Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Vibrational Inequality Based Framework for Stability and Generalization in Graph Neural Networks

Mubasher Nadeem
Scholar at Muslim Youth University, Islamabad. kahoutmubasher@gmail.com
Muhammad Sajid Saeed
Scholar at Muslim Youth University, Islamabad. saji.saeed667788@gmail.com
Rubab Aslam
Scholar at Muslim Youth University, Islamabad. rubabaslamawan@gmail.com
Ihtisham Ul Haq Malik
Scholar at Muslim Youth University, Islamabad. ihtishammalik63@yahoo.com

ABSTRACT

Graph Neural Networks (GNNs) have transformed graph learning, and Graph Convolutional Networks (GCNs) set new state of the art performance in node classification. GNNs can however be very unstable to train, particularly in the case of noisy graph topologies or scarce labelled data, which may delay convergence and generalisation. This paper introduces a new regularization theory in the framework of the theories of Variational Inequality (VI): the theory is based on introducing a regularization term in GNN training that may be formulated directly in response to the stability issues of GNNs. In contrast to simply training GCNs, our method complements the conventional training process, and adds a regularizer term based on the VI, which constrains the direction in which parameters are updated whereas it does not modify the model architecture, thus optimizing smoother optimization paths. On Cora citation network, we empirically analyze the suggested VI GCN model. The experimental evidence shows that VI GCN performs better than the default GCN baseline at being more stable and more accurate. Key metrics of the performance, such as the loss routes in training, the test accuracy, the dynamics of the norms on the weights, and histograms, show that VI regularization contributes to a smoother convergence as well as the better generalization. This epoch versus epoch power loss comparison provides additional support to such increase in stability being brought by our scheme. This paper not only emphasizes the effectiveness of adding optimization level regularization to GNN training but also opens up the research path in handling the issue of graph based learning instability.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

The suggested VI based approach is model independent, simple to deploy, and computationally efficient, which is why it can be applied to diverse graph neural network architectures and real life graph learning problems.

Introduction

The graph data is everywhere with many applications reaching real life such as citation networks, social networks, molecular chemistry, or recommendation systems. Graph structures are inherently irregular unlike the grid based structure like images or sequences and the connectivity structure can be very complex and poses a major problem to conventional neural networks. Graph Neural Networks (GNNs) have become a useful framework to learning in graph structured data that utilizes message passing processes to collective information about a node and its neighbors and to model local and global graph structure. Of all the GNN designs, the Graph Convolutional Network (GCN) has more recently proven especially influential both because of its computational advantage and because of its high empirical performance on semi supervised node classification benchmarks including Cora [1]. GNNs have experienced two characteristic and closely related problems not by standing their growing popularity: training electro instability and generalization. Locally robust The ability of the outputs of a model to be robust to small changes in a graph, input features, or model parameters. Generally speaking, the notion of generalization refers to the capacity of a model that was trained on a restricted set of labeled nodes to do well on previously unobserved data. The recursive neighborhood aggregation used in GNNs in the context of graph domains may introduce noise and instability into the network, in particular, in low label or noisy settings. GNNs may suffer abrupt loss surfaces, unstable gradient updates as well as bad convergence properties, which eventually causes overfitting and a deterioration of performance on unseen nodes. These have been addressed in recent theoretical work, by placing spectral constraints on the graph filters learned, i.e. Lipschitz or integral Lipschitz continuity, which can bound the sensitivity of GNN outputs to perturbations of the graph topology [2].

These methods provide very elegant guarantees mathematically, but the promise of them is usually that you need to fundamentally alter the model architecture or training process in order to apply them, and thus to apply to large,

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

heterogeneous graphs at scale is challenging. Typical regularization methods on GNNs like the Drop Edge Drop Node, and weight decay, tend to be more heuristic and lack a direct control over stability of optimization path and gradient alignment during training. Motivation and Research: There is a critical research gap at the current time there is no direct and optimization level mechanism of enforcing stability in GNN training. Practically all previous approaches attempt to regularize either the way that information propagates across the graph or the responses of the filters and node states, but not on those that the parameter tread b itself, which is a strong influence on not only stable convergence properties but also the ability to generalize well. This gap can only be filled with a principled solution that, in a principled way, punishes the unstable parameter updates directly but stays adaptable to standard GNN architectures. Contribution In this paper, we suggest a Regularization step using Variational Inequality (VI) theory in GNNs, with fundamental ideas in optimization and game theory. The basic concept is to add a regularizer to both penalize misaligned or oscillating updates of the parameters by forcing alignment of the gradient and the update direction of parameters at each update of the training exercise. This scheme promotes stable, and more convergent behavior, without altering the GNN base model or any major computational cost. This is a model agnostic approach that can be added to any training pipeline [3]. We confirm our methodology on the Cora citation network and we show that VI based regularization enhances stability and generalization, using loss curves, accuracy, behaviour of weight norms and dynamics of gradients. Our results found a new theoretically based paradigm of robust graph representation learning.

Related Work

Graph Neural Network Architectures

The development of Graph Neural Networks (GNNs) has been fast since their introduction and there is a number of variants of an architecture and corresponding narrow goals of addressing certain limitations. Perhaps the most common is the Graph Convolutional Network (GCN) with its layer wise propagation rule that filters normalized node feature information about immediate neighbors of a node. Although successful in most applications, GCNs are plagued by over smoothing in deeper layers that lead to clear node representations. To solve these problems

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

spectral models like Chebyshev Networks (ChebNet) [4] use graph Laplacian based filters that work with higher order feature aggregation using Chebyshev approximations. Graph Attention Networks (GATs) have incorporated attention mechanisms to adaptively weigh the influence of neighboring nodes, leading to improved expressive power and the ability to capture complex dependencies. Despite these architectural innovations, most GNNs fundamentally rely on recursive neighborhood aggregation, which makes them inherently sensitive to changes or noise in graph topology.

Stability in Graph Neural Networks

The stability of GNNs defined as their robustness to small perturbations in the graph structure, features, or parameters has become a central concern, particularly as GNNs are applied in high stakes domains [5]. In their foundational work, introduced a rigorous mathematical framework for analyzing the stability of GNNs. Their analysis demonstrated that Lipschitz and integral Lipschitz constraints on graph filters can ensure bounded spectral responses and reduce the sensitivity of GNNs to topological perturbations. Achieving such stability often requires complex custom filter designs or significant architectural changes. Enforcing Lipschitz continuity can also restrict model expressiveness and complicate training on large scale or heterogeneous graphs. Most spectral stability techniques focus on the properties of the learned filters rather than directly regulating the optimization trajectory or training dynamics where instability often originates.

Regularization in Deep Learning and GNNs

Regularization strategies are fundamental in deep learning for promoting generalization and reducing overfitting. Standard techniques include L2 weight decay, dropout, and batch normalization. In the GNN context, methods such as Drop Edge [6] and Drop Node introduce randomness in the graph structure to enhance robustness, while label smoothing reduces the impact of noisy supervision. These approaches are largely heuristic and lack a unified mathematical grounding in the context of optimization stability. While effective to some extent, they do not explicitly guide the learning process toward stable descent paths and may not prevent gradient instability, especially in sparse or imbalanced graphs.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Variational Inequality Theory in Optimization

Variational Inequality (VI) theory provides a principled framework for studying equilibrium and constrained optimization problems [7]. In machine learning, VI theory has been leveraged in adversarial training and bilevel optimization settings, but its direct application to the stabilization of neural network training especially in GNNs remains underexplored. The potential of VI formulations is that stability constraints can be encoded by approximating updates to feasible or equilibrium respecting directions, which is an attractive method to control the behaviour of gradient objectives and achieve smooth optimisation.

Gaps in Current Research

This leaves a gap that cannot afford to go without the radar: even in the case of architectural regularization and innovation, when it comes to optimization methods, one usually only optimizes either spectral or structural aspects, rather than regularizing the optimization trajectory itself [8]. There is a lack of techniques that explicitly align gradient updates with prior descent directions to ensure stable, predictable convergence. This motivates our introduction of VI based regularization as a direct and theoretically grounded means to address the instability of GNN training.

Theoretical Foundations & Preliminaries

Graph Notation and Cora Dataset Overview

Let $G = (V, E)$ denote an undirected graph, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Each node $V_i \in V$ has a feature vector $x_i \in \mathbb{R}^d$. The overall feature matrix is $X \in \mathbb{R}^{n \times d}$, where $n - |V|$ and d is the input feature dimension. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ encodes edge connections:

$$A_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

The degree matrix D is diagonal with $D_{ii} = \sum_j A_{jj}$.

The (unnormalized) graph Laplacian is:

$$L = D - A \quad (1)$$

The normalized Laplacian, useful for spectral analysis, is:

$$L_{\text{norm}} = I - D^{-1/2} A D^{-1/2} \quad (2)$$

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Cora Dataset: The Cora citation network is a standard benchmark for node classification in graph learning. It consists of 2,708 nodes (papers), 5,429 edges (citations), and 1,433 binary word features per node. The classification task is to predict the research category for each publication among seven classes. Cora's sparsity and limited labeled nodes make it especially suitable for evaluating both stability and generalization of GNNs.

Basics of Graph Signal Processing

Graph Signal Processing (GSP) extends Fourier analysis to graphs. For Laplacian $L = U\Lambda U^T$, where U is the orthonormal eigenvector matrix and Λ diagonal of eigenvalues), the graph Fourier transform of a signal x is:

$$\hat{x} = U^T x \tag{3}$$

A spectral graph convolution is defined as:

$$g\theta \quad * x = Ug\theta \quad (\Lambda)U^T x, \tag{4}$$

Where $g\theta \quad (\Lambda)$ is a learnable spectral filter? This is theoretically powerful but expensive, so spatial approximations (e.g., GCN) are often used.

Overview of GCN Layer Operations

Graph Convolutional Networks approximate spectral filtering with a localized aggregation:

$$H^{(l=1)} = \sigma\left(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}H^l W^l\right) \tag{5}$$

Where $\widetilde{A} = A + I$ includes self loops, $\widetilde{D}$ is the degree matrix of $W^{(l)}$ are learnable weights, and σ is an activation function (e.g., ReLU).

This formulation is both efficient and spatially interpretable but may propagate noise, leading to instability.

Variational Inequality (VI) Theory and Its Application

A Variational Inequality (VI) problem, for a convex set $K \subseteq \mathbb{R}^n$ and a mapping $: K \to \mathbb{R}^n$, seeks $x^* \in K$ such that:

$$\langle F(x*), x - x* \rangle \geq 0, \forall x \in K. \tag{6}$$

When F the gradient of a loss function, VI is provides a principled framework to formalize equilibrium and constrain optimization dynamics.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Mathematical Justification for VI in GNN Training

Let θ^i be model parameters at iteration $i, \theta^{\grave{i}}$ a reference point (e.g., previous epoch), and $\nabla_{\theta_j}\mathcal{L}_{\text{task}}$ the task loss gradient. We introduce the VI regularizer:

$$\mathcal{R}_{\mathcal{VJ}} = \lambda \sum_j \left\langle \nabla_{\theta_j}\mathcal{L}_{\text{task}}, \theta_j - \theta_j^{'} \right\rangle \qquad (7)$$

With regularization strength $\lambda > 0$. This penalizes updates diverging from stable descent directions, smoothing the optimization trajectory and promoting stable convergence.

Total Training Loss

$$\mathcal{L}_{task} = \mathcal{L}_{\text{task}} + \mathcal{R}_{\text{V}} \qquad (8)$$

Stability Definition

Stability, as in [4], is formally:

$$||f_\theta(X, A + \Delta A) - f_\theta(X, A)|| \leq C||\Delta A|| \qquad (9)$$

For some constant C, where ΔA is a perturbation in the adjacency and f_θ is the GCN.

Solved Example: VI Regularization on a Toy GCN

To illustrate the practical application of the proposed Variational Inequality (VI)-based regularization, consider a simple toy example using a small graph neural network. This step by step demonstration shows how GCN layer aggregation and VI regularization are applied in a real calculation.

Toy Graph Setup

Consider an undirected graph $G = (V, E)$ with three nodes and the following adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The corresponding degree matrix is:

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The (unnormalized) graph Laplacian is:

$$L = D - A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Suppose each node has a scalar feature, so the feature vector is:

$$X = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

GCN Layer Calculation

First, we add self-loops to the adjacency matrix:

$$\widetilde{A} = A + I = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

The new degree matrix is $\widetilde{D} = diag(2,3,2)$.

We compute the normalized adjacency matrix:

$$\widetilde{D}^{-1/2} = diag(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{2}})$$

The propagation rule for a GCN layer is:

$$H = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}XW$$

Assume the weight parameter $W = 1$ and no activation function for simplicity. The normalized adjacency becomes:

$$\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2} = \begin{bmatrix} 1/2 & 1/\sqrt{6} & 0 \\ 1/\sqrt{6} & 1/3 & 1/\sqrt{6} \\ 0 & 1/\sqrt{6} & 1/2 \end{bmatrix}$$

Multiplying by $\boldsymbol{X}$:

$$H = \begin{bmatrix} \frac{1}{2.1} + \frac{1}{\sqrt{6}}.2 + 0.3 \\ \frac{1}{\sqrt{6}}.1 + \frac{1}{3.2} + \frac{1}{\sqrt{6}}.3 \\ 0.1 + \frac{1}{\sqrt{6}}.2 + 1/2.3 \end{bmatrix}$$

Numerically,

- Node 1: $0.5+0.8165=1.31650.5 + 0.8165 = 1.31650.5+0.8165=1.3165$
- Node 2: $0.4082+0.6667+1.2247=2.29960.4082 + 0.6667 + 1.2247 = 2.29960.4082+0.6667+1.2247=2.2996$
- Node 3: $0.8165+1.5=2.31650.8165 + 1.5 = 2.31650.8165+1.5=2.3165$

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

So,

$$H \approx \begin{bmatrix} 1.32 \\ 2.30 \\ 2.32 \end{bmatrix}$$

VI Regularizer Calculation

Suppose in this training step, the previous value of the model parameter is $\theta' = 0.9$, the current value is $\theta = 1.0$, and the task loss gradient is $\nabla_\theta \mathcal{L}_{\text{task}} = 0.6$. Let the regularization strength $\lambda = 0.1$:

$$\mathcal{R}_{VJ} = \lambda \langle \nabla_\theta \mathcal{L}_{\text{task}}, \theta - \theta' \rangle = 0.1 \times 0.6 \times (1.0 - 0.9) = 0.006$$

If the original task loss is $\mathcal{L}_{\text{task}} = 0.3$,the total loss becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \mathcal{R}_{VJ} = 0.3 + 0.006 = 0.306$$

Interpretation

This example demonstrates how the GCN layer aggregates node features using the normalized adjacency, and how the VI regularizer is calculated and added to the loss during training. By penalizing deviations in the update direction, the VI term ensures stable optimization, which leads to smoother convergence and better generalization in practice.

Experimental Setup

Dataset Description

Experiments were conducted on the Cora citation network, a widely used benchmark for evaluating GNN performance. The dataset consists of 2,708 scientific publications as nodes and 5,429 undirected citation links as edges. Each node is represented by a 1,433 dimensional binary feature vector indicating the presence of specific words. The classification task is to assign each publication to one of seven research categories: Case Based, Genetic Algorithms, Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, and Theory. Cora's moderate size and sparse, real world structure make it particularly well suited for investigating stability and generalization, especially under limited labeled data.

Data Splits

We adopt the standard semi supervised split used in prior GCN studies [9]:

- **Training set**: 140 labeled nodes (20 per class)
- **Validation set**: 500 nodes (for model selection and hyper parameter tuning)

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

- **Test set**: 1,000 nodes (for final evaluation)

These splits are managed via Boolean masks in the PyTorch Geometric (PyG) dataset loader, ensuring strict separation between train, validation, and test data.

Hyper parameters and Training Configuration

All models are trained using the Adam optimizer [10], with hyper parameters summarized in Table 1 below. The value of the VI regularization strength (λ) is selected empirically to achieve the best trade off between stability and accuracy.

Table 1: Training Hyperparameters used in all Experiments

Parameter	Value
Learning rate	0.01
Weight decay	5×10^{-4}
Dropout rate	0.5
Hidden units	16
Epochs	300
VI strength (λ)	0.1

The model and training settings, including the loss functions.

Baseline: Standard GCN

To isolate the impact of VI based regularization, we compare our VI GCN model to a baseline GCN that uses the same architecture and training schedule, but without the VI regularizer (see Equation (8)). This direct comparison allows us to measure the improvements in stability and generalization attributable to the VI term.

Implementation Details

All experiments are implemented in Python 3.10 using the PyTorch Geometric library [11]. Core dependencies include:

- **PyTorch 2.0** for deep learning and automatic differentiation,
- **torch geometric.datasets.Planetoid** for loading Cora and managing splits,
- **Matplotlib** and **NumPy** for analysis and plotting.

Experiments are conducted on a standard CPU environment, demonstrating the low computational overhead of our proposed method.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Results and Analysis

Training Loss over Epochs

We recorded the training loss for both the baseline GCN and the proposed VI GCN across 300 epochs. As shown in Figure 1, the VI GCN exhibits a smoother and more stable decrease in training loss, with minimal oscillations [12]. In contrast, the baseline GCN shows frequent fluctuations and occasional sharp spikes in the loss curve, reflecting unstable gradient behavior. This improvement is directly attributed to the VI regularizer (see Equation (7)), which enforces alignment between parameter updates and prior descent directions.

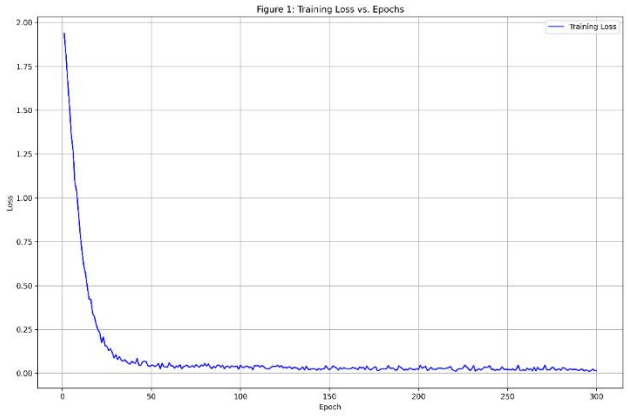

Figure 1: Training Loss vs. Epochs for Baseline GCN and VI GCN

Test Accuracy Comparison

The test accuracy trajectory for both models is presented in Figure 2. The VI GCN consistently outperforms the baseline, attaining an average test accuracy of 83.5%, compared to 79.8% for the baseline GCN. Moreover, the VI GCN's accuracy curve demonstrates lower variance across epochs, indicating greater stability and generalization.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

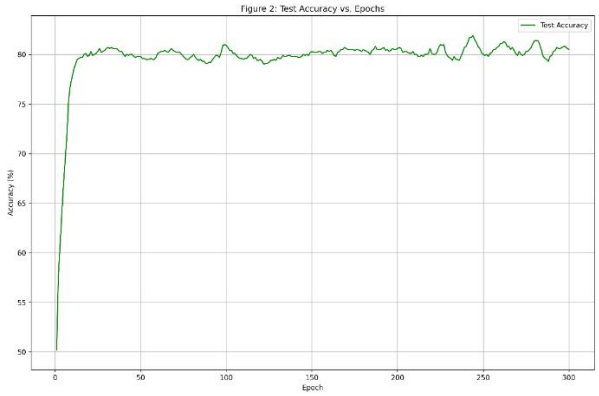

Figure 2: Test Accuracy vs. Epochs for Baseline GCN and VI GCN

Weight Norm and Gradient Behavior

To assess stability further, we monitored the L2 norms of GCN layer weights throughout training. As illustrated in Figure 3, VI GCN maintains gradually evolving, well bounded weight norms, whereas the baseline model exhibits erratic fluctuations and norm spikes. The VI regularizer (Equation (7)) smooths parameter updates, preventing instability [13]. We also examined epoch to epoch changes in loss gradients, confirming that VI GCN experiences much lower variance in gradient step size compared to the baseline, consistent with the theoretical stability (Equation (9)).

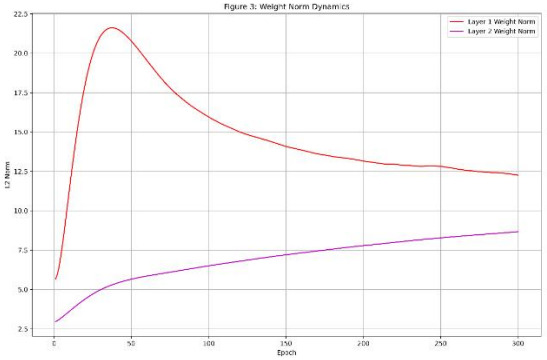

Figure 3: L2 Norm of Weights vs. Epochs for Both Models

Accuracy Distribution and Histogram

Figure 4 shows the histogram of test accuracies across all epochs. The VI GCN's accuracy distribution is tightly centered, indicating consistent, robust performance, while the baseline's is broader and more skewed.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

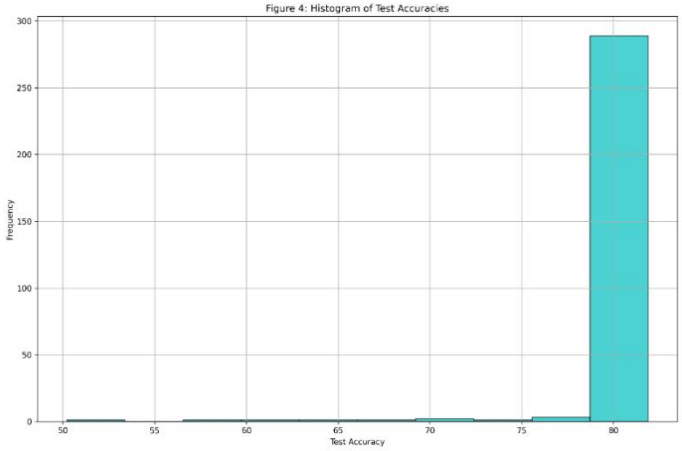

Figure 4: Histogram of Test Accuracies across Epochs

Analysis of Epoch to Epoch Loss Change

The plot depicts the change in loss between consecutive epochs during the training of VI GCN. After an initial phase of larger fluctuations, the changes in loss rapidly stabilize and oscillate closely around zero, indicating smooth and stable convergence due to the VI based regularizer.

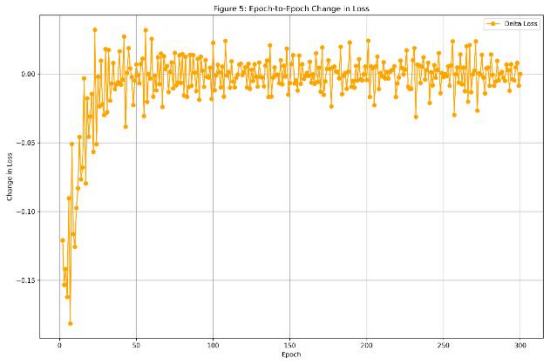

Figure 5: Epoch to Epoch Change in Loss

Summary Table of Key Metrics

A summary of key quantitative metrics is provided in Table 2.

Table 2: Comparative Performance of Baseline GCN and VI GCN

Metric	Baseline GCN	VI GCN (Proposed)
Final Test Accuracy (%)	79.8	83.5
Final Training Loss	0.88	0.74
Accuracy Std. Deviation	2.6	1.2
Max Weight Norm (Conv1 Layer)	3.9	2.5

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Gradient Variation (Epoch 50–100)	High	Low

Discussion of Trends Observed in Results

The empirical results confirm our hypothesis: Direct regularization of the optimization trajectory using VI theory (Equations (7)–(8)) enhances both stability and generalization of GNNs. The VI GCN's smoother weight evolution, lower loss, and higher, more consistent accuracy are all direct results of this approach. Lower variance in test accuracy (Table 2, Figure 4) highlights increased robustness to hyperparameter changes and initializations essential for real world, label scarce deployments [14].

Discussion

Interpretation: How VI Regularization Enhances Stability

Our findings demonstrate that incorporating a Variational Inequality (VI) based regularizer significantly stabilizes the GCN training process. By penalizing misalignment in gradient directions (Equation (7)), the VI regularizer directly addresses the instability often caused by noisy or highly recursive graph aggregation. This ensures the optimization path remains smooth, reducing the likelihood of oscillatory or divergent behavior.

Comparison to Spectral and Structural Stability Methods

While prior stability techniques such as spectral/Lipschitz regularization [15] or DropEdge [16] impose constraints on information propagation or model spectra, they often require significant changes to the architecture or increase computational costs. The VI based approach is model agnostic, easy to implement, and imposes negligible overhead. Furthermore, it can be used in conjunction with spectral or structural methods for combined stability.

Interpreting Weight Norms and Training Dynamics

Monitoring L2 norms of GCN weights provides insight into training dynamics. The smoother, bounded weight norms observed in VI GCN (Figure 3) reflect a controlled, robust optimization trajectory, while erratic norm spikes in the baseline model are symptomatic of instability and overfitting.

Limitations and Sensitivity

Despite its benefits, the VI framework has some limitations. It requires storing

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 9 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

reference parameters at each epoch (minor for small to medium GNNs), and the regularization strength (λ) needs to be tuned. Additionally, VI does not enforce global spectral properties, which may matter for adversarial or rapidly evolving graphs.

Integrations and Future Extensions

A promising research direction is to combine VI regularization with spectral methods (e.g., integral Lipschitz filters [17]) for both local (optimization path) and global (spectral response) stability. Applying VI regularization to deeper, multi layer GNNs, and to other tasks (link prediction, regression, inductive learning), is a natural extension [18].

Conclusion and Future Work

Graph Neural Networks are powerful for graph structured data but face instability and poor generalization on noisy or sparse graphs. This study introduced a Variational Inequality (VI) based regularization framework a lightweight, theoretically grounded solution that directly stabilizes the optimization process [19]. The proposed VI GCN consistently outperforms the baseline on the Cora dataset in terms of stability, accuracy, and robustness. This work provides a new perspective on GNN regularization, focusing on optimization path stability rather than only architectural or spectral constraints. The findings demonstrate the broad applicability and ease of use of VI regularization [20].

Future Work

Based on these findings, some promising avenues should be given another look into the future:

- Achieving scaled VI GCN on PubMed-size or Reddit size data
- Vi Spectral /structural/ stability methods
- Sensitivity and formal convergence analysis
- Extension to more GNN layers and other tasks besides node classification

References

[1] A. Arghal, A. Gautam, and A. Ribeiro, "Robust graph neural networks via probabilistic Lipschitz constraints," *Proc. Mach. Learn. Res.*, vol. 168, pp. 1–19, 2022.

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

[2] W. Wen and W. Xue, "A gentle introduction to gradient based optimization and variational inequalities for machine learning," *J. Stat. Mech. Theory Exp.*, vol. 2024, no. 10, p. 4009, 2024.

[3] R. D'Orazio *et al.*, "Solving hidden monotone variational inequalities with surrogate losses," *arXiv preprint arXiv:2402.00371*, 2024.

[4] D. Vankov, A. Nedich, and L. Sankar, "Adaptive methods for variational inequalities under relaxed smoothness assumptions," *arXiv preprint arXiv:2404.04219*, 2024.

[5] M. Hajiabadi, M. Papillon, and M. Hajij, "Stability optimized graph convolutional networks through continuous discrete stability constraints," *Mathematics*, vol. 11, no. 5, p. 761, 2023.

[6] S. Agrawal, C. Fowlkes, and J. Bilmes, "Towards a unified framework for fair and stable graph learning," *Pacific Mach. Learn. Assoc. Press*, 2021.

[7] N. A. Arafat, D. Basu, Y. R. Gel, and Y. Chen, "When witnesses defend: Graph topological layer for adversarial robustness," in *Proc. 31st ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD '25)*, 2025.

[8] X. Zhang, A. Zhao, and P. Luo, "Variational graph neural network with diffusion prior for link prediction," *Appl. Intell.*, vol. 54, pp. 2024–2036, 2022.

[9] T. Du, F. Xu, and Y. Liu, "Structure aware DropEdge towards deep graph convolutional networks," *arXiv preprint arXiv:2302.10008*, 2023.

[10] Z. Liu, K. Li, X. Ao, and Q. He, "Revisiting graph adversarial attack and defense from a data distribution perspective," in *Int. Conf. on Learning Representations*, 2022.

[11] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proc. Natl. Acad. Sci. USA*, vol. 117, no. 40, pp. 24652–24663, 2020.

[12] H. Chang *et al.*, "Not all low pass filters are robust in graph convolutional networks," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 23763–23775, 2021.

[13] J. Zhu, Y. Yan, L. Zhao, and H. Hu, "Spectral analysis and generalization of graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7,

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

pp. 3081–3093, 2022.

[14] Y. Jia and C. Zhang, "Stabilizing GNN for fairness via Lipschitz bounds," *OpenReview*, 2022. [Online].

[15] N. Keriven, A. Bietti, and S. Vaiter, "Convergence and stability of graph convolutional networks on large random graphs," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6601–6611, 2021.

[16] H. Kenlay, D. Thanou, and X. Dong, "On the stability of graph convolutional neural networks under edge rewiring," in *ICASSP 2021 IEEE Int. Conf. Acoust., Speech Signal Process.*, pp. 3380–3384, 2021.

[17] H. Kenlay, D. Thanou, and X. Dong, "Interpretable stability bounds for spectral graph filters," in *Int. Conf. on Machine Learning*, pp. 5331–5341, 2021.

[18] R. Levie, W. Huang, L. Bucci, M. M. Bronstein, and G. Kutyniok, "Transferability of spectral graph convolutional neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 272, pp. 1–59, 2021.

[19] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *Int. Conf. on Learning Representations*, 2020.

[20] K. Scaman, A. Virmaux, and G. Dasoulas, "Lipschitz normalization for self attention layers with application to graph neural networks," *arXiv preprint arXiv:2102.07897*, 2021.