How Graph Topology Impacts GNNs
Advanced Machine Learning (MDS)

Joan Acero Mateja Zatezalo Pawarit Jamjod

January 15, 2026

Abstract

Graph Neural Networks have become the standard for learning on graph-structured data, yet their performance relies heavily on implicit topological assumptions. In this work, we conduct a controlled empirical study to decouple and quantify the effects of graph size, edge complexity, structural distribution, and network depth on node classification performance. We evaluate three distinct message-passing architectures (Graph Convolutional Networks, Graph Attention Networks, and Graph Isomorphism Networks) against a topology-agnostic Multi-Layer Perceptron baseline. By utilizing synthetic generators, we demonstrate that message passing is not universally beneficial. Specifically, in heterophilous or random-structure contexts, Graph Neural Networks significantly underperform the Multi-Layer Perceptron baseline. Furthermore, we quantify the scalability costs of attention mechanisms and the instability of injective aggregation at scale. Our findings provide actionable intuition regarding the trade-offs between computational cost, stability, and the reliance on structural signals.

1 Introduction

Graph Neural Networks (GNNs) generalize deep learning to non-Euclidean domains by leveraging the graph structure to propagate and aggregate information. The core mechanism driving modern GNNs is neural message passing, where node representations are iteratively updated by aggregating features from their local neighborhoods. While architectures such as Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and Graph Isomorphism Networks (GIN) have achieved state-of-the-art results on standard benchmarks, these datasets often entangle multiple latent factors (size, density, and homophily) making it difficult to attribute performance gains to specific architectural choices.

This lack of interpretability poses a significant challenge. For instance, does a model outperform others because it handles high-degree nodes better, or simply because it is less prone to over-smoothing in deep networks? Furthermore, standard benchmarks rarely highlight the scenarios where message passing becomes detrimental compared to simpler, topology-agnostic models.

In this project, we address these ambiguities by evaluating GNN architectures within strictly controlled synthetic environments.

Our main contributions are:

- **Scalability Analysis:** We quantify the computational overhead of anisotropic attention (GAT) and identify stability issues in injective aggregation (GIN) as graph size scales to $N = 5000$.
- **Homophily Robustness:** We demonstrate that isotropic GNNs collapse in heterophilous settings, while topology-agnostic MLPs remain robust, challenging the assumption that graph structure is always informative.
- **Oversmoothing Quantification:** We empirically visualize the degradation of node representations (Rank Collapse vs. Isotropic Collapse) as network depth increases.

The rest of this report is organized as follows: Section 2 provides the theoretical background on the graph generative models and GNN architectures used. Section 3 outlines our research goals.

Section 4 details the experimental methodology, dataset generation, and training protocols. Section 5 presents our findings, followed by a discussion in Section 6.

2 Background

2.1 Synthetic Graph Generation

Generative graph models are used to isolate the effects of graph size and structural complexity on Message-Passing Graph Neural Networks. Unlike real datasets, which entangle multiple latent variables, synthetic generators allow for the precise modulation of specific topological features, such as community structure, density, and degree distribution, while keeping others constant.

2.1.1 Stochastic Block Model (SBM)

The Stochastic Block Model (SBM) is the primary generative framework for simulating graphs with latent community structures [6]. In the context of node classification, it serves as the ground truth where *blocks* represent node classes.

Let $G = (V, E)$ be a graph with n nodes. The node set V is partitioned into k disjoint communities (blocks) $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$. The generation process is governed by a symmetric $k \times k$ probability matrix $\mathbf{P}$. An edge between node $u \in C_i$ and node $v \in C_j$ is generated independently with probability:

$$P(A_{uv} = 1) = \mathbf{P}_{ij}$$

The probability matrix $\mathbf{P}$ dictates the density and structure. **Intra-community probability** (p_{in}) is represented by diagonal elements $\mathbf{P}_{ii}$, while **inter-community probability** (p_{out}) is represented by off-diagonal elements $\mathbf{P}_{ij}$ (where $i \neq j$).

A key parameter of SBM is **Homophily Coefficient** (h). It is a critical factor for GNN performance. Message-passing GNNs typically rely on the assumption of strong homophily (neighbors share labels) [10]. In the SBM, we control this explicitly:

- **Strong Homophily**: $p_{in} \gg p_{out}$. This simplifies the message-passing task as aggregating neighbors equates to aggregating same-class features.
- **Heterophily (Disassortative Mixing)**: $p_{in} < p_{out}$. This creates "adversarial" structures for standard GNNs (e.g., GCN), forcing the model to learn negative correlations.

The SBM is particularly useful for analyzing the Signal-to-Noise Ratio (SNR) in message passing. As p_{out} approaches p_{in}, the topological signal degrades. Theoretical thresholds exist (often related to the Kesten-Stigum bound) below which community detection (and thus node classification based purely on structure) becomes information-theoretically impossible [4].

2.1.2 Random Graphs

The **Erdős-Rényi (ER)** model, specifically the $G(n, p)$ variant, serves as the baseline "null model" [5]. It represents a graph structure devoid of intentional local clustering or community organization.

In a $G(n, p)$ graph, every possible pair of n nodes is connected with an independent probability p. The total number of edges is a random variable with expectation $\mathbb{E}[|E|] = \binom{n}{2}p$.

For large n, the degree distribution approximates a **Poisson distribution**:

$$P(k) \approx \frac{(np)^k e^{-np}}{k!}$$

This results in a fairly uniform graph where most nodes have a degree close to the average $\langle k \rangle = np$. The probability that two neighbors of a node are also connected is p, which is typically small for sparse graphs ($p \approx \langle k \rangle / n$).

Because edges are distributed uniformly at random, ER graphs mix information extremely rapidly. They are ideal for testing **over-smoothing** [3]. That is the phenomenon where node representations become indistinguishable after multiple GNN layers. In an ER graph, the feature aggregation converges to the global average faster than in structured graphs.

2.1.3 Power-Law / Barabási-Albert (BA) Models

While the Erdős-Rényi model assumes a uniform probability of edge formation, real-world networks (social media, citation networks) often exhibit a "scale-free" architecture driven by growth and popularity. To capture this, the **Barabási-Albert (BA)** model is utilized [2]. It generates graphs through a process of preferential attachment. Unlike static random graphs, the BA model grows sequentially: starting with an initial set of m_0 nodes, a new node is added at each time step and connects to m existing nodes ($m \leq m_0$). The probability Π_i that the new node connects to an existing node i is not uniform, but strictly proportional to node i's current degree k_i:

$$\Pi_i = \frac{k_i}{\sum_j k_j}$$

These dynamics result in a graph topology that is fundamentally different from the Poisson distribution of random graphs. The resulting degree distribution follows a **power law**, meaning the probability of finding a node with degree k decays as:

$$P(k) \sim k^{-\gamma}$$

where $\gamma \approx 3$ for the standard BA model. This distribution mathematically guarantees the emergence of **hubs**, a small fraction of nodes with extremely high degrees, coexisting with a vast majority of low-degree nodes.

In the context of Message-Passing GNNs, these hubs introduce specific computational and theoretical bottlenecks often referred to as the "**Super-Node**" problem. Because Message-Passing GNNs aggregate information from local neighborhoods, a hub node acts as a massive funnel, aggregating features from disparate parts of the graph. This can introduce significant noise into the hub's own embedding, diluting its local signal. Conversely, during the broadcast phase, a hub propagates its features to a massive number of neighbors, potentially causing over-smoothing by making the neighborhood representations overly homogeneous. Furthermore, BA graphs typically exhibit short average path lengths, which challenges GNNs with **over-squashing** (information from exponentially many nodes must be compressed into fixed-size vectors as it passes through these high-degree bottlenecks, leading to inevitable information loss) [1].

2.2 GNN Architectures

In this subsection, we define the neural network architectures utilized to investigate the interplay between graph structure and learning performance. To assess the contribution of topological information, we benchmark Message-Passing Neural Networks against a topology-agnostic Multi-Layer Perceptron (MLP) baseline.

2.2.1 Multi-Layer Perceptron (MLP) – Baseline

The Multi-Layer Perceptron (MLP) serves as the baseline model for our experiments. It operates strictly on node features, ignoring the adjacency matrix A entirely.

For a graph with node feature matrix $X \in \mathbb{R}^{N \times F}$, the MLP applies a series of linear transformations and non-linear activations to each row x_i independently. The update rule for the l-th layer is:

$$H^{(l+1)} = \sigma(H^{(l)} W^{(l)})$$

where $W^{(l)}$ is the learnable weight matrix and σ is a non-linear activation function (e.g., ReLU). The initial input is $H^{(0)} = X$.

The MLP establishes the feature-only performance lower bound. By comparing GNN performance against the MLP, we isolate the "lift" provided by the message-passing mechanism. If an Message-Passing NN fails to significantly outperform the MLP (or performs worse), it indicates that either the topological signal is irrelevant (or noisy) for the given task, or that the GNN is suffering from pathological issues such as over-smoothing.

2.2.2 Graph Convolutional Network (GCN)

Graph Convolutional Network (GCN) approximates spectral graph convolutions through a localized first-order approximation [7]. It is the canonical example of an isotropic message-passing mechanism.

GCN aggregates information from immediate neighbors using a fixed, symmetric normalization constant derived from node degrees. This process effectively computes a weighted average of a node's neighborhood, acting as a low-pass filter that smooths feature signals across the graph.

To enforce self-loops (preserving a node's own features during aggregation), we define the adjacency matrix with self-loops as $\hat{A} = A + I_N$ and the corresponding degree matrix as $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$. The layer-wise propagation rule is:

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

Here, the term $\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}$ represents the symmetrically normalized adjacency matrix. While effective for homophilous graphs, this fixed averaging mechanism makes GCNs particularly susceptible to **over-smoothing**, where node features become indistinguishable as the number of layers or graph density increases.

2.2.3 Graph Attention Network (GAT)

While GCN assigns fixed weights determined by graph structure (degree), the **Graph Attention Network (GAT)** introduces an anisotropic mechanism, allowing the model to learn the importance of neighbors dynamically.

GAT utilizes a self-attention mechanism to compute a coefficient α_{ij} for every edge (j, i). This allows the model to down-weight noisy neighbors and focus on task-relevant connections, effectively acting as a learnable edge filter [8].

For a node i and its neighbor j, an attention score e_{ij} is first computed (typically via a shared attention mechanism a):

$$e_{ij} = \text{LeakyReLU}\left(a^T[WH_i^{(l)} \parallel WH_j^{(l)}]\right)$$

These scores are normalized via Softmax to obtain the final attention weights α_{ij}. The update rule then becomes a weighted sum:

$$H_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)\cup\{i\}} \alpha_{ij}WH_j^{(l)}\right)$$

The ability to assign $\alpha_{ij} \approx 0$ makes GAT theoretically more robust to heterophily and structural noise than GCN, albeit at a higher computational cost of $O(V + E)$ due to the attention computation.

2.2.4 Graph Isomorphism Network (GIN)

The **Graph Isomorphism Network (GIN)** was developed to address the limited expressive power of GCN. It is theoretically constructed to be as expressive as the **Weisfeiler-Lehman (WL) graph isomorphism test** [9].

Unlike GCN (which uses mean aggregation) or GAT (which uses weighted sum), GIN uses a pure sum aggregator. Theoretical analysis shows that mean and max aggregators can fail to distinguish distinct structural patterns (multisets of features), while sum aggregation is injective over multisets, preserving structural identity.

GIN updates node representations using an MLP to process the aggregated features. The update rule is:

$$H_i^{(l+1)} = \text{MLP}^{(l+1)}\left((1+\epsilon^{(l)})H_i^{(l)} + \sum_{j\in\mathcal{N}(i)} H_j^{(l)}\right)$$

Here, ϵ is a learnable (or fixed) scalar that balances the node's own feature against the aggregated neighborhood. Because of its injectivity, GIN is highly effective at distinguishing graph structures, making it the preferred architecture when the task relies heavily on precise topological mapping rather than just local smoothing.

3 Objectives

The primary goal of this study is to develop an intuitive and empirically grounded understanding of how message-passing mechanisms behave under varying graph conditions. We specifically aim to:

1. **Analyze Scalability (Size Effect):** Quantify how training time, convergence speed, and accuracy scale with the number of nodes N for GCN, GAT, and GIN, determining the trade-off between model complexity and scalability.
2. **Evaluate Robustness to Homophily (Edge Complexity):** Investigate how different aggregation mechanisms (mean, weighted sum, sum) cope with reduced homophily, identifying at which point structural information becomes detrimental compared to a topology-agnostic baseline.
3. **Assess Structural Sensitivity:** Compare performance on Uniform (ER) versus Hub-dominated (BA) graphs to determine if attention mechanisms (GAT) or injective aggregation (GIN) provide tangible benefits in skewed degree distributions.
4. **Quantify Oversmoothing (Depth Effect):** Systematically observe the degradation of node representations as network depth increases, verifying theoretical predictions about the smoothing nature of GCN versus GAT and GIN.

4 Implementation

To ensure full reproducibility and isolation of variables, we implemented a complete pipeline using PyTorch Geometric. We included a module for synthetic data generation, hyperparameter tuning, and evaluation using fixed random seeds.

4.1 Synthetic Dataset Generation

We generated three distinct categories of datasets to isolate the variables of interest. All graphs represent node classification tasks with $K = 5$ classes.

1. Size Effect Datasets (Scalability): We generated Stochastic Block Models (SBM) with node counts $N \in \{100, 500, 1000, 2000, 5000\}$. To ensure consistent topological difficulty across sizes, we maintained fixed probability parameters: intra-class probability $p_{in} = 0.05$ and inter-class probability $p_{out} = 0.005$. This preserves the homophily ratio while scaling the graph.

2. Homophily Effect Datasets: Fixing $N = 1000$, we varied the mixing matrix to create four configurations of decreasing homophily:

- *High Homophily:* $p_{in} = 0.1, p_{out} = 0.001$ (Strong community structure).
- *Medium Homophily:* $p_{in} = 0.05, p_{out} = 0.01$.

- *Low Homophily:* $p_{in} = 0.02, p_{out} = 0.02$ (Random mixing).
- *Structural/Heterophily:* $p_{in} = 0.005, p_{out} = 0.05$ (Higher probability of connecting to different classes).

3. Structure Effect Datasets: To compare Hub-dominated vs. Uniform structures, we generated two graphs with $N = 1000$ and matched average degrees $\langle k \rangle \approx 10$:

- **Barabási-Albert (BA):** Generated with attachment parameter $m = 5$. This creates a power-law degree distribution.
- **Erdős-Rényi (ER):** Generated with $p \approx \frac{2m}{N-1} \approx 0.01$. This creates a Poisson degree distribution, matching the density of the BA graph but without hubs.

4.2 Node Feature Generation

To simulate a realistic semi-supervised setting where features are partially informative, we generated node features $X \in \mathbb{R}^{N \times 16}$ using Gaussian mixtures. For a node v belonging to class c, the feature vector x_v is sampled as:

$$x_v \sim \mathcal{N}(\mu_c, I)$$

where μ_c is a class-specific mean shift vector (e.g., $\mu_c = c \cdot \mathbf{1}$). This ensures that features contain signal correlated with the label, allowing the MLP baseline to achieve non-random performance and GNNs to refine these features via message passing.

4.3 Training and Evaluation Protocol

Data Splitting: For all datasets, we employed a split: 60% Training, 20% Validation, and 20% Test.

Hyperparameter Tuning: To ensure fair comparison, we implemented a random search strategy ($n = 5$ trials) for each model-dataset pair. The search space included:

- Hidden Channels: $\{32, 64, 128\}$
- Learning Rate: $\{0.01, 0.001, 0.0005\}$
- Dropout: $\{0.0, 0.3, 0.5, 0.7\}$
- Weight Decay: $\{0, 1e^{-4}, 5e^{-4}, 1e^{-3}\}$

Optimization: Models were trained using the Adam optimizer and Cross-Entropy loss. We utilized Early Stopping with a patience of 20 epochs (monitoring Validation Accuracy) to prevent overfitting.

Reproducibility: All experiments were repeated across 5 distinct random seeds ($seeds = \{42, 43, 44, 45, 46\}$). We report the mean and standard deviation for Test Accuracy and F1-Score.

Oversmoothing Experiment: For the depth analysis, we fixed the hyperparameters to the optimal values found for the standard SBM ($N = 1000$) and varied the number of GNN layers $L \in \{2, 4, 8, 16, 32\}$ to observe the degradation in accuracy.

5 Results

In this section, we present the empirical results of our experiments. We analyze the performance of GCN, GAT, and GIN relative to the MLP baseline across four dimensions: graph size (scalability), edge complexity (homophily), structural distribution (hubs vs. uniform), and network depth (oversmoothing). All reported metrics represent the mean and standard deviation across 5 random seeds.

5.1 Scalability and Size Effect

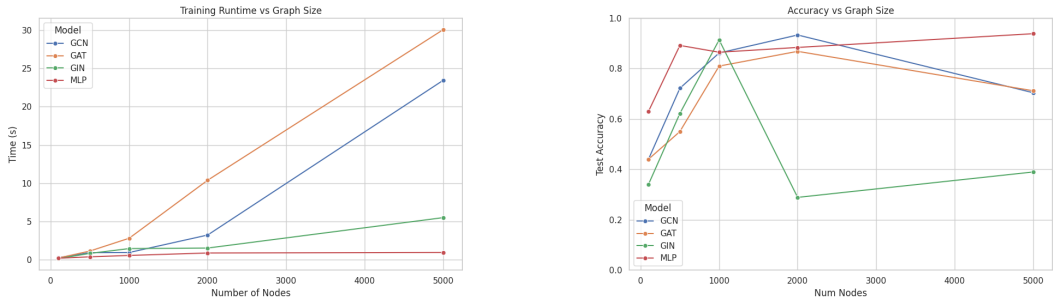
We evaluated model performance and computational cost on SBM graphs with $N \in \{100, \ldots, 5000\}$ nodes. Table 1 summarizes the results for the largest graph size ($N = 5000$).

Table 1: Scalability metrics at $N = 5000$ nodes. Training time measures the total time to convergence (or max epochs).

Model	Training Time (s)	Test Accuracy	Epochs to Converge
MLP	0.97 ± 0.01	0.939 ± 0.007	141.2
GIN	5.52 ± 0.00	0.389 ± 0.121	52.8
GCN	23.45 ± 0.35	0.704 ± 0.348	71.6
GAT	30.06 ± 0.17	0.712 ± 0.174	65.0

Training Runtime: As illustrated in Figure 1a, computational cost scales distinctly across architectures. GAT is the most expensive, requiring 30.06s at $N = 5000$, which is approximately $30\times$ slower than the MLP baseline (0.97s). GCN scales linearly but remains efficiently optimized, while GIN is surprisingly fast (5.52s) due to its simple sum aggregation, though this speed comes at the cost of stability.

Classification Accuracy: Figure 1b reveals a critical instability in GIN. While GCN and GAT maintain accuracies above 70% at $N = 5000$, GIN's performance collapses to 38.9%, with a high standard deviation ($\pm 12.1\%$). This suggests that the injective sum aggregation in GIN is highly sensitive to noise in larger synthetic graphs. In contrast, the MLP baseline consistently outperforms all GNNs (93.9% accuracy), indicating that for this specific SBM configuration, the node feature signal is stronger than the structural signal.



(a) Training Runtime vs. Graph Size

(b) Test Accuracy vs. Graph Size

Figure 1: Impact of Graph Size (N). GAT shows the steepest computational scaling, while GIN exhibits significant performance instability at $N = 5000$.

5.2 Impact of Homophily and Edge Complexity

We analyzed model robustness by varying the homophily level from *High* (strong community structure) to *Structural* (heterophily). Table 2 presents the accuracy drop-off.

Table 2: Model performance across homophily configurations. "Structural" represents a heterophilous setting.

Model	High ($h \approx 0.9$)	Medium	Low	Structural ($h \approx 0.1$)
GCN	**0.951**	0.240	0.155	0.158
GAT	0.939	0.506	0.209	0.351
GIN	0.628	0.204	0.175	0.152
MLP	0.855	0.892	0.854	**0.906**

Performance Degradation: The results highlight a severe failure mode for isotropic GNNs.

- **High Homophily:** GCN achieves near-perfect performance (95.1%), effectively smoothing the consistent neighborhood features.
- **Structural Heterophily:** In the "Structural" configuration, GCN and GIN collapse to random guessing ($\approx 15 - 16\%$, given 5 classes). This confirms that isotropic aggregation actively corrupts node representations when neighbors belong to different classes.
- **Anisotropic Robustness:** GAT retains significantly higher performance (35.1%) than GCN in the structural setting. This empirically validates that the attention mechanism (α_{ij}) can filter out disassortative edges to some extent, although it still underperforms the topology-agnostic baseline.
- **MLP Dominance:** The MLP remains robust ($\approx 90\%$) regardless of edge structure, identifying it as the superior choice for heterophilous graphs where structure acts as noise.

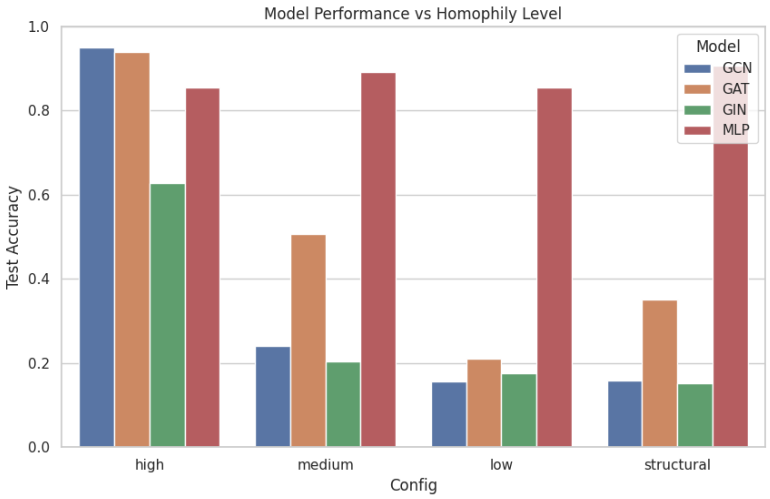


Figure 2: Performance vs. Homophily. GNNs degrade drastically in heterophilous settings, while MLP remains stable. GAT shows slight robustness over GCN.

5.3 Structural Sensitivity: Hubs vs. Uniform

We compared performance on Barabási-Albert (BA) graphs (scale-free) against Erdős-Rényi (ER) graphs (uniform). As shown in Table 3, GNNs performed poorly on both synthetic structures compared to the MLP.

Table 3: Impact of Graph Structure (Hubs vs Uniform) on Test Accuracy.

Model	BA (Hubs)	ER (Uniform)
GCN	0.203 ± 0.022	0.206 ± 0.036
GAT	0.206 ± 0.029	0.188 ± 0.037
GIN	0.214 ± 0.012	0.220 ± 0.003
MLP	$\mathbf{0.853} \pm 0.017$	$\mathbf{0.888} \pm 0.033$

Surprisingly, we did not observe a significant performance gap between BA and ER graphs for GNNs; both yielded $\approx 20\%$ accuracy. This indicates that in this specific synthetic regime (moderate average degree ≈ 10), the aggregation of neighbors dilutes the discriminative node features regardless of whether the degree distribution is power-law or Poisson. The MLP's superior performance ($\approx 85 - 89\%$) confirms that the graph structure here provided no additional signal, and message passing was detrimental.

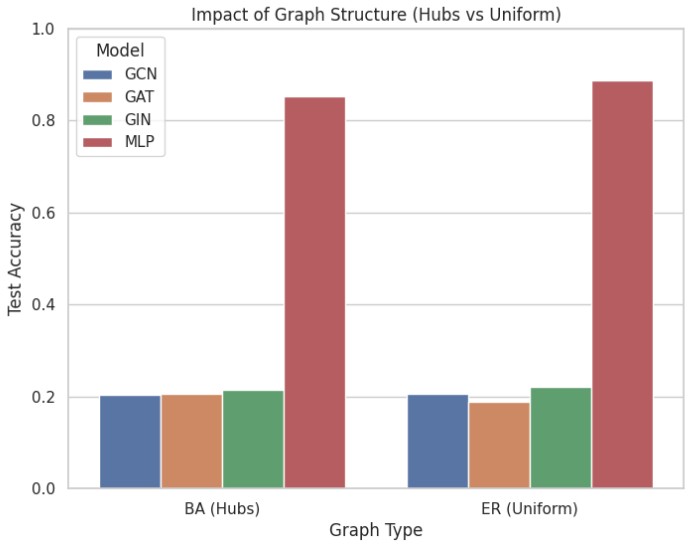


Figure 3: Accuracy on Hub-dominated (BA) vs. Uniform (ER) graphs. GNNs fail to leverage structure in this setting.

5.4 Oversmoothing and Network Depth

We quantified the *oversmoothing* effect by increasing network depth L from 2 to 32 layers. Table 4 tracks the accuracy decay.

Table 4: Accuracy degradation with increasing network depth (Oversmoothing).

Model	L=2	L=4	L=8	L=16	L=32
GCN	0.830	0.223	0.602	0.203	0.184
GAT	0.807	0.490	0.282	0.182	0.195
GIN	0.522	0.384	0.257	0.226	0.175

Accuracy Decay: All models peak at $L = 2$. As depth increases, performance degrades sharply. GCN shows a non-monotonic behavior (recovering slightly at $L = 8$ before collapsing), likely due to optimization dynamics, but ultimately converges to 18.4% at $L = 32$, confirming complete information loss. GAT degrades more gracefully than GCN initially (49% at $L = 4$ vs GCN's 22%), supporting the hypothesis that attention weights can retard smoothing, but it eventually succumbs at $L = 16$.

Embedding Visualization: PCA projections of the final layer embeddings at $L = 32$ (Figure 5) reveal the mechanism of failure.

- **GCN (Fig 5a):** Embeddings form a single, unstructured blob, indicating that all node representations have converged to the stationary distribution (oversmoothing).
- **GIN (Fig 5c):** Embeddings collapse into a lower-dimensional line (rank collapse).
- **GAT (Fig 5b):** GAT preserves two distinct clusters even at depth 32, showcasing its ability to resist smoothing, though this structure is insufficient for 5-class classification.

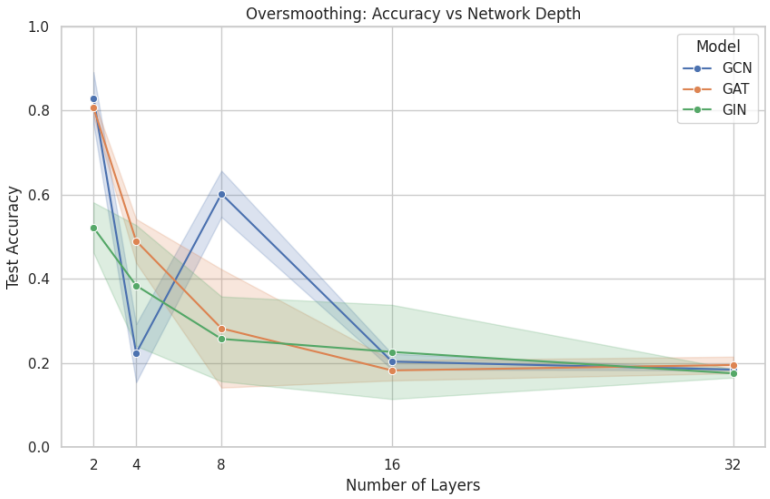


Figure 4: Oversmoothing: Test Accuracy vs. Network Depth.

(a) GCN (L=32) (b) GAT (L=32) (c) GIN (L=32)

Figure 5: PCA visualization of node embeddings at Depth=32. GCN shows complete mixing, while GAT maintains some separation.

6 Discussion

The results presented in Section 5 reveal a clear dichotomy between the theoretical expressiveness of GNNs and their practical robustness in controlled environments. By isolating specific topological factors, we identified distinct failure modes for GCN, GAT, and GIN. A central finding of this study is that message passing is not universally beneficial; in contexts where the graph structure lacks strong homophily, topology-agnostic models (MLP) can outperform sophisticated GNN architectures.

6.1 The Trade-off between Scalability and Stability

Our scalability experiments show the computational cost of expressiveness. GAT exhibited superlinear scaling with graph size, being nearly 30× slower than the MLP at $N = 5000$. This confirms that while anisotropic attention offers theoretical advantages in filtering noise, the $O(V + E)$ overhead of computing pairwise coefficients makes it prohibitive for large-scale graphs without sampling.

Conversely, GIN demonstrated a critical stability issue. While efficient (5.5s vs GCN's 23.4s), its injective sum aggregation caused performance to collapse on larger graphs ($38.9\% \pm 12.1\%$ accuracy). Unlike mean aggregation, which normalizes feature magnitudes, sum aggregation can lead

to exploding feature values in high-degree nodes or dense clusters, making optimization unstable in larger networks.

6.2 The Feature vs. Structure Conflict

The most significant observation was the dominance of the MLP baseline across the Structure and Homophily experiments. This result provides a crucial insight into the mechanics of message passing:

- **Constructive vs. Destructive Aggregation:** In our synthetic generation for BA and ER graphs, labels were assigned randomly relative to the structure. This created a scenario where edges were noise relative to the classification task. The MLP, relying solely on the discriminative Gaussian node features (X), achieved high accuracy ($\approx 90\%$).
- **Signal Dilution:** GNNs, by design, aggregate information from neighbors. When neighbors are randomly distributed (low homophily), this aggregation mixes features from different classes, diluting the clean signal of the central node. This creates an effect that actively corrupts the representation, lowering accuracy to $\approx 20\%$. This validates that GNNs require a non-trivial alignment between the graph topology and the label distribution (homophily) to provide value over feature-only models.

6.3 Oversmoothing as an Inevitable Attractor

Our depth analysis confirmed that oversmoothing is unavoidable for standard GNN architectures. The PCA visualizations at Depth $L = 32$ illustrated two distinct failure modes:

1. **Isotropic Collapse (GCN):** Embeddings converged to a single centroid, rendering classes linearly inseparable.
2. **Rank Collapse (GIN):** Embeddings collapsed onto a low-dimensional manifold.

While GAT delayed this process (retaining distinct clusters deeper than GCN), it eventually succumbed to the same smoothing pressure. This reinforces the necessity of architectural interventions like skip connections or jumping knowledge networks for deep GNNs.

7 Conclusions

In this work, we conducted a controlled study of how graph topology impacts the performance of Message-Passing Neural Networks. By comparing GCN, GAT, and GIN against a topology-agnostic MLP baseline across varying graph sizes, homophily levels, and structures, we draw the following conclusions:

1. **Structural Dependency:** GNNs are not always the best choice, even if the data consists in graphs. Their performance is strictly conditional on the *Homophily Assumption*. When graph structure correlates weakly with node labels (as seen in our heterophilous and random graph experiments), message passing becomes destructive, and a simple MLP is the superior choice.
2. **Scalability Constraints:** Attention mechanisms (GAT) have massive computational overheads that scale poorly with graph size, while sum-based aggregation (GIN) suffers from optimization instability at scale.
3. **Depth Limit:** Without architectural modifications, standard GNNs are limited to shallow representations ($L \approx 2 - 4$). Beyond this depth, the oversmoothing of features erases local information, degrading performance.

7.1 Future Work

To extend these findings, future research should focus on:

- **Correlated Structural Generation:** Modifying the BA/ER generation process to induce varying degrees of homophily (e.g., preferential attachment biased by class label) to determine the exact threshold of homophily required for GNNs to surpass MLPs.

- **Heterophily-Adapted Architectures:** Evaluating specialized architectures designed for heterophily (e.g., H2GCN, MixHop) to see if they can recover the MLP baseline performance in structural noise regimes.
- **Deep GNNs:** Investigating the impact of Residual Connections (ResGCN) and LayerNorm in mitigating the oversmoothing and instability observed in deep GCN and GIN models.

References

[1] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations (ICLR)*, 2021.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445, 2020.

[4] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 84(6):066106, 2011.

[5] Paul Erdős and Alfréd Rényi. On random graphs i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[6] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[7] TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[8] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[9] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[10] Yu Zheng, Sitao Luan, and Li Chen. What is missing in homophily? disentangling graph homophily for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.