)}) + \lambda \boldsymbol{I}$ can be seen as a normalizer of the layer output.

Relations with Existing GNNs. RUNG can adopt different ρ that Theorem 2 allows, thus covering many classic GNNs as special cases. When $\rho(y) = y^2$, RUNG in Eq. (8) exactly reduces to APPNP [28] ($\boldsymbol{F}^{(k+1)} = \frac{1}{1+\lambda} \tilde{\boldsymbol{A}} \boldsymbol{F}^{(k)} + \frac{\lambda}{1+\lambda} \boldsymbol{F}^{(0)}$) and GCN ($\boldsymbol{F}^{(k+1)} = \tilde{\boldsymbol{A}} \boldsymbol{F}^{(k)}$) if chosing $\lambda = 0$. When $\rho(y) = y$, the objective of RUGE is equivalent to ElasticGNN with $p = 2$, which is analogous to SoftMedian and TWIRLS due to their inherent connections as ℓ_1-based graph smoothing.

Complexity analysis. RUNG is scalable with time complexity of $O(k(m+n)d)$ and space complexity $O(m + nd)$, where m is the number of edges, d is the number of features, n is the number of nodes, and k is the number of GNN layers. Therefore, the complexity of our RUNG is comparable to normal GCN (with a constant difference) and it is feasible to implement. The detailed discussions about computation efficiency can be found in Appendix C.

4 Experiment

In this section, we perform comprehensive experiments to validate the robustness of the proposed RUNG. Besides, ablation studies show its convergence and defense mechanism.

4.1 Experiment Setting

Datasets. We test our RUNG with the node classification task on two widely used real-world citation networks, Cora ML and Citeseer [29], as well as a large-scale networks Ogbn-Arxiv [30]. We adopt the data split of 10% training, 10% validation, and 80% testing, and report the classification accuracy of the attacked nodes following [5]. Each experiment is averaged over 5 different random splits.

Baselines. To evaluate the performance of RUNG, we compare it to ℓ_2 other representative baselines. Among them, MLP, GCN [16], APPNP [28], and GAT [31] are undefended vanilla models. GNN-Guard [2], RGCN [11], GRAND [8], ProGNN [3], Jaccard-GCN [6], SVD-GCN [7], EvenNet [32], HANG [13], NoisyGNN [33], and GARNET [34] are representative robust GNNs. Besides, SoftMedian and TWIRLS are representative methods with ℓ_1-based graph smoothing [5]. We also evaluate a variant of TWIRLS with thresholding attention (TWIRLS-T). For RUNG, we test two variants: default RUNG (Eq. (8)) and RUNG-ℓ_1 with ℓ_1 penalty ($\rho(y) = y$).

Hyperparameters. The model hyperparameters including learning rate, weight decay, and dropout rate are tuned as in [5]. Other hyperparameters follow the settings in the original papers. RUNG uses an MLP connected to 10 graph aggregation layers following the decoupled GNN architecture of APPNP. $\hat{\lambda} = \frac{1}{1+\lambda}$ is tuned in {0.7, 0.8, 0.9}, and γ tuned in $\{0.5, 1, 2, 3, 5\}$. We chose the

[5] We do not include ElasticGNN because it is still unclear how to attack it adaptively due to its special incident matrix formulation [15]. In the preliminary study (Section 2.1), we evaluate the robustness of ElasticGNN following the unit test setting proposed in [5].

hyperparameter setting that yields the best robustness without a notable impact (smaller than 1%) on the clean accuracy following [35].

Attack setting. We use the PGD attack [36] to execute the adaptive *evasion* and *poisoning* topology attack since it delivers the strongest attack in most settings [5]. The adaptive attack setting is provided in Appendix F. The adversarial attacks aim to misclassify specific target nodes (*local attack*) or the entire set of test nodes (*global attack*). To avoid a false sense of robustness, our *adaptive* attacks directly target the victim model instead of the surrogate model. Additionally, we include the *transfer* attacks with a 2-layer GCN as the surrogate model. We also include graph injection attack following the setting in TDGIA [37].

4.2 Adversarial Robustness

Table 1: Adaptive local attack on Cora ML. The **best** and second are marked.

Model	0% (Clean)	20%	50%	100%	150%	200%
MLP	72.6 ± 6.4	72.6 ± 6.4	72.6 ± 6.4	$\mathbf{72.6 \pm 6.4}$	$\mathbf{72.6 \pm 6.4}$	$\mathbf{72.6 \pm 6.4}$
GCN	82.7 ± 4.9	40.7 ± 10.2	12.0 ± 6.2	2.7 ± 2.5	0.0 ± 0.0	0.0 ± 0.0
APPNP	$\mathbf{84.7 \pm 6.8}$	50.0 ± 13.0	27.3 ± 6.5	14.0 ± 5.3	3.3 ± 3.0	0.7 ± 1.3
GAT	80.7 ± 10.0	30.7 ± 16.1	16.0 ± 12.2	11.3 ± 4.5	1.3 ± 1.6	2.0 ± 1.6
GNNGuard	82.7 ± 6.7	44.0 ± 11.6	30.7 ± 11.6	14.0 ± 6.8	5.3 ± 3.4	2.0 ± 2.7
RGCN	84.6 ± 4.0	46.0 ± 9.3	18.0 ± 8.1	6.0 ± 3.9	0.0 ± 0.0	0.0 ± 0.0
GRAND	84.0 ± 6.8	47.3 ± 9.0	18.7 ± 9.1	7.3 ± 4.9	1.3 ± 1.6	0.0 ± 0.0
ProGNN	$\mathbf{84.7 \pm 6.2}$	47.3 ± 10.4	21.3 ± 7.8	4.0 ± 2.5	0.0 ± 0.0	0.0 ± 0.0
Jaccard-GCN	81.3 ± 5.0	46.0 ± 6.8	17.3 ± 4.9	4.7 ± 3.4	0.7 ± 1.3	0.7 ± 1.3
GARNET	82.4 ± 6.8	70.9 ± 7.5	61.9 ± 7.9	42.7 ± 9.3	11.6 ± 3.4	9.6 ± 3.5
HANG	83.1 ± 7.4	71.2 ± 7.8	60.1 ± 6.3	39.5 ± 3.4	9.4 ± 2.3	5.6 ± 2.3
NoisyGNN	82.5 ± 5.6	57.4 ± 5.7	47.8 ± 6.2	36.1 ± 4.5	5.8 ± 3.4	4.1 ± 1.2
EvenNet	83.4 ± 8.1	64.8 ± 6.9	56.1 ± 5.6	29.5 ± 5.3	3.1 ± 1.1	1.1 ± 1.3
GraphCON	81.3 ± 7.1	67.3 ± 7.3	54.7 ± 7.1	41.3 ± 6.2	4.1 ± 2.1	2.3 ± 1.2
SoftMedian	80.0 ± 10.2	72.7 ± 13.7	62.7 ± 12.7	46.7 ± 11.0	8.0 ± 4.5	8.7 ± 3.4
TWIRLS	83.3 ± 7.3	71.3 ± 8.6	60.7 ± 11.0	36.0 ± 8.8	20.7 ± 10.4	12.0 ± 6.9
TWIRLS-T	82.0 ± 4.5	70.7 ± 4.4	62.7 ± 7.4	54.7 ± 6.2	44.0 ± 11.2	40.7 ± 11.8
RUNG-ℓ_1 (Ours)	84.0 ± 6.8	72.7 ± 7.1	62.7 ± 11.2	53.3 ± 8.2	22.0 ± 9.3	14.0 ± 7.4
RUNG (Ours)	84.0 ± 5.3	$\mathbf{75.3 \pm 6.9}$	$\mathbf{72.7 \pm 8.5}$	70.7 ± 10.6	69.3 ± 9.8	69.3 ± 9.0

Table 2: Adaptive global attack on Cora ML. The **best** and second are marked.

Model	0% (Clean)	5%	10%	20%	30%	40%
MLP	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0
GCN	85.0 ± 0.4	75.3 ± 0.5	69.6 ± 0.5	60.9 ± 0.7	54.2 ± 0.6	48.4 ± 0.5
APPNP	$\mathbf{86.3 \pm 0.4}$	75.8 ± 0.5	69.7 ± 0.7	60.3 ± 0.9	53.8 ± 1.2	49.0 ± 1.6
GAT	83.5 ± 0.5	75.8 ± 0.8	71.2 ± 1.2	65.0 ± 0.9	60.5 ± 0.9	56.7 ± 0.9
GNNGuard	83.1 ± 0.7	74.6 ± 0.7	70.2 ± 1.0	63.1 ± 1.1	57.5 ± 1.6	51.0 ± 1.2
RGCN	85.7 ± 0.4	75.0 ± 0.8	69.1 ± 0.4	59.8 ± 0.7	52.8 ± 0.7	46.1 ± 0.7
GRAND	86.1 ± 0.7	76.2 ± 0.8	70.7 ± 0.7	61.6 ± 0.7	56.7 ± 0.8	51.9 ± 0.9
ProGNN	85.6 ± 0.5	76.5 ± 0.7	71.0 ± 0.5	63.0 ± 0.7	56.8 ± 0.7	51.3 ± 0.6
Jaccard-GCN	83.7 ± 0.7	73.9 ± 0.5	68.3 ± 0.7	60.0 ± 1.1	54.0 ± 1.7	49.1 ± 2.4
GARNET	84.0 ± 0.5	76.5 ± 0.4	72.1 ± 0.1	66.4 ± 0.7	62.1 ± 1.3	58.7 ± 1.5
HANG	84.5 ± 0.2	75.7 ± 0.6	74.2 ± 0.3	69.5 ± 0.4	64.7 ± 0.9	58.4 ± 0.1
NoisyGNN	83.9 ± 0.5	76.7 ± 0.1	72.1 ± 0.3	64.7 ± 0.7	58.0 ± 0.2	53.3 ± 0.6
EvenNet	84.8 ± 0.9	75.8 ± 0.9	70.7 ± 0.6	63.9 ± 0.5	58.7 ± 0.7	54.7 ± 0.8
GraphCON	83.7 ± 0.6	75.8 ± 0.3	70.9 ± 0.8	65.9 ± 0.7	62.2 ± 0.9	56.6 ± 0.3
SoftMedian	85.0 ± 0.7	78.6 ± 0.3	75.5 ± 0.9	69.5 ± 0.5	62.8 ± 0.8	58.1 ± 0.7
TWIRLS	84.2 ± 0.6	77.3 ± 0.8	72.9 ± 0.3	66.9 ± 0.2	62.4 ± 0.6	58.7 ± 1.1
TWIRLS-T	82.8 ± 0.5	76.8 ± 0.6	73.2 ± 0.4	67.7 ± 0.4	63.8 ± 0.2	60.8 ± 0.3
RUNG-ℓ_1 (Ours)	85.8 ± 0.5	78.4 ± 0.4	74.3 ± 0.3	68.1 ± 0.6	63.5 ± 0.7	59.8 ± 0.8
RUNG (Ours)	84.6 ± 0.5	$\mathbf{78.9 \pm 0.4}$	$\mathbf{75.7 \pm 0.2}$	$\mathbf{71.8 \pm 0.4}$	$\mathbf{67.8 \pm 1.3}$	$\mathbf{65.1 \pm 1.2}$

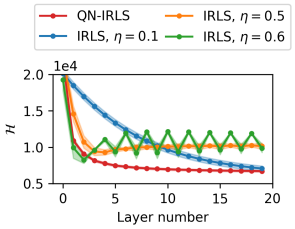

Figure 5: Convergence of our QN-IRLS compared to IRLS.

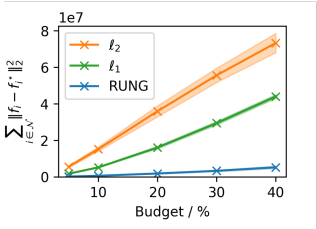

Figure 6: Bias induced by different attack budgets.

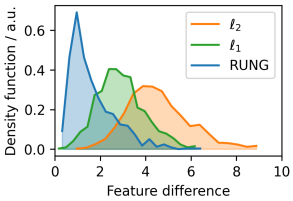

Figure 7: Distribution of feature difference on attacked edges.

Here we evaluate the the performance of RUNG against the baselines under different settings. The results of local and global adaptive attacks on Cora ML are presented in Table 1 and Table 2, while those on Citeseer are presented in Table 3 and Table 4 in Appendix E due to space limits. We summarize the following analysis from Cora ML, noting that the same observations apply to Citeseer.

- Under adaptive attacks, many existing defenses are not significantly more robust than undefended models. The ℓ_1-based models such as TWIRLS, SoftMedian, and RUNG-ℓ_1 demonstrate considerable and closely aligned robustness under both local and global attacks, which supports our unified ℓ_1-based robust view analysis in Section 2.2.
- RUNG exhibits significant improvements over all baselines across various budgets under both global and local attacks. Local attacks are stronger than global attacks since local attacks concentrate on targeted nodes. The robustness improvement of RUNG appears to be more remarkable in local attacks.
- When there is no attack, RUNG largely preserves an excellent clean performance. RUNG also achieves state-of-the-art performance under small attack budgets.

4.3 Ablation study

Convergence. To verify the advantage of our QN-IRLS method in Eq (7) over the first-order IRLS in Eq (6), we show the objective $\mathcal{H}$ on each layer in Figure 5. It can be observed that our stepsize-free QN-IRLS demonstrates the best convergence as discussed in Section 3.

Estimation bias. The bias effect in ℓ_1-based GNNs and the unbiasedness of our RUNG can be empirically verified. We quantify the bias with $\sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^\star\|_2^2$, where $\boldsymbol{f}_i^\star$ and $\boldsymbol{f}_i$ denote the aggregated feature on the clean graph and attacked graph, respectively. As shown in Figure 6, when the budget scales up, ℓ_1 GNNs exhibit a notable bias, whereas RUNG has nearly zero bias. We provide comprehensive discussion of unbiasedness of RUNG in Appendix D.

Defense Mechanism To further investigate how our defense takes effect, we analyze the edges added under adaptive attacks. The distribution of the node feature differences $\left\|\boldsymbol{f}_i/\sqrt{d_i} - \boldsymbol{f}_j/\sqrt{d_j}\right\|_2$ of attacked edges is shown in Figure 7 for different graph signal estimators. It can be observed that our RUNG forces the attacker to focus on the edges with a small feature difference, indicating that our RUNG can improve robustness by down-weighting or pruning some malicious edges that connect distinct nodes. Therefore, the attacks become less influential, which explains why RUNG demonstrates outstanding robustness.

Transfer Attacks. In addition to the adaptive attack, we also conduct a set of transfer attacks that take every baseline GNN as the surrogate model to comprehensively test the robustness of RUNG, following the unit test attack protocol proposed in [5]. We summarize the results on Cora ML and Citeseer in Figure 9 and Figure 10 in Appendix E due to space limits. All transfer attacks are weaker than the adaptive attack in Section 4.2, indicating the necessity to evaluate the strongest adaptive attack to avoid the false sense of security emphasized in this paper. Note that the attack transferred from RUNG model is slightly weaker than the adaptive attack since the surrogate and victim RUNG models have different model parameters in the transfer attack setting.

Hyperparameters. Due to the space limit, we provide the additional ablation studies on the hyperparameters (including γ and λ in MCP as well as the number of layers) of RUNG in Appendix G.

The results offer an overview strategy for the choice of optimal hyperparameters. We can observe that γ in MCP has a significant impact on the performance of RUNG. Specifically, a larger γ makes RUNG closer to an ℓ_1-based model, while a smaller γ encourages more edges to be pruned. This pruning helps RUNG to remove more malicious edges and improve robustness, although a small γ may slightly reduce clean performance.

Robustness under various scenarios. Besides the evaluation under strong adaptive attacks, we also validate the consistent effectiveness of our proposed RUNG under various scenarios, including *Transfer attacks* (Appendix E.2), *Poisoning attacks* (Appendix E.3), *Large scale datasets* (Appendix E.4), *Adversarial training* (Appendix E.5), *Graph injection attacks* (Appendix E.6).

5 Related Work

To the best of our knowledge, although there are works unifying existing GNNs from a graph signal denoising perspective [20], no work has been dedicated to uniformly understand the robustness and limitations of robust GNNs such as SoftMedian [4], SoftMedoid [18], TWIRLS [14], ElasticGNN [15], and TVGNN [19] from the ℓ_1 robust statistics and bias analysis perspectives. To mitigate the estimation bias, MCP penalty is promising since it is well known for its near unbiasedness property [25] and has been applied to the graph trend filtering problem [26] to promote piecewise signal modeling, but their robustness is unexplored. Nevertheless, other robust GNNs have utilized alternative penalties that might alleviate the bias effect. For example, GNNGuard [2] prunes the edges whose cosine similarity is too small. Another example is that TWIRLS [14] with a thresholding penalty can also exclude edges using graph attention. However, the design of their edge weighting or graph attention is heuristic-based and exhibits suboptimal performance compared to the RUNG proposed in this work.

6 Conclusion & Limitation

In this work, we propose a unified view of ℓ_1 robust graph smoothing to uniformly understand the robustness and limitations of representative robust GNNs. The established view not only justifies their improved and closely aligned robustness but also explains their severe performance degradation under large attack budgets by a novel estimation bias analysis. To mitigate the estimation bias, we propose a robust and unbiased graph signal estimator. To solve this non-trivial estimation problem, we design a novel and efficient Quasi-Newton IRLS algorithm that can better capture the landscape of the optimization problem and converge stably with a theoretical guarantee. This algorithm can be unfolded and used as a building block for constructing robust GNNs with Robust Unbiased Aggregation (RUNG). As verified by our experiments, RUNG provides the best performance under strong adaptive attacks among all the baselines. Furthermore, RUNG also covers many classic GNNs as special cases. Most importantly, this work provides a deeper understanding of existing approaches and reveals a principled direction for designing robust GNNs.

Regarding the limitations, first, the improvement of RUNG is more significant under large budgets compared to the robust baselines. Second, we primarily include experiments on homophilic graphs in the main paper, but we can generalize the proposed robust aggregation to heterophilic graphs in future work. Third, although our Quasi-Newton IRLS algorithm has exhibited excellent convergence compared to the vanilla IRLS, the efficiency of RUNG could be further improved.

Acknowledgment

Zhichao Hou and Dr. Xiaorui Liu are supported by the National Science Foundation (NSF) National AI Research Resource Pilot Award, Amazon Research Award, NCSU Data Science Academy Seed Grant Award, and NCSU Faculty Research and Professional Development Award.

References

[1] Yao Ma and Jiliang Tang. *Deep learning on graphs*. Cambridge University Press, 2021.

[2] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

[3] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[4] Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7637–7649. Curran Associates, Inc., 2021.

[5] Felix Mujkanovic, Simon Geisler, Stephan Günnemann, and Aleksandar Bojchevski. Are defenses for graph neural networks robust? In *Neural Information Processing Systems, NeurIPS*, 2022.

[6] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples on graph data: Deep insights into attack and defense. *arXiv preprint arXiv:1903.01610*, 2019.

[7] Negin Entezari, Saba A Al-Sayouri, Amirali Darvishzadeh, and Evangelos E Papalexakis. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 169–177, 2020.

[8] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33:22092–22103, 2020.

[9] Zhijie Deng, Yinpeng Dong, and Jun Zhu. Batch virtual adversarial training for graph convolutional networks. *AI Open*, 2023.

[10] Jinyin Chen, Xiang Lin, Hui Xiong, Yangyang Wu, Haibin Zheng, and Qi Xuan. Smoothing adversarial training for gnn. *IEEE Transactions on Computational Social Systems*, 8(3):618–629, 2020.

[11] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1399–1407, 2019.

[12] T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-coupled oscillator networks. In *International Conference on Machine Learning*, pages 18888–18909. PMLR, 2022.

[13] Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Adversarial robustness in graph neural networks: A hamiltonian approach. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Yongyi Yang, Tang Liu, Yangkun Wang, Jinjing Zhou, Quan Gan, Zhewei Wei, Zheng Zhang, Zengfeng Huang, and David Wipf. Graph neural networks inspired by classical iterative algorithms. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11773–11783. PMLR, 18–24 Jul 2021.

[15] Xiaorui Liu, Wei Jin, Yao Ma, Yaxin Li, Hua Liu, Yiqi Wang, Ming Yan, and Jiliang Tang. Elastic graph neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6837–6849. PMLR, 18–24 Jul 2021.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[17] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[18] Simon Geisler, Daniel Zügner, and Stephan Günnemann. Reliable graph neural networks via robust aggregation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[19] Jonas Berg Hansen and Filippo Maria Bianchi. Total variation graph neural networks. In *Proceedings of the 40th international conference on Machine learning*. ACM, 2023.

[20] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. A unified view on graph neural networks as graph signal denoising. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 1202–1211, New York, NY, USA, 2021. Association for Computing Machinery.

[21] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[22] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[23] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[24] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[25] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894 – 942, 2010.

[26] Rohan Varma, Harlin Lee, Jelena Kovačević, and Yuejie Chi. Vector-valued graph trend filtering with non-convex penalties. *IEEE transactions on signal and information processing over networks*, 6:48–62, 2019.

[27] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[28] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

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

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

[31] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[32] Runlin Lei, Zhen Wang, Yaliang Li, Bolin Ding, and Zhewei Wei. Evennet: Ignoring odd-hop neighbors improves robustness of graph neural networks. *Advances in Neural Information Processing Systems*, 35:4694–4706, 2022.

[33] Sofiane Ennadir, Yassine Abbahaddou, Johannes F Lutzeyer, Michalis Vazirgiannis, and Henrik Boström. A simple and yet fairly effective defense for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21063–21071, 2024.

[34] Chenhui Deng, Xiuyu Li, Zhuo Feng, and Zhiru Zhang. Garnet: Reduced-rank topology learning for robust and scalable graph neural networks. In *Learning on Graphs Conference*, pages 3–1. PMLR, 2022.

[35] Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations. In *Neural Information Processing Systems*, 2019.

[36] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: an optimization perspective. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3961–3967, 2019.

[37] Xu Zou, Qinkai Zheng, Yuxiao Dong, Xinyu Guan, Evgeny Kharlamov, Jialiang Lu, and Jie Tang. Tdgia: Effective injection attacks on graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2461–2471, 2021.

[38] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, Dec 2008. Funding by NSF.

[39] Amir Beck and Shoham Sabach. Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, 164(1):1–40, 2015.

[40] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[41] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. 2010.

[42] Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. *Advances in Neural Information Processing Systems*, 34:7637–7649, 2021.

A Bias Accumulation of ℓ_1 Models

A.1 Details of the Numerical Simulation Settings

To provide a more intuitive illustration of the estimation biases of different models, we simulate a mean estimation problem on synthetic data since most message passing schemes in GNNs essentially estimate the mean of neighboring node features. In the context of mean estimation, the bias is measured as the distances between different mean estimators and the true mean. We firstly generated clean samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ (blue dots) and the outlier samples $\{\boldsymbol{x}_i\}_{i=n+1}^{n+m}$(red dots) from 2-dimensional Gaussian distributions, $\mathcal{N}((0,0),1)$ and $\mathcal{N}((8,8),0.5)$, respectively. We calculate the mean of clean samples $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$ as the ground truth of the mean estimator. Then we estimate the mean of all the samples by solving $\arg\min_{\boldsymbol{z}}\sum_{i=1}^{n+m}\eta(\boldsymbol{z}-\boldsymbol{x}_i)$ using the Weiszfeld method [38, 39], where $\eta(\cdot)$ can take different norms such as ℓ_2 norm $\|\cdot\|_2^2$ and ℓ_1 norm $\|\cdot\|_2$.

The mean estimators are formulated as minimization operators

$$\bar{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \sum_{i}^{n+m} \eta(\boldsymbol{z}-\boldsymbol{x}_i), \tag{9}$$

where n is the number of clean samples and m is the number of adversarial samples.

ℓ_1 **estimator.** The ℓ_1 estimator ($\eta(\boldsymbol{y}) \coloneqq \|\boldsymbol{y}\|_2$), essentially is the geometric median. We adopted the Weiszfeld method to iteratively reweight $\boldsymbol{z}$ to minimize the objective, following

$$\boldsymbol{z}^{(k+1)} = \frac{\sum_i w_i^{(k)}\boldsymbol{x}_i}{\sum_i w_i^{(k)}}, \tag{10}$$

where $w_i^{(k)} = \frac{1}{\|\boldsymbol{z}^{(k)}-\boldsymbol{x}_i\|_2}$. This can be seen as a gradient descent step of $\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \alpha\nabla_{\boldsymbol{z}}\sum_i\|\boldsymbol{z}-\boldsymbol{x}_i\|_2 = \boldsymbol{z}^{(k+1)} - \alpha\sum_i\frac{\boldsymbol{z}^{(k)}-\boldsymbol{x}_i}{\|\boldsymbol{z}^{(k)}-\boldsymbol{x}_i\|_2}$. Taking $\alpha = \frac{1}{\sum_i w_i^{(k)}}$ instantly yields Eq. (10).

MCP-based estimator. We therefore adopt a similar approach for the MCP-based estimator ("Ours" in Fig. Figure 2), where $\eta(\boldsymbol{y}) \coloneqq \rho_\gamma(\boldsymbol{y})$:

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \alpha\nabla_{\boldsymbol{z}}\sum_i\rho_\gamma(\|\boldsymbol{z}-\boldsymbol{x}_i\|_2) \tag{11}$$

$$= \boldsymbol{z}^{(k)} - \alpha\sum_i\max(0, \frac{1}{\|\boldsymbol{z}^{(k)}-\boldsymbol{x}_i\|_2} - \frac{1}{\gamma})(\boldsymbol{z}^{(k)}-\boldsymbol{x}_i). \tag{12}$$

Denoting $\max(0, \|\boldsymbol{z}^{(k)}-\boldsymbol{x}_i\|_2^{-1} - \frac{1}{\gamma})$ as w_i, and then $\alpha = \frac{1}{\sum_i w_i}$ yields a similar reweighting iteration $\boldsymbol{z}^{(k+1)} = \frac{\sum_i w_i^{(k)}\boldsymbol{x}_i}{\sum_i w_i^{(k)}}$.

ℓ_2 **estimator.** It is worth noting that the same technique can be applied to the ℓ_2 estimator with $\rho(\boldsymbol{z}) \coloneqq \|\boldsymbol{z}\|_2^2$. The iteration becomes

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \alpha\nabla_{\boldsymbol{z}}\sum_i\|\boldsymbol{z}-\boldsymbol{x}_i\|_2^2 \tag{13}$$

$$= \boldsymbol{z}^{(k)} - \alpha\sum_i(\boldsymbol{z}^{(k)}-\boldsymbol{x}_i), \tag{14}$$

and $\alpha = \frac{1}{n+m}$ yields $\boldsymbol{z}^{(k+1)} = \frac{1}{n+m}\sum_i\boldsymbol{x}_i$, which gives the mean of all samples in one single iteration.

Similarities between this mean estimation scenario and our QN-IRLS in graph smoothing can be observed, both of which involve iterative reweighting to estimate similar objectives. The approximated Hessian in our QN-IRLS resembles the Weiszfeld method, canceling the $\boldsymbol{z}^{(k)}$ by tuning the stepsize.

In Figure 2, we visualize the generated clean samples and outliers, as well as the ground truth means and the mean estimators with $\eta(\cdot) = \|\cdot\|_2^2$ or $\|\cdot\|_2$ under different outlier ratios (15%,

30%, 45%). The results show that the ℓ_2-based estimator deviates far from the true mean, while the ℓ_1-based estimator is more resistant to outliers, which explains why ℓ_1-based methods exhibit stronger robustness. However, as the ratio of outliers escalates, the ℓ_1-based estimator encounters a greater shift from the true mean due to the accumulated bias caused by outliers. This observation explains why ℓ_1-based graph smoothing models suffer from catastrophic degradation under large attack budgets. Our estimator keeps much closer to the ground truth than other estimators with the existence of outliers.

A.2 Additional Simulation Results and Discussions

Here, we complement Figure 2 with the settings of higher attack budgets. As the outlier ratio exceeds the breakdown point 50%, we observe that our MCP-based mean estimator can correctly recover the majority of the samples, i.e. converge to the center of "outliers".

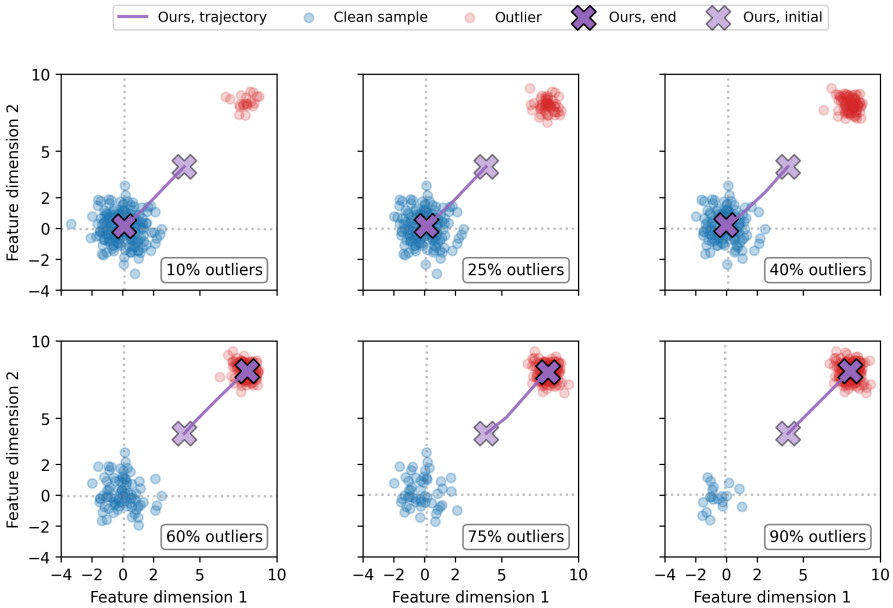

Figure 8: The trajectory of our MCP-based mean estimator.

B Convergence Analysis

To begin with, we will provide an overview of our proof, followed by a detailed presentation of the formal proof for the convergence analysis.

Overview of proof. First, for both IRLS and QN-IRLS, we construct, for $\boldsymbol{F}^{(k)}$ in every iteration k, a quadratic upper bound $\hat{\mathcal{H}}$ that satisfies $\hat{\mathcal{H}} + C \geq \mathcal{H}$ where the equality is reached at $\boldsymbol{F}^{(k)}$. Then we can minimize $\hat{\mathcal{H}}$ to guarantee the iterative descent of $\mathcal{H}$ since $\mathcal{H}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) + C \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) + C = \mathcal{H}(\boldsymbol{F}^{(k)})$.

To find the $\boldsymbol{F}^{(k+1)}$ such that $\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$, IRLS simply adopts the plain gradient descent $\boldsymbol{F}^{(k+1)} = \boldsymbol{F}^{(k)} - \eta \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$ whose convergence condition can be analyzed with the β-smoothness of the quadratic $\hat{\mathcal{H}}$ (Theorem 1). To address the problems of IRLS as motivated in Section 3.2, our Quasi-Newton IRLS utilizes the diagonal approximate Hessian $\hat{\boldsymbol{Q}}$ to scale the update step size in different dimensions respectively as $\boldsymbol{F}^{(k+1)} = \boldsymbol{F}^{(k)} - \hat{\boldsymbol{Q}}^{-1} \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$. Thereafter, by bounding the Hessian with $2\hat{\boldsymbol{Q}}$, the descent condition of $\hat{\mathcal{H}}$ is simplified (Theorem 2).

Lemma 1. *For any $\rho(y)$ satisfying $\frac{d\rho(y)}{dy^2}$ is non-increasing, denote $y_{ij} := \left\| \frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}} \right\|_2$, then $\mathcal{H}(\boldsymbol{F}) = \sum_{(i,j)\in\mathcal{E}, i\neq j} \rho(y_{ij}) + \lambda \sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2$ has the following upper bound:*

$$\mathcal{H}(\boldsymbol{F}) \leq \hat{\mathcal{H}}(\boldsymbol{F}) + C = \sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij}^{(k)} y_{ij}^2 + \lambda \sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2 + C, \tag{15}$$

where $\boldsymbol{W}_{ij}^{(k)} = \frac{\partial \rho(y)}{\partial y^2}\big|_{y=y_{ij}^{(k)}}$ and $y_{ij}^{(k)} = \left\| \frac{\boldsymbol{f}_i^{(k)}}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j^{(k)}}{\sqrt{d_j}} \right\|_2$ and $C = \mathcal{H}(\boldsymbol{F}^{(k)}) - \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$ is a constant. The equality in Eq. (15) *is achieved when $\boldsymbol{F} = \boldsymbol{F}^{(k)}$.*

Proof. Let $v = y^2$ and define $\psi(v) := \rho(y) = \rho(\sqrt{v})$. Then ψ is concave since $\frac{d\psi(v)}{dv} = \frac{d\rho(y)}{dy^2}$ is non-increasing. According to the concavity property, we have $\psi(v) \leq \psi(v_0) + \psi'(\nu)\big|_{\nu=v_0}(v - v_0)$. Substitute $v = y^2, v_0 = y_0^2$, we obtain:

$$\rho(y) \leq y^2 \frac{\partial \rho(y)}{\partial y^2}\big|_{y=y_0} - y_0^2 \frac{\partial \rho(y)}{\partial y^2}\big|_{y=y_0} + \rho(y_0) \tag{16}$$

$$= y^2 \frac{\partial \rho(y)}{\partial y^2}\big|_{y=y_0} + C(y_0) \tag{17}$$

where the inequality is reached when $y = y_0$. Next, substitute $y = y_{ij}$ and $y_0 = y_{ij}^{(k)}$, we can get $\rho(y_{ij}) \leq \boldsymbol{W}_{ij}^{(k)} y_{ij}^2 + C(y_{ij}^{(k)})$ which takes the equality at $y_{ij} = y_{ij}^{(k)}$. Finally, by summing up both sides and add a regularization term, we can prove the Eq. (15). □

Remark 1. *It can be seen that the definition of $\hat{\mathcal{H}}$ depends on $\boldsymbol{F}^{(k)}$, which ensures that the bound is tight when $\boldsymbol{F} = \boldsymbol{F}^{(k)}$. This tight bound condition is essential in the majorization-minimization algorithm as seen in Theorem 1.*

Lemma 2. *For $\hat{\mathcal{H}} = \sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} y_{ij}^2 + \lambda \sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2$, the gradient and Hessian w.r.t. $\boldsymbol{F}$*[6] *satisfy*

$$\nabla_{\boldsymbol{F}_{mn}} \hat{\mathcal{H}}(\boldsymbol{F}) = 2\left((diag(\boldsymbol{q}) - \boldsymbol{W} \odot \tilde{\boldsymbol{A}} + \lambda \boldsymbol{I})\boldsymbol{F} - \lambda \boldsymbol{F}^{(0)} \right)_{mn}, \tag{18}$$

and

$$\nabla_{\boldsymbol{F}_{mn}} \nabla_{\boldsymbol{F}_{kl}} \hat{\mathcal{H}}(\boldsymbol{F}) = 2\left(diag(\boldsymbol{q}) - \boldsymbol{W} \odot \tilde{\boldsymbol{A}} + \lambda \boldsymbol{I} \right)_{mk}, \tag{19}$$

where $\boldsymbol{q}_m = \sum_j \boldsymbol{W}_{mj} \boldsymbol{A}_{mj} / d_m$ and $\tilde{\boldsymbol{A}}_{ij} = \frac{\boldsymbol{A}_{ij}}{\sqrt{d_i d_j}}$ is the symmetrically normalized adjacency matrix.

[6] Here are some explanations on the tensor 'Hessian' $\nabla^2 \hat{\mathcal{H}}$. Since $\hat{\mathcal{H}}(\boldsymbol{F})$ is dependent on a matrix, there are some difficulties in defining the Hessian. However, as can be observed in Eq. (27) and Eq. (32), the feature dimension can be accounted for by the following. Initially, we treat the feature dimension as an irrelevant dimension that is excluded from the matrix operations. E.g., $\boldsymbol{F}\nabla^2\hat{\mathcal{H}}\boldsymbol{F} = \sum_{ik} \boldsymbol{F}_{ij} \nabla^2 \hat{\mathcal{H}}_{ijkl} \boldsymbol{F}_{kl}$ where the feature dimensions j and l remain free indices while the node indices i and k are eliminated as dummy indices. Finally, we take the trace of the resulting #feature×#feature matrix to get the desired value.

Proof. Follow $\boldsymbol{A} = \boldsymbol{A}^\top$ and define $y_{ij}^2 := \left\| \frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}} \right\|_2^2$, then the first-order gradient of $\sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} y_{ij}^2$ will be

$$\nabla_{\boldsymbol{F}_{mn}} \left(\sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} y_{ij}^2 \right) \tag{20}$$

$$= \sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} \frac{\partial y_{ij}^2}{\partial \boldsymbol{F}_{mn}} \tag{21}$$

$$= \sum_{(m,j)\in\mathcal{E}} \boldsymbol{W}_{mj} \frac{\partial y_{mj}^2}{\partial \boldsymbol{F}_{mn}} \tag{22}$$

$$= \sum_{(m,j)\in\mathcal{E}} \boldsymbol{W}_{mj} \frac{\partial \left(\frac{\boldsymbol{F}_{mn}}{\sqrt{d_m}} - \frac{\boldsymbol{F}_{jn}}{\sqrt{d_j}} \right)^2}{\partial \boldsymbol{F}_{mn}} \tag{23}$$

$$= \sum_{j\in\mathcal{N}(m)} 2\boldsymbol{W}_{mj} \left(\frac{\boldsymbol{F}_{mn}}{d_m} - \frac{\boldsymbol{F}_{jn}}{\sqrt{d_m d_j}} \right) \tag{24}$$

$$= 2\sum_j \boldsymbol{W}_{mj} \left(\frac{\boldsymbol{A}_{mj}}{d_m} \boldsymbol{F}_{mn} - \frac{\boldsymbol{A}_{mj}}{\sqrt{d_m d_j}} \boldsymbol{F}_{jn} \right) \tag{25}$$

$$= 2\left(\frac{\sum_j \boldsymbol{W}_{mj} \boldsymbol{A}_{mj}}{d_m} \right) \boldsymbol{F}_{mn} - 2((\boldsymbol{W} \odot \tilde{\boldsymbol{A}})\boldsymbol{F})_{mn} \tag{26}$$

$$= \left(2(\mathrm{diag}(\boldsymbol{q}) - \boldsymbol{W} \odot \tilde{\boldsymbol{A}})\boldsymbol{F} \right)_{mn}, \tag{27}$$

and the second-order hessian will be:

$$\nabla^2_{\boldsymbol{F}_{mn}\boldsymbol{F}_{kl}} \left(\sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} y_{ij}^2 \right) \tag{28}$$

$$= \sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} \frac{\partial y_{ij}^2}{\partial \boldsymbol{F}_{mn} \partial \boldsymbol{F}_{kl}} \tag{29}$$

$$= 2\frac{\partial}{\partial \boldsymbol{F}_{kl}} \left(\sum_j \boldsymbol{W}_{mj} \left(\frac{\boldsymbol{A}_{mj}}{d_m} \boldsymbol{F}_{mn} - \frac{\boldsymbol{A}_{mj}}{\sqrt{d_m d_j}} \boldsymbol{F}_{jn} \right) \right) \tag{30}$$

$$= 2(\boldsymbol{q}_m \delta_{mk} - \sum_j \boldsymbol{W}_{mj} \frac{\boldsymbol{A}_{mj}}{\sqrt{d_m d_j}} \delta_{jk}) \delta_{nl} \tag{31}$$

$$= 2(\mathrm{diag}(\boldsymbol{q}) - \boldsymbol{W} \odot \tilde{\boldsymbol{A}})_{mk} \delta_{nl}. \tag{32}$$

□

Remark 2. *From Eq.* (21) *to Eq.* (24)*, one can assume* $m \notin \mathcal{N}(m)$*, and thus* $\boldsymbol{W}_{mm} = 0$*. However, as we know, a self-loop is often added to* $\boldsymbol{A}$ *to facilitate stability by avoiding zero-degree nodes that cannot be normalized. This is not as problematic as it seems, though. Because* $\sum_{(i,j)\in\mathcal{E}, i\neq j}$ *intrinsically excludes the diagonal terms, we can simply assign zero to the diagonal terms of* $\boldsymbol{W}$ *so that the term of* $j = m$ *is still zero in Eq.* (24)*, as defined in Eq.* (5)*.*

To minimize $\hat{\mathcal{H}}$, the gradient descent update rule takes the form of Eq. (6). One may assume that when η is chosen to be small enough, $\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$. For a formal justification, we have Theorem 1 to determine the convergence condition of η.

Theorem 1. *If* $\boldsymbol{F}^{(k)}$ *follows the update rule in Eq.* (6)*, where the* ρ *satisfies that* $\frac{d\rho(y)}{dy^2}$ *is non-decreasing for* $y \in (0, \infty)$*, then a sufficient condition for* $\mathcal{H}(\boldsymbol{F}^{(k+1)}) \leq \mathcal{H}(\boldsymbol{F}^{(k)})$ *is that the step size* η *satisfies* $0 < \eta \leq \|diag(\boldsymbol{q}^{(k)}) - \boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}} + \lambda \boldsymbol{I}\|_2^{-1}$*.*

Proof. The descent of $\hat{\mathcal{H}}(\boldsymbol{F})$ can ensure the descent of $\mathcal{H}(\boldsymbol{F})$ since $\mathcal{H}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) = \mathcal{H}(\boldsymbol{F}^{(k)})$. Therefore, we only need to prove $\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$.

Noting that $\hat{\mathcal{H}}$ is a quadratic function and $\boldsymbol{F}^{(k+1)} - \boldsymbol{F}^{(k)} = -\eta \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$, then $\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)})$ and $\hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$ can be connected using Taylor expansion[6], where $\nabla \hat{\mathcal{H}}$ and $\nabla^2 \hat{\mathcal{H}}$ is given in Lemma 2:

$$\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) - \hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) \tag{33}$$

$$= \mathrm{tr}\left(\nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})^\top (\boldsymbol{F}^{(k+1)} - \boldsymbol{F}^{(k)})\right) \tag{34}$$

$$+ \frac{1}{2}\mathrm{tr}\left((\boldsymbol{F}^{(k+1)} - \boldsymbol{F}^{(k)})^\top \nabla^2 \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})(\boldsymbol{F}^{(k+1)} - \boldsymbol{F}^{(k)})\right) \tag{35}$$

$$= \mathrm{tr}\left(-\eta \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})^\top \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})\right) \tag{36}$$

$$+ \mathrm{tr}\left(\frac{\eta^2}{2} \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})^\top \nabla^2 \hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) \nabla \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})\right). \tag{37}$$

Insert $\nabla^2 \hat{\mathcal{H}}(\boldsymbol{F}^{(k)}) = 2(\mathrm{diag}(\boldsymbol{q}) - \boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}} + \lambda \boldsymbol{I})$ from Lemma 2 into the above equation and we can find a sufficient condition for $\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$ to be

$$-\eta + \|\mathrm{diag}(\boldsymbol{q}) - \boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}} + \hat{\lambda} \boldsymbol{I}\|_2 \eta^2 \leq 0, \tag{38}$$

or

$$\eta \leq \|\mathrm{diag}(\boldsymbol{q}) - \boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}} + \hat{\lambda} \boldsymbol{I}\|_2^{-1}. \tag{39}$$

□

Now we prove that when taking the Quasi-Newton-IRLS step as in Eq. (7), the objective $\hat{\mathcal{H}}$ is guaranteed to descend. Since the features in different dimensions are irrelevant, we simplify our notations as if feature dimension was 1. One may easily recover the general scenario by taking the trace.

Lemma 3. *$2\hat{\boldsymbol{Q}} - \nabla^2 \hat{\mathcal{H}}(\boldsymbol{y})$ is positive semi-definite, where $\hat{\mathcal{H}} = \sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} y_{ij}^2 + \lambda \sum_{i\in\mathcal{V}} \|\boldsymbol{f}_i - \boldsymbol{f}_i^{(0)}\|_2^2$, $\hat{\boldsymbol{Q}} = 2(\mathrm{diag}(\boldsymbol{q}) + \lambda \boldsymbol{I})$, and $\boldsymbol{q}_m = \sum_j \boldsymbol{W}_{mj} \boldsymbol{A}_{mj}/d_m$.*

Proof. In Lemma 2, we have $\nabla^2 \hat{\mathcal{H}}(\boldsymbol{y}) = 2(\mathrm{diag}(\boldsymbol{q}) + \lambda \boldsymbol{I} - \boldsymbol{W} \odot \tilde{\boldsymbol{A}})$, then

$$2\hat{\boldsymbol{Q}} - \nabla^2 \hat{\mathcal{H}}(\boldsymbol{y}) = 2(\mathrm{diag}(\boldsymbol{q}) + \lambda \boldsymbol{I} + \boldsymbol{W} \odot \tilde{\boldsymbol{A}}). \tag{40}$$

Recall how we derived Eq. (27) from Eq. (15), where we proved that

$$\sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} \|\frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}}\|_2^2 = \mathrm{tr}(\boldsymbol{F}^\top (\mathrm{diag}(\boldsymbol{q}) - \boldsymbol{W} \odot \tilde{\boldsymbol{A}}) \boldsymbol{F}), \tag{41}$$

which holds for all $\boldsymbol{F}$. Similarly, the equation still holds after flipping the sign before $\boldsymbol{f}_j/\sqrt{d_j}$ and $\boldsymbol{W} \odot \tilde{\boldsymbol{A}}$. We then have this inequality: $\forall \boldsymbol{F}, \forall \lambda \geq 0$

$$0 \leq \sum_{(i,j)\in\mathcal{E}, i\neq j} \boldsymbol{W}_{ij} \|\frac{\boldsymbol{f}_i}{\sqrt{d_i}} + \frac{\boldsymbol{f}_j}{\sqrt{d_j}}\|_2^2 = \mathrm{tr}(\boldsymbol{F}^\top (\mathrm{diag}(\boldsymbol{q}) + \boldsymbol{W} \odot \tilde{\boldsymbol{A}}) \boldsymbol{F}) \tag{42}$$

$$\leq \mathrm{tr}(\boldsymbol{F}^\top (\mathrm{diag}(\boldsymbol{q}) + \boldsymbol{W} \odot \tilde{\boldsymbol{A}} + \lambda \boldsymbol{I}) \boldsymbol{F}). \tag{43}$$

Thus, $(\mathrm{diag}(\boldsymbol{q}) + \boldsymbol{W} \odot \tilde{\boldsymbol{A}} + \lambda \boldsymbol{I}) \succeq 0$, and thus $2\hat{\boldsymbol{Q}} - \nabla^2 \hat{\mathcal{H}}(\boldsymbol{y}) \succeq 0$.

□

Using Lemma 1 and Lemma 3 we can prove Theorem 2. Note that we continue to assume #feature$= 1$ for simplicity but without loss of generality[6].

Theorem 2. *If $\boldsymbol{F}^{(k+1)}$ follows update rule in Eq. (7), where ρ satisfies that $\frac{d\rho(y)}{dy^2}$ is non-decreasing $\forall y \in (0, \infty)$, it is guaranteed that $\mathcal{H}(\boldsymbol{F}^{(k+1)}) \leq \mathcal{H}(\boldsymbol{F}^{(k)})$.*

Proof. Following the discussions in Theorem 1, we only need to prove $\hat{\mathcal{H}}(\boldsymbol{F}^{(k+1)}) \leq \hat{\mathcal{H}}(\boldsymbol{F}^{(k)})$.For the quadratic $\hat{\mathcal{H}}$, we have:

$$\hat{\mathcal{H}}(\boldsymbol{x}) = \hat{\mathcal{H}}(\boldsymbol{y}) + \nabla\hat{\mathcal{H}}(\boldsymbol{y})^\top(\boldsymbol{x}-\boldsymbol{y}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top\nabla^2\hat{\mathcal{H}}(\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y}). \tag{44}$$

We can define $\mathcal{Q}(\boldsymbol{y}) = 2\hat{\boldsymbol{Q}}(\boldsymbol{y})$ in Lemma 3 such that $\mathcal{Q}(\boldsymbol{y}) - \nabla^2\hat{\mathcal{H}}(\boldsymbol{y}) \succeq 0$, then

$$\forall \boldsymbol{z}, \boldsymbol{z}^\top\mathcal{Q}(\boldsymbol{y})\boldsymbol{z} \geq \boldsymbol{z}^\top\nabla^2\hat{\mathcal{H}}(\boldsymbol{y})\boldsymbol{z}. \tag{45}$$

Then an upper bound of $\hat{\mathcal{H}}(\boldsymbol{x})$ can be found by inserting Eq. (45) into Eq. (44).

$$\hat{\mathcal{H}}(\boldsymbol{x}) \leq \hat{\mathcal{H}}(\boldsymbol{y}) + \nabla\hat{\mathcal{H}}(\boldsymbol{y})^\top(\boldsymbol{x}-\boldsymbol{y}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top\mathcal{Q}(\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y}). \tag{46}$$

Then, insert $\mathcal{Q} = 2\hat{\boldsymbol{Q}}$ into Eq. (46). Note that $\hat{\boldsymbol{Q}} \coloneqq 2(\mathrm{diag}(\boldsymbol{q}) + \hat{\lambda}\boldsymbol{I})$, so $\hat{\boldsymbol{Q}} \succeq 0$ and $\hat{\boldsymbol{Q}}^\top = \hat{\boldsymbol{Q}}$. Thereafter, the update rule $\boldsymbol{x} = \boldsymbol{y} - \hat{\boldsymbol{Q}}^{-1}\nabla\hat{\mathcal{H}}(\boldsymbol{y})$ in Eq. (7) gives

$$\begin{aligned}
&\hat{\mathcal{H}}(\boldsymbol{x}) - \hat{\mathcal{H}}(\boldsymbol{y}) && (47)\\
\leq&\nabla\hat{\mathcal{H}}(\boldsymbol{y})^\top(\boldsymbol{x}-\boldsymbol{y}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top\mathcal{Q}(\boldsymbol{y})(\boldsymbol{x}-\boldsymbol{y}) && (48)\\
=&\nabla\hat{\mathcal{H}}(\boldsymbol{y})^\top(\boldsymbol{x}-\boldsymbol{y}) + 2\left(\hat{\boldsymbol{Q}}^{\frac{1}{2}}(\boldsymbol{x}-\boldsymbol{y})\right)^\top\left(\hat{\boldsymbol{Q}}^{\frac{1}{2}}(\boldsymbol{x}-\boldsymbol{y})\right) && (49)\\
=&2\nabla\hat{\mathcal{H}}(\boldsymbol{y})^\top\hat{\boldsymbol{Q}}^{-1}\nabla\hat{\mathcal{H}}(\boldsymbol{y}) - 2\nabla\hat{\mathcal{H}}(\boldsymbol{y})^\top(\hat{\boldsymbol{Q}}^{-\frac{1}{2}})^\top\hat{\boldsymbol{Q}}^{-\frac{1}{2}}\nabla\hat{\mathcal{H}}(\boldsymbol{y}) && (50)\\
=&0. && (51)
\end{aligned}$$

Therefore, our QN-IRLS in Eq. (7) is guaranteed to descend. □

C Computation Efficiency

Our RUNG model preserves advantageous efficiency even adopting the quasi-Newton IRLS algorithm.

C.1 Time Complexity Analysis

Each RUNG layer involves computing $\boldsymbol{W}$, $\boldsymbol{q}$, and the subsequent aggregations. We elaborate on them one by one. We denote the number of feature dimensions d, the number of nodes n, and the number of edges m, which are assumed to satisfy $m \gg 1$, $n \gg 1$ and $d \gg 1$. The number of layers is denoted as k. The asymptotic computation complexity is denoted as $\mathcal{O}(\cdot)$.

Computation of $\boldsymbol{W} \odot \boldsymbol{A}$ and $\boldsymbol{W} \odot \tilde{\boldsymbol{A}}$. $\boldsymbol{W} \coloneqq \mathbf{1}_{i\neq j}\frac{d\rho_\gamma(y_{ij})}{dy_{ij}^2}$ is the edge weighting matrix dependent on the node feature matrix $\boldsymbol{F}$. The computation of $y_{ij} = \|\frac{\boldsymbol{f}_i}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j}{\sqrt{d_j}}\|_2$ is $\mathcal{O}(d)$ and that of $\frac{d\rho_\gamma(y_{ij})}{dy_{ij}^2}$ is $\mathcal{O}(1)$. $\boldsymbol{W}_{ij}$ only needs computing when $(i,j) \in \mathcal{E}$, because $\forall(i,j) \notin \mathcal{E}$, $\boldsymbol{W}_{ij}$ will be masked out by $\boldsymbol{A}$ or $\tilde{\boldsymbol{A}}$ anyways. Each element of $\boldsymbol{W}$ involves computation time of $\mathcal{O}(d)$ and m elements are needed. In total, $\boldsymbol{W}$ costs $\mathcal{O}(md)$, and $\boldsymbol{W} \odot \boldsymbol{A}$ and $\boldsymbol{W} \odot \tilde{\boldsymbol{A}}$ cost $\mathcal{O}(md + m) = \mathcal{O}(md)$.

Computation of $\hat{\boldsymbol{Q}}^{-1}$. $\hat{\boldsymbol{Q}}^{-1} \coloneqq \frac{1}{2}(\mathrm{diag}(\boldsymbol{q}^{(k)}) + \lambda\boldsymbol{I})^{-1}$ is the inverse Hessian in our quasi-Newton IRLS. Because $\hat{\boldsymbol{Q}}$ is designed to be a diagonal matrix, its inverse can be evaluated as element-wise reciprocal which is efficient. As for $\boldsymbol{q} \coloneqq \sum_j \boldsymbol{W}_{mj}^{(k)}\boldsymbol{A}_{mj}/d_m$, only existing edges $(i,j) \in \mathcal{E}$ need evaluation in the summation. Therefore, this computation costs $\mathcal{O}(m)$. Thus, $\hat{\boldsymbol{Q}}^{-1}$ costs $\mathcal{O}(m)$ in total.

Computation of aggregation. An RUNG layer follows

$$\boldsymbol{F}^{(k+1)} = 2\hat{\boldsymbol{Q}}^{-1}\left((\boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}})\boldsymbol{F}^{(k)} + \lambda\boldsymbol{F}^{(0)}\right), \tag{52}$$

which combines the quantities calculated above. An extra graph aggregation realized by the matrix multiplication between $\boldsymbol{W} \odot \tilde{\boldsymbol{A}}$ and $\boldsymbol{F}$ is required, costing $\mathcal{O}(md)$. The subsequent addition to $\boldsymbol{F}^{(0)}$ and the multiplication to the diagonal $\hat{\boldsymbol{Q}}^{-1}$ both cost $\mathcal{O}(nd)$.

Stacking layers. RUNG unrolls the QN-IRLS optimization procedure, which has multiple iterations. Therefore, the convergence increase that QN-IRLS introduces allows a RUNG with fewer layers and increases the overall complexity. It is worth noting that the QN-IRLS utilizes a diagonal approximated Hessian, and thus the computation per iteration is also kept efficient as discussed above.

Summing up all the costs, we have the total computational complexity of our RUNG, $\mathcal{O}((m+n)kd)$. Our RUNG thus scales well to larger graph datasets such as ogbn-arxiv.

Space Complexity Analysis The only notable extra storage cost is $\boldsymbol{W}$ whose sparse layout takes up $\mathcal{O}(m)$. This is the same order of size as the adjacency matrix itself, thus not impacting the total asymptotic complexity.

C.2 Alternative Perspective

In fact, the above analysis can be simplified when we look at the local aggregation behavior of RUNG. For node i, it's updated via aggregation $\boldsymbol{f}_i = \frac{2}{\bar{\boldsymbol{Q}}_{ii}^{-1}}((\sum_{j\in\mathcal{N}(i)} \boldsymbol{W}_{ij}\boldsymbol{f}_j) + \lambda \boldsymbol{f}_i^{(0)})$. The summation over neighbors' $\boldsymbol{f}_j$ will give $\mathcal{O}(m)$ in the total time complexity in each feature dimension, and $\boldsymbol{W}_{ij}$ involves $\mathcal{O}(d)$ computations for each neighbor. This sums up to $\mathcal{O}(md)$ as well. Essentially, the high efficiency of RUNG originates from that every edge weighting in our model involves only the 2 nodes on this edge.

D Comprehensive Bias Analysis

In this section, we provide multiple evidence from various perspectives to reveal the estimation bias of ℓ_1-based estimation as follows.

(1) From the theoretical perspective, the extensive literature on high-dimensional statistics [40, 41] has proved that ℓ_1 regularization induces an estimation bias.

(2) From the algorithm perspective, in Section 2.3, we provide the explanation on the ℓ_1-based estimation problem solver. Specifically, the soft-thresholding operator $S_\lambda(\theta) := \text{sign}(\theta)\max(|\theta| - \lambda, 0)$ induced by the ℓ_1 regularized problem causes a constant shrinkage for θ larger than λ, enforcing the estimator to be biased towards zero with the magnitude λ.

(3) From the numerical simulation in Section 2.3, we provide an example of mean estimation to verify this estimation bias. As shown in Figure 2, the ℓ_1 estimator (green) deviates further from the true mean as the ratio of outliers escalates. This can be clearly explained as the effect of the accumulation of estimation bias. In other words, each outlier results in a constant bias, and the bias accumulates with more outliers.

(4) From the performance perspective, ℓ_1-based GNNs such as SoftMedian, TWIRLS, and RUNG-ℓ_1 (the ℓ_1 variant of our model) suffer from significant performance degradation when the attack budget increases.

(5) From our ablation study in Figure 6, we quantify the estimation bias of the aggregated feature $f_i^\star$ on the attacked graph from the feature f_i on the clean graph: $\sum_{i\in\mathcal{V}} \|f_i - f_i^*\|_2^2$. The results demonstrate that ℓ_1-based GNN produces biased estimation under adversarial attacks and the bias indeed scales up with the attack budget. However, our proposed RUNG method exhibits a nearly zero estimation bias under the same attacking budgets.

All of this evidence can convincingly support our claim that ℓ_1-based robust estimator suffers from the estimation bias, which validates the motivation of our new algorithm design.

E Additional experiment results

In this section, we present the experiment results that are not shown in the main paper due to space limits.

E.1 Adaptive Attacks

Table 3 and Table 4 are the results of adaptive local and global attacks on Citeseer, referred to in Section 4.2.

Table 3: Adaptive local attack on Citeseer. The **best** and second are marked.

Model	0%	20%	50%	100%	150%	200%
MLP	69.3 ± 2.5	69.3 ± 2.5	69.3 ± 2.5	$\mathbf{69.3 \pm 2.5}$	$\mathbf{69.3 \pm 2.5}$	$\mathbf{69.3 \pm 2.5}$
GCN	79.3 ± 3.3	44.7 ± 8.8	27.3 ± 7.7	6.7 ± 3.7	0.7 ± 1.3	0.0 ± 0.0
APPNP	$\mathbf{80.7 \pm 4.4}$	50.0 ± 6.7	39.3 ± 6.5	16.7 ± 12.3	16.0 ± 8.0	0.0 ± 0.0
GAT	74.7 ± 5.0	15.3 ± 17.5	13.3 ± 13.5	12.0 ± 11.3	12.7 ± 6.5	9.3 ± 5.3
GNNGuard	74.7 ± 4.5	46.0 ± 10.4	32.7 ± 11.0	18.0 ± 7.5	6.0 ± 3.9	4.0 ± 3.9
RGCN	80.0 ± 2.1	46.7 ± 9.4	32.7 ± 8.8	10.0 ± 5.2	0.7 ± 1.3	0.7 ± 1.3
GRAND	77.3 ± 2.5	56.7 ± 4.2	44.0 ± 3.9	16.7 ± 6.3	0.7 ± 1.3	0.0 ± 0.0
ProGNN	80.0 ± 2.1	42.7 ± 7.4	26.0 ± 5.3	10.0 ± 4.7	0.7 ± 1.3	0.0 ± 0.0
Jaccard-GCN	78.7 ± 3.4	46.7 ± 7.3	28.0 ± 7.5	6.7 ± 4.7	0.7 ± 1.3	0.0 ± 0.0
SoftMedian	78.7 ± 3.4	69.3 ± 6.5	66.0 ± 7.1	56.0 ± 4.4	8.7 ± 6.9	3.3 ± 3.0
TWIRLS	77.3 ± 2.5	69.3 ± 1.3	68.7 ± 1.6	57.3 ± 2.5	36.7 ± 4.7	26.7 ± 4.7
TWIRLS-T	76.0 ± 3.3	70.7 ± 2.5	68.7 ± 2.7	62.0 ± 3.4	52.7 ± 5.3	47.3 ± 8.3
RUNG-ℓ_1 (ours)	80.0 ± 3.7	$\mathbf{75.3 \pm 4.5}$	$\mathbf{73.3 \pm 3.0}$	67.3 ± 3.3	36.0 ± 9.3	26.0 ± 8.3
RUNG (ours)	77.3 ± 1.3	70.7 ± 5.7	69.3 ± 6.8	67.3 ± 7.1	64.0 ± 5.7	61.3 ± 5.8

Table 4: Adaptive global attack on Citeseer. The **best** and second are marked.

Model	Clean	5%	10%	20%	30%	40%
MLP	67.7 ± 0.3	67.7 ± 0.3	67.7 ± 0.3	$\mathbf{67.7 \pm 0.3}$	$\mathbf{67.7 \pm 0.3}$	$\mathbf{67.7 \pm 0.3}$
GCN	74.8 ± 1.2	66.1 ± 1.0	60.9 ± 0.8	53.0 ± 1.0	47.0 ± 0.8	41.2 ± 1.1
APPNP	$\mathbf{75.3 \pm 1.1}$	65.8 ± 0.9	60.7 ± 1.6	52.3 ± 1.6	46.0 ± 2.0	41.2 ± 2.2
GAT	73.4 ± 1.2	65.4 ± 1.3	60.4 ± 1.4	52.6 ± 2.5	47.2 ± 3.4	41.2 ± 4.8
GNNGuard	72.4 ± 1.1	65.6 ± 0.9	61.8 ± 1.4	55.6 ± 1.4	51.0 ± 1.3	47.3 ± 1.3
RGCN	74.4 ± 1.0	66.0 ± 0.8	60.6 ± 0.9	52.5 ± 0.8	46.1 ± 0.9	40.2 ± 1.0
GRAND	74.8 ± 0.6	66.6 ± 0.7	61.8 ± 0.7	53.6 ± 1.1	47.4 ± 1.2	42.2 ± 0.9
ProGNN	74.2 ± 1.3	65.6 ± 1.1	60.3 ± 1.1	52.7 ± 1.4	46.2 ± 0.9	40.8 ± 0.6
Jaccard-GCN	74.8 ± 1.2	66.3 ± 1.2	60.9 ± 1.2	53.3 ± 0.9	46.5 ± 0.9	41.1 ± 1.0
EvenNet	74.6 ± 0.5	66.8 ± 0.5	62.0 ± 0.6	55.9 ± 0.4	51.1 ± 0.4	47.4 ± 0.8
GARNET	74.8 ± 1.3	68.0 ± 0.9	64.0 ± 1.1	58.2 ± 0.7	53.9 ± 0.8	51.0 ± 0.9
SoftMedian	74.6 ± 0.7	68.0 ± 0.7	64.4 ± 0.9	59.3 ± 1.1	55.2 ± 2.0	51.9 ± 2.1
TWIRLS	74.2 ± 0.8	69.2 ± 0.8	66.4 ± 0.7	61.6 ± 0.9	58.1 ± 1.2	51.8 ± 1.5
TWIRLS-T	73.7 ± 1.1	69.1 ± 1.2	66.4 ± 1.0	62.8 ± 1.5	60.0 ± 1.4	57.4 ± 1.5
RUNG-ℓ_1 (ours)	$\mathbf{75.5 \pm 1.1}$	69.3 ± 1.2	65.9 ± 1.2	61.1 ± 1.1	57.2 ± 1.4	53.9 ± 1.3
RUNG (ours)	74.3 ± 0.7	$\mathbf{71.4 \pm 1.0}$	$\mathbf{69.8 \pm 1.3}$	67.6 ± 1.2	66.5 ± 1.3	65.3 ± 1.5

E.2 Transfer Attacks

Table 5 and Table 6 are the results of transfer global attacks on Cora ML and Citeseer. Figure 9 and Figure 10 are the experiment results of our RUNG attacked by transfer attacks generated on different surrogate models as mentioned in Section 4.3.

Figure 11 shows results of global evasion transfer attacks between different models on Cora ML. Our observations are summarized below:

- The attacks generated by RUNG are stronger when applied to more robust models like SoftMedian, while are not strong against undefended or weakly defended models.

Table 5: Transfer global evasion attack on Cora ML.

Model	Clean	5%	10%	20%	30%	40%
MLP	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0	65.0 ± 1.0
GCN	85.0 ± 0.4	76.0 ± 0.7	70.6 ± 0.9	62.1 ± 1.0	55.4 ± 1.0	49.8 ± 1.1
APPNP	$\mathbf{86.3 \pm 0.4}$	78.0 ± 1.0	72.7 ± 1.2	64.4 ± 1.8	57.7 ± 1.5	53.0 ± 1.2
GAT	83.5 ± 0.5	78.4 ± 0.7	75.1 ± 0.9	69.5 ± 1.7	65.0 ± 1.8	61.4 ± 2.4
GNNGuard	83.1 ± 0.7	79.9 ± 0.7	78.1 ± 0.7	74.8 ± 0.9	71.8 ± 1.1	69.9 ± 1.1
RGCN	85.7 ± 0.4	77.7 ± 0.6	72.7 ± 0.8	64.4 ± 0.8	57.9 ± 0.8	52.6 ± 1.0
GRAND	86.1 ± 0.7	80.2 ± 0.7	76.4 ± 1.0	70.0 ± 1.3	64.6 ± 1.4	60.1 ± 1.1
ProGNN	85.6 ± 0.5	77.8 ± 0.7	72.8 ± 0.9	64.8 ± 1.1	58.7 ± 1.3	53.2 ± 1.3
Jaccard-GCN	83.7 ± 0.7	77.8 ± 0.7	74.1 ± 1.1	68.4 ± 1.1	63.5 ± 1.6	59.3 ± 1.4
ElasticGNN	85.9 ± 0.5	81.9 ± 0.9	79.0 ± 1.0	73.3 ± 1.3	68.1 ± 1.4	63.6 ± 1.5
SoftMedian	85.0 ± 0.7	81.8 ± 0.4	79.0 ± 0.6	73.0 ± 1.1	66.3 ± 1.6	60.4 ± 2.4
TWIRLS	84.2 ± 0.6	81.5 ± 0.7	79.7 ± 0.6	76.8 ± 0.5	74.3 ± 0.7	72.8 ± 0.9
RUNG-ℓ_1 (ours)	85.8 ± 0.5	82.3 ± 0.8	79.7 ± 0.6	75.1 ± 0.8	70.5 ± 0.8	67.1 ± 0.9
RUNG (ours)	84.6 ± 0.5	$\mathbf{83.7 \pm 0.6}$	$\mathbf{82.9 \pm 0.9}$	$\mathbf{81.3 \pm 1.3}$	$\mathbf{79.2 \pm 1.6}$	$\mathbf{77.9 \pm 2.0}$

Table 6: Transfer global evasion attack on Citeseer.

Model	Clean	5%	10%	20%	30%	40%
MLP	67.7 ± 0.3	67.7 ± 0.3	67.7 ± 0.3	67.7 ± 0.3	67.7 ± 0.3	67.7 ± 0.3
GCN	74.8 ± 1.2	66.7 ± 1.3	62.1 ± 1.3	54.5 ± 1.7	48.7 ± 1.8	43.4 ± 2.1
APPNP	75.3 ± 1.1	67.7 ± 1.2	62.9 ± 1.0	55.6 ± 1.0	50.3 ± 1.0	45.8 ± 1.6
GAT	73.4 ± 1.2	68.2 ± 1.3	64.6 ± 1.4	58.2 ± 2.1	53.2 ± 3.1	48.7 ± 3.7
GNNGuard	72.4 ± 1.1	70.6 ± 1.2	69.1 ± 1.3	67.1 ± 1.4	65.4 ± 1.7	64.2 ± 1.9
RGCN	74.4 ± 1.0	67.8 ± 1.0	63.5 ± 1.3	56.3 ± 1.5	51.0 ± 1.4	46.1 ± 1.7
GRAND	74.8 ± 0.6	68.9 ± 0.8	65.0 ± 0.9	59.2 ± 1.3	54.8 ± 1.5	51.2 ± 1.9
ProGNN	74.2 ± 1.3	66.8 ± 1.0	62.3 ± 1.0	55.0 ± 1.1	49.2 ± 1.0	43.7 ± 1.3
Jaccard-GCN	74.8 ± 1.2	68.5 ± 1.1	65.1 ± 1.0	59.0 ± 1.7	54.6 ± 1.9	51.1 ± 1.7
ElasticGNN	75.2 ± 1.2	70.9 ± 1.4	67.7 ± 1.6	62.0 ± 2.5	57.9 ± 3.2	53.4 ± 4.0
SoftMedian	74.6 ± 0.7	68.0 ± 0.7	64.4 ± 0.9	59.3 ± 1.1	55.2 ± 2.0	51.9 ± 2.2
TWIRLS	74.2 ± 0.8	71.8 ± 1.0	70.1 ± 0.9	68.4 ± 1.4	67.4 ± 1.6	66.4 ± 1.8
RUNG-ℓ_1 (ours)	$\mathbf{75.5 \pm 1.1}$	72.0 ± 1.3	69.3 ± 1.4	65.1 ± 1.8	61.8 ± 2.0	58.7 ± 2.4
RUNG (ours)	74.3 ± 0.7	$\mathbf{74.2 \pm 1.0}$	$\mathbf{74.2 \pm 1.0}$	$\mathbf{74.3 \pm 1.0}$	$\mathbf{74.2 \pm 1.1}$	$\mathbf{74.1 \pm 1.1}$

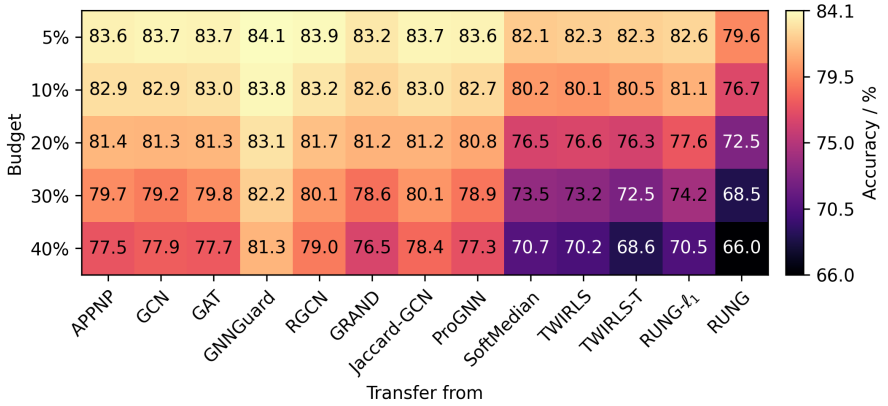

Figure 9: Transfer global attack from different surrogate models to our RUNG on Cora ML.

- For ℓ_1 GNNs, the attacks are the strongest when transferred from ℓ_1 GNNs. This supports again our unified view on ℓ_1 GNNs. An exception is TWIRLS because it only has one attention layer and does not always converge to the actual ℓ_1 objective.

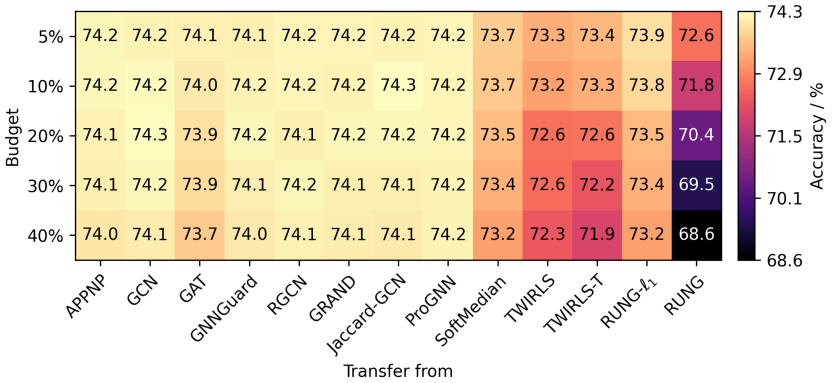

Figure 10: Transfer global attack from different surrogate models to our RUNG on Citeseer.

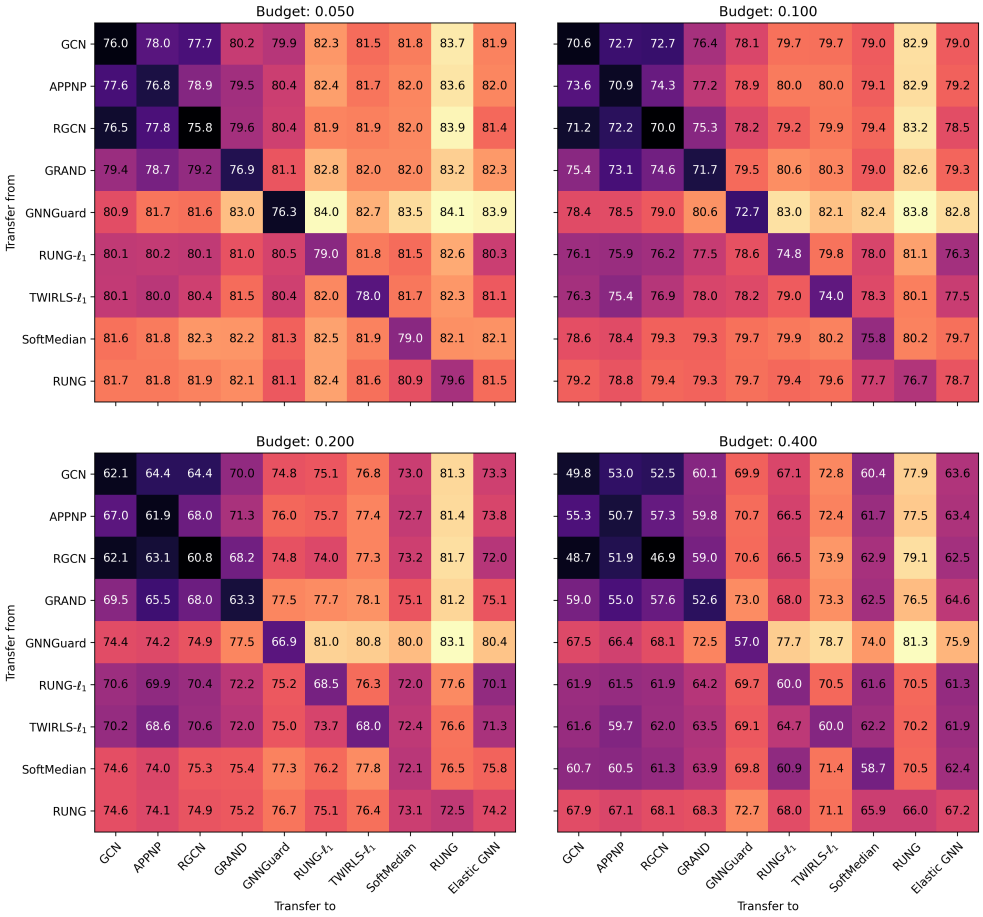

Figure 11: Transfer global attack between different model pairs on Cora.

E.3 Poisoning Attacks

We provide the experiment results under poisoning attacks on Cora ML and Citeseer in Table 7 and Table 8, respectively.

Table 7: Poisoning Attacks on Cora ML.

Budget	5%	10%	20%	30%	40%
GCN	74.9 ± 0.4	69.7 ± 0.7	60.7 ± 0.7	54.0 ± 1.0	48.7 ± 1.0
APPNP	76.3 ± 0.9	71.1 ± 1.2	63.0 ± 1.3	57.1 ± 0.6	53.2 ± 1.1
SoftMedian	79.2 ± 0.7	75.6 ± 0.3	67.8 ± 0.6	62.9 ± 1.0	58.6 ± 0.7
RUNG-ℓ_1 (ours)	79.7 ± 0.6	76.4 ± 0.6	68.1 ± 0.6	63.8 ± 0.5	60.1 ± 0.9
RUNG (ours)	78.5 ± 0.5	75.5 ± 0.3	71.5 ± 0.4	67.1 ± 1.6	64.6 ± 1.3

Table 8: Poisoning Attacks on Citeseer.

Budget	5%	10%	20%	30%	40%
GCN	65.5 ± 1.1	59.8 ± 1.0	51.0 ± 1.0	44.0 ± 1.2	37.9 ± 1.0
APPNP	64.2 ± 1.8	58.1 ± 2.6	49.8 ± 2.5	43.4 ± 2.3	40.6 ± 2.7
SoftMedian	67.1 ± 1.0	63.8 ± 1.0	58.5 ± 1.1	54.3 ± 1.9	51.2 ± 2.4
RUNG-ℓ_1 (ours)	68.9 ± 1.0	65.9 ± 1.1	61.0 ± 1.0	57.2 ± 1.1	53.9 ± 1.3
RUNG (ours)	72.4 ± 0.9	72.1 ± 1.2	71.3 ± 1.4	70.8 ± 1.3	69.7 ± 1.4

E.4 Large Scale Ogbn-Arxiv

In the large scale datasets, we can not directly apply the vanilla PGD attack [36] on them due to excessive requirement of memory and computation on dense matrix. Alternatively, we leverage the PRBCD [42] to scale the PGD attack instead of manipulating on the dense adjacency matrix. We conduct experiments on large scale Ogbn-Arxiv and the results in Table 9 verified the superior robustness of our RUNG model. RUNG's robustness outperforms its ℓ_1 variant which delivers similar performance as SoftMedian and Elastic GNN due to their shared ℓ_1 graph smoothing.

Table 9: Global PGD Attacks on Ogbn-Arxiv.

Model	Clean	1%	5%	10%
GCN	71.9 ± 0.5	63.1 ± 0.4	48.9 ± 3.2	41.8 ± 0.5
APPNP	71.7 ± 0.3	64.2 ± 0.4	50.1 ± 2.3	42.2 ± 1.3
SoftMedian	71.2 ± 0.5	65.1 ± 0.3	54.1 ± 1.6	50.1 ± 2.6
RUNG-ℓ_1 (ours)	71.6 ± 0.6	65.5 ± 0.4	55.0 ± 1.1	49.6 ± 1.1
RUNG (ours)	70.2 ± 2.1	65.2 ± 0.2	64.0 ± 0.8	61.2 ± 0.7

E.5 Adversarial Training

We conduct the adversarial training following [36] and present the results in Table 10. From the results, we can observe that with adversarial training, the robustness of RUNG can be further improved.

Table 10: Adversarial Training vs Normal Training on RUNG.

Budget	Clean	5%	10%	20%	30%	40%
Normal Training	84.6 ± 0.5	78.9 ± 0.4	75.7 ± 0.2	71.8 ± 0.4	67.8 ± 1.3	65.1 ± 1.2
Adversarial Training	84.3 ± 1.2	81.8 ± 0.7	79.9 ± 0.3	77.3 ± 1.1	72.8 ± 0.6	69.1 ± 0.9

E.6 Graph Injection Attack

The injection attack was conducted following the settings in [37] to evaluate the robustness of different methods. We set up the budget on the number of injected nodes as 100 and the budget on degree as 200. The results in Table 11 show that our RUNG significantly outperforms the baseline models.

Table 11: Graph Injection Attack on Citeseer.

Model	Clean	Attacked
GCN	75.40	28.14
APPNP	75.84	25.24
GNNGuard	73.47	34.73
SoftMedian	74.82	38.91
RUNG-ℓ_1	75.01	39.22
RUNG-MCP (ours)	75.65	51.13

F Adaptive Attack Settings

For the adaptive PGD attack we adpoted in the experiments, we majorly followed the algorithm in [5] in the adaptive evasion attack. For the sake of completeness, we describe it below:

In summary, we consider the topology attack setting where the adjacency matrix $\boldsymbol{A}$ is perturbed by $\delta\boldsymbol{A}$ whose element $\delta\boldsymbol{A}_{ij} \in \{0, 1\}$. The budget B is defined as $B \geq \|\delta\boldsymbol{A}\|_0$. The PGD attack involves first relaxing $\boldsymbol{A}$ from binary to continuous so that a gradient ascent attack can be conducted on the relaxed graph.

During the attack, the minimization problem below is solved:

$$\delta\boldsymbol{A}_\star = \arg\min_{\delta\boldsymbol{A}} \mathcal{L}_{\text{attack}}(\text{GNN}_\theta(\boldsymbol{A} + (\boldsymbol{I} - 2\boldsymbol{A}) \odot \delta\boldsymbol{A}, \boldsymbol{F}), y_{\text{target}}), \tag{53}$$

where $\mathcal{L}$ is carefully designed attack loss function [4, 5], $\boldsymbol{A}$, $\boldsymbol{F}$ and y_{target} are respectively the graph, node feature matrix and ground truth labels in the dataset, θ is the parameters of the GNN under attack which are not altered in the evasion attack setting. $(\boldsymbol{I} - 2\boldsymbol{A}) \odot \delta\boldsymbol{A}$ is the calculated perturbation that "flips" the adjacency matrix between 0 and 1 when it is perturbed. The gradient of $\frac{\mathcal{L}_{\text{attack}}}{\delta\boldsymbol{A}}$ is computed and utilized to update the perturbation matrix $\delta\boldsymbol{A}$.

After the optimization problem is solved, $\delta\boldsymbol{A}$ is projected back to the feasible domain of $\delta\boldsymbol{A}_{ij} \in \{1\}$. The adjacency matrix serves as a probability matrix allowing a Bernoulli sampling of the binary adjacency matrix $\boldsymbol{A}'$. The sampling is executed repeatedly so that an $\boldsymbol{A}'$ producing the strongest perturbation is finally generated.

G Additional Ablation Study of RUNG

G.1 Hyperparameters

The choice of the hyperparameters γ and λ is crucial to the performance of RUNG. We therefore experimented with their different combinations and conducted adaptive attacks on Cora as shown in Fig. 12.

Recall the formulation of RUNG in Eq.(8):

$$\boldsymbol{F}^{(k+1)} = (\text{diag}(\boldsymbol{q}^{(k)}) + \lambda\boldsymbol{I})^{-1}\left((\boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}})\boldsymbol{F}^{(k)} + \lambda\boldsymbol{F}^{(0)}\right), \tag{54}$$

where $\boldsymbol{q}_m^{(k)} = \sum_j \boldsymbol{W}_{mj}^{(k)}\boldsymbol{A}_{mj}/d_m$, $\boldsymbol{W}_{ij}^{(k)} = \mathbf{1}_{i\neq j}\max(0, \frac{1}{2y_{ij}^{(k)}} - \frac{1}{2\gamma})$ and $y_{ij}^{(k)} = \left\|\frac{\boldsymbol{f}_i^{(k)}}{\sqrt{d_i}} - \frac{\boldsymbol{f}_j^{(k)}}{\sqrt{d_j}}\right\|_2$.

In the formulation, λ controls the intensity of the regularization in the graph smoothing. In our experiments, we tune $\hat{\lambda} := \frac{1}{1+\lambda}$ which is normalized into $(0,\ 1)$. In Figure 12, the optimal value of $\hat{\lambda}$ can be found almost always near 0.9 regardless of the attack budget. This indicates that our penalty function ρ_γ is decoupled from γ which makes the tuning easier, contrary to the commonly used formulation of MCP [25].

On the other hand, γ has a more intricate impact on the performance of RUNG. Generally speaking, the smaller γ is, the more edges get pruned, which leads to higher robustness and a lower clean accuracy. We begin our discussion in three cases:

Small attack budget ($0\%, 5\%, 10\%$). The performance is largely dependent on clean accuracy. Besides, when $\gamma \to \infty$, RUNG becomes a state-of-the-art robust ℓ_1 model. Therefore, a small γ

likely introduces more harm to the clean performance than robustness increments over ℓ_1 models. The optimal γ thus at least recovers the performance of ℓ_1 models.

Large attack budget ($20\%, 30\%, 40\%$). In these cases, $\gamma \to \infty$ is no longer a good choice because ℓ_1 models are beginning to suffer from the accumulated bias effect. The optimal γ is thus smaller (near 0.5). However, for fairness, we chose the same γ under different budgets in our experiments, so the reported RUNG fixes $\gamma = 3$. In reality, however, when we know the possible attack budgets in advance, we can tune γ for an even better performance.

Very large attack budget ($50\%, 60\%$). We did not include these scenarios because almost all GNNs perform poorly in this region. However, we believe it can provide some insights into robust graph learning. Under these budgets, more than half of the edges are perturbed. In the context of robust statistics (e.g. mean estimation), the estimator will definitely break down. However, in our problem of graph estimation, the input node features offer extra information allowing us to exploit the graph information even beyond the breakdown point. In the "peak" near $(0.9, 0.5)$, RUNG achieves $> 70\%$ accuracy which is higher than MLP. This indicates that the edge weighting of RUNG is capable of securely harnessing the graph information even in the existence of strong adversarial attacks. The "ridge" near a $\hat{\lambda} = 0.2$, on the other hand, emerges because of MLP. When the regularization dominates, $\lambda \to \infty$, and $\hat{\lambda} \to 0$. A small λ is then connected to a larger emphasis on the input node feature prior. Under large attack budgets, MLP delivers relatively good estimation, so a small $\hat{\lambda}$ is beneficial.

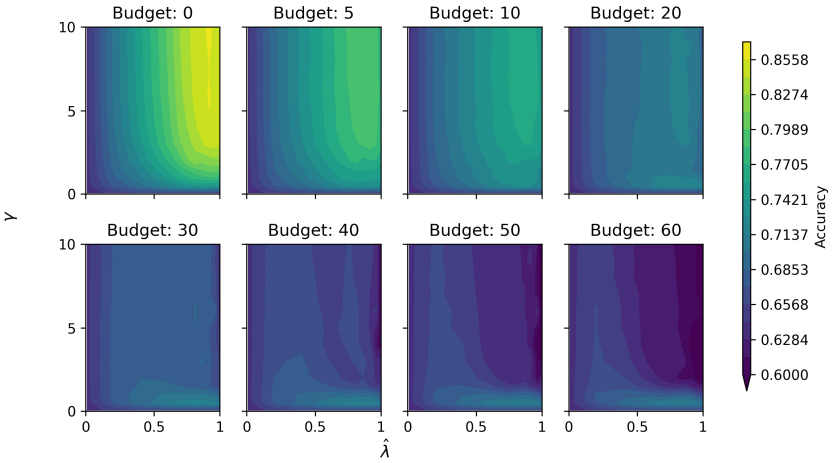

Figure 12: The performance dependence of RUNG with different hyperparameters γ and λ. The performance is evaluated under different attack budgets. The attack setting is the global evasion attack and the dataset is Cora. Note the x-axis is $\hat{\lambda} := \frac{1}{1+\lambda}$ instead of λ.

G.2 GNN Layers

In RUNG, QN-IRLS is unrolled into GNN layers. We would naturally expect RUNG to have enough number of layers so that the estimator converges as desired. We conducted an ablation study on the performance (clean and adversarial) of RUNG with different layer numbers and the results are shown in Fig. Figure 13. We make the following observations:

- As the layer number increases, RUNG exhibits better performance. This verifies the effectiveness of our proposed RUGE, as well as the stably converging QN-IRLS.
- The performance of RUNG can achieve a reasonably good level even with a small layer number (3-5 layers) with accelerated convergence powered by QN-IRLS. This can further reduce the computation complexity of RUNG.

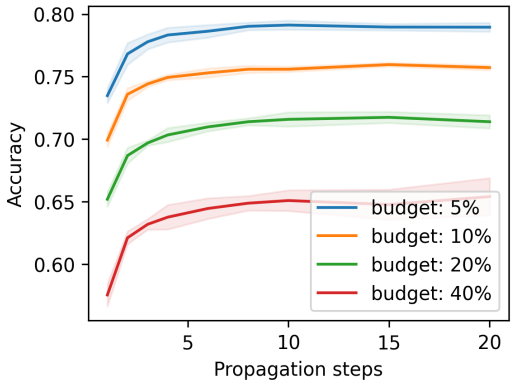

Figure 13: The performance dependence of RUNG on the number of aggregation layers.

H Robustness of GCN and APPNP

In addition to the formulation in section 3 the main text, we can simply apply our edge reweighting technique to the GCN architecture. Essentially the aggregation operation in GCN can be viewed as an APPNP layer with $\hat{\lambda} = 0$. The GCN version of a layer in our model can then be formulated as

$$\boldsymbol{F}^{(k+1)} = \mathrm{ReLU}((\boldsymbol{W}^{(k)} \odot \tilde{\boldsymbol{A}})\boldsymbol{F}^{(k)}\boldsymbol{M}^{(k)}), \tag{55}$$

where $\boldsymbol{M}$ is the learned weight matrix in GCN.

GCN-based defenses are less robust than APPNP-based defenses GCN consists of layers in which both feature transformation and message passing are included. This graph convolution operation will weaken defense methods that rely on edge weighting, such as GNNGuard, models using l_1 penalty as well as our method using MCP penalty.

Consider an edge that is added by the attacker[7]. A successful defense should detect the attack on this edge by the large difference of nodes connected by this edge, and then assign a zero weight or at least a smaller weight to this edge. However in GCN, even if this edge is detected in the first layer's message passing, the subsequent feature transformation makes the node difference less likely to be preserved until the second layer. This is where the attack can evade the defense and is thus a vulnerability allowing adaptive attacks through. According to our experiments, using different defense parameters in different layers of GCN, unfortunately, does not help much either.

On the other hand, in APPNP node features are also altered along the message-passing layers, but the node distance change is more regulated than in GCN since MLP is decoupled from the graph smoothing layers. In the latter submodule, node differences simply decrease, which allows our defense based on node differences.

Experiments It can be seen from Figure 14 that the correlation of node feature differences in different layers of GCN is about 4 times less than in APPNP, which means that an attack detected in the first layer is less likely to continue to be detected in the second layer than in APPNP. This property of GCN makes the many defense methods using GCN architecture as a backbone less robust, as shown in the experiment results in Table 12 and Table 13. Nevertheless, our GCN-based MCP model outperforms the SOTA models using l_1 methods.

[7]Almost all attacks add new edges as shown by our experiments, so this is almost always the case

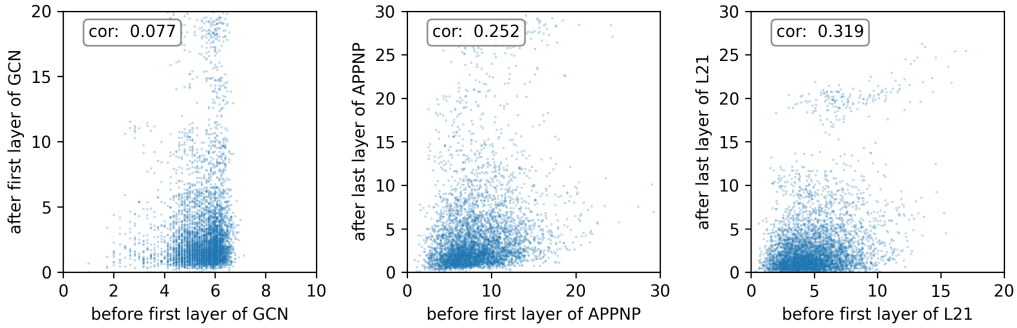

Figure 14: The distances $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2$ between connected nodes in different layers are shown. The linear transform operation in aggregation layers of GCN reduces the correlation between the node distance at different propagation layers.

Model	Clean	5%	10%	20%	30%	40%
RUNG-ℓ_1	85.8 ± 0.5	78.4 ± 0.4	74.3 ± 0.3	68.1 ± 0.6	63.5 ± 0.7	59.8 ± 0.8
RUNG	84.6 ± 0.5	78.9 ± 0.4	75.7 ± 0.2	71.8 ± 0.4	67.8 ± 1.3	65.1 ± 1.2
RUNG-ℓ_1-GCN	84.0 ± 0.4	74.7 ± 0.6	69.7 ± 0.7	62.7 ± 0.6	57.6 ± 0.6	53.5 ± 0.8
RUNG-GCN	82.6 ± 0.6	76.1 ± 0.7	71.0 ± 0.9	64.3 ± 1.1	59.9 ± 0.8	56.5 ± 1.1

Table 12: Comparison between APPNP-based and GCN-based QN-IRLS on Cora.

Model	Clean	5%	10%	20%	30%	40%
RUNG-ℓ_1	75.5 ± 1.1	69.3 ± 1.2	65.9 ± 1.2	61.1 ± 1.1	57.2 ± 1.4	53.9 ± 1.3
RUNG	74.3 ± 0.7	71.4 ± 1.0	69.8 ± 1.3	67.6 ± 1.2	66.5 ± 1.3	65.3 ± 1.5
RUNG-ℓ_1-GCN	73.8 ± 1.2	66.1 ± 1.1	61.9 ± 1.0	56.0 ± 0.7	51.3 ± 0.8	47.7 ± 0.6
RUNG-GCN	73.0 ± 0.8	68.2 ± 0.8	64.3 ± 1.3	59.4 ± 1.0	55.7 ± 1.4	53.1 ± 1.6

Table 13: Comparison between APPNP-based and GCN-based QN-IRLS on Citeseer.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction (Section 1) closely follow the contribution (at the end of Section 1) in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in the Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the theory in Section 3.2 and the detailed proof in the Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the detailed experimental setting in Section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will organize the data and code after the submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the detailed experimental setting in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We include the theoretical computation analysis in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper closely follows NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: Our paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.