Empirical Study of Over-Squashing in GNNs and Causal Estimation of Rewiring Strategies

Danial Saber
danial.saber@ontariotechu.ca
Ontario Tech University
Oshawa, Ontario, Canada

Amirali Salehi-Abari
abari@ontariotechu.ca
Ontario Tech University
Oshawa, Ontario, Canada

Abstract

Graph neural networks (GNNs) have exhibited state-of-the-art performance across a wide range of domains. Yet message-passing GNNs suffer from over-squashing—exponential compression of long-range information from distant nodes—which limits expressivity. Rewiring techniques can ease this bottleneck, but their practical impacts are unclear due to the lack of a direct empirical over-squashing metric. We propose a topology-focused method for assessing over-squashing between node pairs using the decay rate of their mutual sensitivity. We then extend these pairwise assessments to graph-level statistics. Coupling these metrics with a within-graph causal design, we quantify how rewiring strategies affect over-squashing on diverse graph- and node-classification benchmarks. Our extensive empirical analyses show that most graph classification datasets suffer from over-squashing (but to various extents), and rewiring effectively mitigates it—though the degree of mitigation, and its translation into performance gains, varies by dataset and method. We also found that over-squashing is less notable in node classification datasets, where rewiring often increases over-squashing, and performance variations are uncorrelated with over-squashing changes. These findings suggest that rewiring is most beneficial when over-squashing is both substantial and corrected with restraint—while overly aggressive rewiring, or rewiring applied to minimally over-squashed graphs, is unlikely to help and may even harm performance. Our plug-and-play diagnostic tool lets practitioners decide whether rewiring is likely to pay off.

CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Graph Neural Networks, Over-Squashing, Rewiring

ACM Reference Format:
Danial Saber and Amirali Salehi-Abari. 2025. Empirical Study of Over-Squashing in GNNs and Causal Estimation of Rewiring Strategies. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761157

CIKM '25, November 10–14, 2025, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761157

1 Introduction

Graph Neural Networks (GNNs) [18, 24, 31] have become a powerful learning framework for graph-structured data. Message-passing Neural Networks (MPNNs) [17]—a prominent subclass of GNNs—iteratively aggregate messages from neighboring nodes at each layer, enabling information propagation across the graph through layer stacking. To enable interactions between distant nodes, deeper networks with more layers are often required [6]. However, as the number of layers increases, the receptive field of each node (i.e., the set of nodes that influence a node's representation through message passing) can expand rapidly, leading to excessive compression of information into fixed-size node representations. This phenomenon, known as *over-squashing* [3], ultimately hampers effective information flow and learning.

As over-squashing is strongly connected with the topological properties of input graphs (e.g., commute time and effective resistance [7, 13]) , most of its mitigation approaches are rewiring techniques [3, 4, 16, 26, 33] , which modify a graph's connectivity to improve information flow between distant, weakly connected nodes. Despite its promise, the effectiveness of rewiring techniques remains challenging to assess due to the absence of a direct, empirical measure of over-squashing. The Jacobian norm offers a formal foundation for measuring over-squashing, but it is computationally prohibitive, and does not isolate the graph's topology effect on over-squashing due to its high dependency on the model's choices and parameters. Due to these limitations, effective resistance has emerged as a proxy [7, 13], which offers relative insights—e.g., which of two node-pairs (or two graphs) is more susceptible to suffering over-squashing. However, this measurement lacks a clear threshold to identify (e.g., whether or not over-squashing occurs for a node pair or a graph) or quantify the extent of over-squashing. This ambiguity obscures the need or justification for rewiring as an over-squashing mitigation strategy.

To tackle these challenges, we propose a topology-focused measurement framework for over-squashing built upon a formal characterization of over-squashing—rather than being a proxy. We quantify pairwise over-squashing by modeling node-pair sensitivity exponentially decaying with the model depth (i.e., number of layers). This assumption mirrors the over-squashing theoretical definitions of Topping et al. [33], which show that sensitivity diminishes rapidly along long paths in over-squashed graphs. Using decay rates of node pairs as a direct and interpretable indicator of over-squashing, we derive graph-level over-squashing metrics and then leverage them in a causal inference framework to evaluate the rewiring effectiveness for over-squashing mitigation. This enables a rigorous evaluation of rewiring strategies on over-squashing across a diverse range of graph and node classification tasks.

We applied our measurement framework to address four key questions across node- and graph-classification tasks: *(extent)* How much over-squashing does each dataset exhibit?; *(mitigation)* How effectively do current rewiring methods reduce it?; *(translation)* Do these reductions translate into performance gains?; and *(responsiveness)* How responsive is each dataset to over-squashing mitigation? Our results show that most graph classification datasets suffer from substantial over-squashing, making rewiring a sensible intervention. Among the rewiring strategies, DIGL [16] is the most effective in mitigating over-squashing, yet FoSR [20] and BORF [26] exhibit stronger correlations between over-squashing reduction and performance improvements (i.e., more effective in translation). Every graph dataset is responsive—over-squashing falls after rewiring—except Reddit-B, which is counter-responsive. In most node-classification datasets, rewiring often increases over-squashing, and performance changes are independent of it (i.e., no translation). Also, node datasets are mostly counter-responsive to the rewiring. Our findings suggest rewiring is most effective when over-squashing is significant, as in most graph-classification datasets, and less justified when over-squashing is minimal (as in most node-classification datasets). Our plug-and-play diagnostic framework enables practitioners to quantify over-squashing and decide—before expending training cycles—on applying rewiring.

2 Preliminaries and Related Work

We consider an undirected graph $G = (V, E)$ with n nodes and m edges, represented by its adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. To include self-loops, we define $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{n\times n}$ being the identity matrix. Each node v has a d-dimensional feature vector $\mathbf{x}_v \in \mathbb{R}^d$. Message-Passing Neural Networks (MPNNs) propagate information through the graph by the L-stack of graph convolution layers (or message-passing layers), where L represents the depth of the model. Each layer ℓ is composed of the aggregator $\text{agg}^{(\ell)}$ function (e.g., mean) and update $\text{up}^{(\ell)}$ function (e.g., MLP). Node v's representation $\mathbf{h}_v^{(\ell)}$ at layer ℓ is updated by

$$\mathbf{h}_v^{(\ell)} = \text{up}^{(\ell)}\left(\mathbf{h}_v^{(\ell-1)}, \text{agg}^{(\ell)}\left(\{\mathbf{h}_u^{(\ell-1)} : u \in N_v\}\right)\right), \tag{1}$$

where $N_v = \{u \in V : (u, v) \in E\}$ denotes the 1-hop neighborhood of node v. The number of layers L (i.e., model depth) determines how far information flows across the graph, defining the *receptive field* for each node v—the set of nodes whose initial features (at layer 0) can influence v's final representation $\mathbf{h}_v^{(L)}$. As each layer propagates information one hop further, the receptive field grows with depth.

Over-Squashing. When a task relies on long-range information exchange between distant node pairs, effective information propagation requires the model depth L to be at least as large as the geodesic distance between nodes, allowing them to fall within each other's receptive fields. However, in most real-world graphs, receptive fields grow exponentially with the number of layers, forcing MPNNs to compress increasingly large sets of node features into fixed-width node embeddings. This excessive compression leads to information loss and reduces the model's expressivity, a phenomenon known as *over-squashing* [3]. The over-squashing of information can be understood by assessing the sensitivity of node v's representation after ℓ layers of message passing to node u's input feature $\mathbf{h}_u^{(0)}$ through the *absolute Jacobian's norm* [33]:[1]

$$\mathcal{J}_\ell(v, u) = \|\partial \mathbf{h}_v^{(\ell)} / \partial \mathbf{h}_u^{(0)}\|. \tag{2}$$

Of special interest for assessing over-squashing is *normalized Jacobian's norm*

$$\tilde{\mathcal{J}}_\ell(v, u) = \frac{\mathcal{J}_\ell(v, u)}{\sum_k \mathcal{J}_\ell(v, k)}, \tag{3}$$

which measures relative sensitivity [33, 37]—the sensitivity of node v's feature at layer ℓ to node u's initial feature, relative to v's sensitivity to all nodes. Without this normalization, the model might overestimate a node's sensitivity based solely on its absolute Jacobian norm. A small $\tilde{\mathcal{J}}_\ell(v, u)$ indicates that node v is negligibly sensitive to node u, signaling over-squashing. In severe cases—e.g., tree-like graphs [33]—both $\mathcal{J}_\ell(v, u)$ and $\tilde{\mathcal{J}}_\ell(v, u)$ decay exponentially with ℓ, causing sensitivity to vanish. This vanishing sensitivity reflects the progressive suppression of messages from u at v, a signature of over-squashing. Our over-squashing measures build upon the theoretical characterization of $\tilde{\mathcal{J}}_\ell(v, u)$, which links the decay of pairwise sensitivity in node embeddings to the number of GNN layers and the graph's topology.

Over-Squashing Measurement. The Jacobian norm (and its variants) is a principled measure of over-squashing, but has practical shortcomings. (i) It is *computationally prohibitive*: for n nodes with feature dimension d, the full Jacobian is an $(nd) \times (nd)$ matrix, requiring $O(n^2d^2)$ memory and time. (ii) It is *parameter-dependent*, varying with weight updates and model-specific hyperparameters (e.g., hidden size). (iii) It fails to isolate the graph's topological effects, being highly dependent on model choices and parameters.

To focus more directly on graph topology, recent work resorts to measuring over-squashing through the lens of effective resistance [7, 13]. Node pairs with high effective resistance are more susceptible to over-squashing [7, 13], and a graph's total effective resistance serves as a global proxy for over-squashing. However, effective resistance has key limitations: (a) it only allows relative comparisons—offering no threshold for when, or how severely, over-squashing occurs; (b) Though related to over-squashing, the effective resistance is not derived from its formal characterizations (e.g., $\tilde{\mathcal{J}}_\ell(v, u)$), leaving uncertainty about whether a given pair is truly over-squashed. To avoid these shortcomings, we approximate the *relative Jacobian norm* directly, yielding a measure that is both topology-centered and largely model-agnostic without the heavy computational cost of full Jacobian computation.

Rewiring. Rewiring—the primary mitigation for over-squashing [3]—modifies a graph's edges while keeping its nodes unchanged to improve information flow. Spatial connectivity methods add edges to shorten distances by including virtual nodes [10, 32], leveraging higher-order structures [8, 9], fully connecting the last GNN layer [3], or linking nodes within certain distances or across layers [1, 2, 5, 15, 16, 19, 27, 34]; Graph Transformers take this to the extreme by connecting all nodes via attention-based edges [21, 28, 38]. Other approaches optimize graph-theoretical properties to reduce topological bottlenecks: SDRF [33] adds edges in low-curvature regions, BORF [26] adds in minimally curved regions and prunes

[1] Some use the term *influence* for the same Jacobian-based norm quantity; we adopt *sensitivity* throughout for consistency.

highly curved edges, FoSR [20] maximizes the spectral gap, GTR [7] minimizes total effective resistance, and DIGL [16] applies diffusion-based rewiring (e.g., personalized PageRank, heat kernel) followed by sparsification.

While rewiring aims to mitigate over-squashing, its true impact is unclear; performance gains may arise from reduced over-squashing or confounding factors such as implicit regularization or altered graph smoothness. To disentangle these effects, we propose a measurement framework using causal inference to evaluate rewiring interventions.

3 Measurement and Causality Framework

We first propose a method to measure pairwise over-squashing, extend it to graph-level metrics, and then apply these metrics to causally evaluate the impact of rewiring.

3.1 Pairwise Over-Squashing Measurement

Our goal is to derive a pairwise over-squashing measure between node pairs in a graph that is (i) computed once per graph, (ii) aligned with the relative Jacobian norm as a foundation for measuring over-squashing, (iii) focused on graph topology, (iv) dependent only on model depth as a contributing factor,[2] and (v) theoretically-founded on the rigorous definition of over-squashing. We achieve (i–iv) by introducing approximations to relative Jacobian norms, and (v) by considering the exponential decay rate.

Approximation to Normalized Jacobian Norm. To quantify over-squashing, we focus on the relative Jacobian norm $\tilde{\mathcal{J}}_\ell(v,u)$, which measures node v's sensitivity to node u as a fraction of its total sensitivity to all nodes, overcoming the limitation of the absolute norm $\mathcal{J}\ell(v,u)$ that ignores total information received by v. Directly computing $\tilde{\mathcal{J}}_\ell(v,u)$ is prohibitively expensive and must be recomputed with any change in model parameters or hyperparameters, conflating topological effects with model-level factors. To address these issues, we introduce:

Proposition 3.1 (Approximation of the Normalized Jacobian Norm). *Let $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ be the adjacency matrix of an undirected graph augmented with self-loops , and assume a* linear *message-passing GNN. Then, for any pair of nodes u, v and layer depth $\ell \geq 0$, the* normalized *Jacobian norm can be written as*

$$\tilde{\mathcal{J}}_\ell(v,u) = \frac{\tilde{\mathbf{A}}^\ell_{uv}}{\sum_k \tilde{\mathbf{A}}^\ell_{kv}}, \tag{4}$$

where $\tilde{\mathbf{A}}^\ell_{uv}$ is the (u,v)-entry of $\tilde{\mathbf{A}}$ to the power ℓ.

Remark. Self-loops guarantee reachability for every choice of ℓ: e.g., in a dyad (two nodes joined by one edge), walks of even length between the nodes vanish without self-loops, but $\tilde{\mathbf{A}}$ ensures non-zero counts for all ℓ.

The proof of this proposition is in Appendix A. In practice, GNNs typically include nonlinearities (e.g., ReLU), which yield computation of nontrivial Jacobians that require recursive application of the chain rule . Moreover, certain paths in the computational graph may become inactive (e.g., due to ReLU zeroing gradients), making exact computation intractable. Thus, equality no longer holds for nonlinear MPNNs. However, linear MPNNs are empirically competitive and theoretically well-founded [14, 22, 29, 36], and omitting nonlinearity helps remove model-specific factors (see the proof of Proposition 3.1), enabling a sensitivity measure that reflects only the graph structure and the depth. We approximate the nonlinear normalized Jacobian's norm using its simplified single-computation form in Eq. 4, and henceforth denote this approximation as $\tilde{\mathcal{J}}_\ell(v,u)$. This approximation satisfies our design criteria outlined earlier: it is once-pre-computed (criterion i), derived from normalized Jacobian norms under some simplification assumptions (i.e., removing nonlinearities) (criterion ii), and only dependent on graph topology $\tilde{\mathbf{A}}$ and the layer depth ℓ, remaining mainly topology-focused and model-agnostic (criteria iii and iv).

Exponential Decay Rate as an Over-Squashing Indicator. A key signature of over-squashing is the rapid decay of sensitivity (e.g., normalized Jacobian norms) with increasing model depth ℓ. For rigor, we model this decay as exponential, similar to Di Giovanni et al. [13], and consistent with theoretical observations in tree-like graphs [33], where both $\mathcal{J}_\ell(v,u)$ and $\tilde{\mathcal{J}}_\ell(v,u)$ diminish exponentially with ℓ, leading to vanishing sensitivity:

$$\tilde{\mathcal{J}}_\ell(v,u) = N_0 e^{-k_{vu}\ell}, \tag{5}$$

where $N_0 = \tilde{\mathcal{J}}_0(v,u)$ is the initial sensitivity ($\ell = 0$), and k_{vu} is the decay rate specific to the pair (v,u).[3] A positive k_{vu} indicates over-squashing, with larger values reflecting stronger decay. Taking the natural logarithm linearizes this relationship:

$$\ln \tilde{\mathcal{J}}_\ell(v,u) = \ln N_0 - k_{vu}\ell. \tag{6}$$

To estimate k_{vu}, we fit a linear regression model of $\ln \tilde{\mathcal{J}}_\ell(v,u)$ against ℓ, where the slope corresponds to $-k_{vu}$. A negative slope (i.e., positive k_{vu}) confirms exponential decay, with the magnitude of k_{vu} reflecting the severity of over-squashing.[4] Following Di Giovanni et al. [13], we change ℓ in the interval $[\mathrm{D}, 2\mathrm{D}-1]$, where D is the graph diameter, ensuring reachability for any pair of nodes.

3.2 Graph-Level Over-Squashing Measurement

To derive a graph-level assessment, we summarize the distribution of positive decay rates using four statistics:

- **Prevalence** is the fraction of node pairs with positive decay rates ($k_{vu} > 0$). It reflects the *spread* of over-squashing across the graph.
- **Intensity** is the average of all positive decay rates, indicating the typical *strength* of over-squashing among affected node pairs.
- **Variability** is the standard deviation of positive decay rates, measuring the *consistency or disparity* in over-squashing strength across node pairs.
- **Extremity** is the largest observed positive decay rate in the graph, capturing the *worst-case* over-squashing instance.

[2] Model depth is necessary for any over-squashing measurement as the definition of over-squashing is based on it: the progressive compression of information as the model depth increases (i.e., the number of message-passing layers grows).

[3] The decay rates k_{vu} and k_{uv} need not be equal because the number of length-ℓ walks that reach the target node can differ for v and u. For instance, in a star graph with center v and a leaf u, we have $\tilde{\mathcal{J}}_1(v,u) \neq \tilde{\mathcal{J}}_1(u,v)$ since $(\sum_k \tilde{A}^1_{kv} \neq \sum_k \tilde{A}^1_{ku})$.

[4] For pairwise analyses, one can use statistics such as R^2 and p-values to assess model fit and decay trend significance. However, since our focus is on graph-level over-squashing, we aggregate pairwise decay rates into graph-level metrics and evaluate statistical significance within our causal framework.

For datasets involving multiple graphs, we compute dataset-level summaries by averaging each metric over all graphs. For ease of interpretation, we sometimes map each graph-level statistic into three ordinal categories. Intensity and Extremity categorized as *weak* (< 0.13), *moderate* (0.13–0.23), *strong* (> 0.23), based on corresponding *pairwise sensitivity half-lives* of ≥ 5, 3–5, and < 3 layers, respectively (i.e., the number of layers needed for sensitivity to halve). Under the same thresholds, variability is classified as *low* (< 0.13), *moderate* (0.13–0.23), *high* (> 0.23). Prevalence is grouped as *small* ($< 25\%$), *moderate* ($25 - 50\%$), *large* ($> 50\%$); these cut-points align with intuitive quartile boundaries: fewer than one quarter of node pairs indicates sparse over-squashing, 25–50% a moderate regime, and more than 50% an affected majority.

3.3 Causal Estimation of Rewiring Effects

We evaluate the impact of a rewiring method $\mathcal{R}$ using our graph-level over-squashing metrics (prevalence, intensity, variability, extremity) within a causal inference framework, where rewiring acts as the treatment T and our metrics are the outcomes. The *treated* graph is $\mathcal{R}(G)$ ($T = 1$) and the *control* graph is G ($T = 0$). Outcomes $Y_{\mathcal{M}}(G)$ and $Y_{\mathcal{M}}(\mathcal{R}(G))$ are measured for the control and treated graphs, respectively, with $\mathcal{M}$ representing any of our over-squashing metrics (e.g, prevalence, intensity, etc.). Treating each graph as a unit, we compare it before and after rewiring to isolate rewiring effects from structural confounders (e.g., number of nodes). To ensure valid causal attribution, we adopt standard causal inference assumptions: SUTVA, Positivity, Exchangeability, and Consistency—detailed in the full version of this paper [30].

We assess how a rewiring $\mathcal{R}$ influences the over-squashing measurement $\mathcal{M}$ for a graph G through *Individual Treatment Effect (ITE)*:

$$\mathrm{ITE}_{\mathcal{M}}(G, \mathcal{R}) = Y_{\mathcal{M}}(\mathcal{R}(G)) - Y_{\mathcal{M}}(G). \tag{7}$$

For graph classification with a dataset of N graphs $\mathcal{D} = \{G_i\}$, we compute the *Average Treatment Effect (ATE)* to quantify the overall impact of rewiring $\mathcal{R}$ across the dataset:

$$\mathrm{ATE}_{\mathcal{M}}(\mathcal{D}, \mathcal{R}) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{ITE}_{\mathcal{M}}(G_i, \mathcal{R}). \tag{8}$$

For each dataset, we evaluate the effect of rewiring $\mathcal{R}$ on prevalence, intensity, variability, and extremity. A negative ATE/ITE indicates mitigation. For example, a negative ATE on prevalence indicates that rewiring reduces the number of over-squashed node pairs; a negative ATE on intensity reflects a decrease in the average severity of over-squashing; and a negative ATE on extremity suggests that the most severe cases of over-squashing have been mitigated.

Statistical Significance of Treatment Effects. For graph classification, we test ATE significance with a two-tailed t-test and apply Bonferroni correction [35] to control for multiple comparisons. For node classification, we assess ITE significance at the node-pair level: prevalence is tested with McNemar's test [23] for paired binary data, and intensity with a paired t-test.

4 Experiments

We first measure over-squashing levels across datasets, then apply our causal framework to assess how effectively rewiring mitigates it in graph and node classification tasks.

Table 1: Statistics of graph-classification datasets, averaged over all graphs in each dataset. Color coding: weak/low/small, moderate, and strong/high/large.

	Statistic	Bioinformatics			Social Networks		
		Mutag	Proteins	Enzymes	IMDB	Collab	Reddit
Topology	#Graphs	188	1109	600	1000	5000	2000
	Nodes	18	39	33	20	74	430
	Edges	28	92	78	106	2494	712
	Diameter	8.21	11.56	10.89	1.86	1.86	9
	Components	1.00	1.07	1.24	1.00	1.00	2.48
Over-Squashing	Prevalence	5.93e-1	5.97e-1	6.03e-1	6.28e-1	5.57e-1	4.72e-1
	Intensity	1.09e-1	1.37e-1	1.30e-1	3.12e-1	2.56e-1	1.96e-2
	Variability	1.06e-1	1.34e-1	1.31e-1	1.57e-1	1.93e-1	1.88e-2
	Extremity	4.54e-1	5.71e-1	5.96e-1	5.49e-1	9.10e-1	1.35e-1

4.1 Methodology and Experimental Setup

We discuss our experimental methodology, including empirical research questions, datasets, rewiring baselines, hyperparameters, measurements, and statistical tests.

Research Questions and their Importance. In our experiments, we address four key questions for graph and node-level tasks:

- (**Q1**) *How do over-squashing measurements (i.e., prevalence, intensity, variability, and extremity) vary across datasets? Which datasets are inherently most or least susceptible under each measurement?* This question identifies which datasets are inherently more or less prone to over-squashing, guiding benchmark selection for over-squashing research and the necessity of mitigation strategies. It also informs whether over-squashing trends are dataset- or domain-specific (e.g., social vs. biological networks).
- (**Q2**) *What are the treatment effects of each rewiring method across datasets? Which rewiring strategy most (or least) effectively reduces over-squashing measurements?* This question quantifies the treatment effects of rewiring strategies and enables their comparative "effective" ranking.
- (**Q3**) *How do treatment effects correlate with performance gains for a rewiring method over datasets?* This evaluates if reducing over-squashing translates into improved generalization. By assessing the correlation between treatment effects and performance gains (i.e., the change in predictive performance before and after rewiring), we distinguish rewiring methods that improve performance by mitigating over-squashing from those whose gains are from other factors.
- (**Q4**) *Which datasets are most responsive to rewiring—that is, show the largest relative reductions (treatment effect divided by the pre-treatment value)—and which are most resistant?* Answering this question sheds light on the inherent difficulty of reducing over-squashing across different graph structures, datasets, or domains.

Datasets. We study node and graph classification datasets commonly employed in over-squashing and rewiring research [3, 20,

26, 32, 33]. Graph classification datasets are from the TUDataset benchmark [25]: three *bioinformatics* datasets of Mutag, Enzymes, and Proteins, and three *social network* datasets of IMDB-B, Collab, and Reddit-B (see Table 1 for their statistics). For node classification, we evaluate six datasets: Cora, Citeseer, Texas, Cornell, Wisconsin, and Chameleon, analyzing only their largest connected component (see Table 4 for statistics).

Rewiring Baselines (Subjects). We examine the effectiveness of five rewiring methods commonly used in mitigating over-squashing: FoSR [20], DIGL [16], SDRF [33], GTR [7][5], and BORF [26].[6]

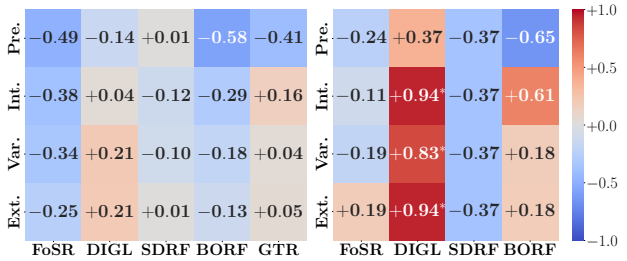
Hyperparameters. Our measurement framework has a single parameter—the message-passing depth ℓ—varied from the graph's diameter to twice its diameter [13]. Rewiring methods have their own hyperparameters (e.g., iterations for FoSR/SDRF, edges added or removed per iteration for BORF, edges added in GTR, sparsification threshold for DIGL)—tuned for specific GNN architectures (e.g., GCN, GIN). To control for this dependency, we evaluate each method using all performance-optimal configurations from prior work for each architecture—four for graph classification (GCN, GIN, R-GCN, R-GIN) and two for node classification (GCN, GIN). This avoids bias from a single configuration and ensures fair comparisons. Combinations not previously evaluated (e.g., BORF with R-GCN/R-GIN, DIGL with node-level GIN) are omitted.

Measurements. To address *Q1–Q4*, we apply our over-squashing measurement framework. For *Q1*, we measure over-squashing metrics (e.g., prevalence, intensity, etc.) on the original graphs (see Tables 1 and 4). For *Q2–Q4*, we compute $\text{ITE}_{\mathcal{M}}(G, \mathcal{R})$ for node classification and $\text{ATE}_{\mathcal{M}}(\mathcal{D}, \mathcal{R})$ for graph classification. Each rewiring method is evaluated using its performance-optimal, architecture-specific hyperparameters, yielding one ATE/ITE per (method, architecture) pair. To control for architectural dependency and avoid bias from any single configuration, we report *aggregated ATE/ITE* by averaging across all relevant configurations. Statistical significance is tested at $\alpha = 0.05$ with Bonferroni correction [35]. Performance gains (i.e., the change in task performance before and after rewiring) are taken from the original papers under the same replicated hyperparameter settings.[7]

4.2 Results on Graph Classification Tasks

We report our results of Q1–4 for graph classification.

Dataset Over-Squashing Levels (Q1): Table 1 shows that over-squashing prevalence is relatively consistent (55%–62%) across most datasets, except for Reddit-B (47%). For other measures, the bioinformatic datasets have low intensity (0.10–0.13), low variability (0.10–0.13), and high extremity (0.45–0.59). However, social network datasets exhibit more severe over-squashing: IMDB-B and Collab have the highest over-squashing intensities (0.31 and 0.25, respectively) over all datasets, with Collab showing the highest variability (0.19) and extremity (0.91). Reddit-B stands out as an outlier with all metrics an order of magnitude lower (intensity: 0.02, variability: 0.02, extremity: 0.13), confirming it is far less affected. Overall, social network datasets exhibit stronger over-squashing measurements—particularly in intensity and extremity—than bioinformatics datasets.



(a) Graph classification task **(b) Node classification task**

Figure 1: Spearman correlation coefficients of treatment effects and performance gains for every metric–rewiring pair in (a) graph- and (b) node-classification tasks. Each coefficient is computed over all GNN-baseline hyperparameter configurations; an asterisk (*) indicates significance after multiple-comparison correction. Negative values imply that stronger mitigation (smaller treatment effects) aligns with larger performance gains—the desirable direction.

Rewiring Effectiveness (Q2): Table 2 reports rewiring method's ATE across graph-classification datasets and shows that rewiring generally reduces over-squashing (green markers • dominate red markers •). FoSR reduces all metrics across datasets with a few exceptions on Collab and Reddit-B. It lowers prevalence by up to −0.04 (Proteins), intensity by −0.14 (IMDB-B), variability by −0.04 (Proteins), and extremity by −0.08 (Enzymes). DIGL mitigates over-squashing metrics in all datasets except Reddit-B, where intensity (+0.057), variability (+0.062), and extremity (+0.63) worsen; elsewhere, it reduces prevalence by 28–63%, intensity by 0.067–0.31, variability by 0.064–0.19, and extremity by 0.27–0.90. SDRF and GTR offer the weakest and most inconsistent effects. SDRF slightly reduces prevalence and intensity in most datasets by up to −0.036 (in IMDB-B for prevalence), while GTR often increases all metrics by up to +0.089 (in Mutag for intensity). BORF reduces over-squashing in most cases, showing strong extremity reduction (up to −0.21 on IMDB-B) and consistently lowering variability (−0.0026 to −0.37) and intensity, while increasing prevalence in Proteins and Enzymes.

Treatment-effect rankings show DIGL as the strongest mitigator in almost all cases, except four (three on Reddit-B). SDRF and GTR are usually the least effective, with SDRF being weakest in variability and extremity (on three datasets each) and GTR being worst for prevalence (on three datasets) and intensity (on four datasets).

Averaged over datasets (Avg. ATE in Table 2), DIGL is most effective (−40% prevalence, −0.13 intensity, and −0.31 extremity). For variability, its effect is also close to the most effective strategy (BORF with −0.11). SDRF and GTR are the least effective, with GTR showing adverse effects on prevalence and intensity, and SDRF performing worst in terms of variability and extremity. Overall, aggressive densification (e.g., DIGL) alleviates over-squashing more effectively than surgical or sparsity-preserving rewiring (e.g., FoSR, SDRF, GTR, BORF).

[5]Since prior work evaluated GTR only on graph classification, we also restrict our study of it to that task.

[6]For future work, one can easily extend our experiments to dynamic and training-time rewiring methods. In this work, to avoid hyperparameter tuning or ad-hoc design choices, we focused on the widely used methods, whose reported results are based on a shared experimental setup.

[7]The code is available at https://github.com/Danial-sb/Over-Squashing-Measurement.

Table 2: Treatment effects (ATEs) for graph classification, averaged over each GNN baseline's hyperparameter configuration. For each dataset–metric combination, the background highlights the best and worst rewiring method. Desirable negative ATEs are marked with •, and undesirable positive ATEs with •. Gain is the percentage change in classification accuracy after rewiring. † marks not statistically significant results. "Avg ATE" summarizes each method's average effect across all datasets.

Rew.	Dataset	Average Treatment Effect				Gain (%)
		Prevalence	Intensity	Variability	Extremity	
FoSR	Mutag	−2.3e-2 ± 6.3e-3 •	−1.5e-2 ± 5.5e-3 •	−2.7e-2 ± 9.3e-3 •	−3.9e-2 ± 2.1e-2 •	6.6 ± 6.5
	Proteins	−4.0e-2 ± 2.8e-2 •	−3.1e-2 ± 1.3e-2 •	−3.9e-2 ± 1.3e-2 •	−7.5e-2 ± 3.7e-2 •	3.8 ± 0.9
	Enzymes	−1.6e-2 ± 2.3e-2 •	−2.2e-2 ± 1.7e-2 •	−3.7e-2 ± 2.2e-2 •	−7.7e-2 ± 7.0e-2 •	1.6 ± 6.0
	IMDB-B	−1.2e-2 ± 5.1e-3 •	−1.4e-1 ± 4.8e-2 •	−2.2e-2 ± 1.4e-2† •	−1.4e-2 ± 2.0e-0 •	4.1 ± 6.6
	Collab	8.9e-3 ± 1.0e-3 •	−1.2e-2 ± 9.7e-3 •	−8.0e-3 ± 8.9e-3 •	−1.5e-2 ± 2.7e-3 •	9.6 ± 18.2
	Reddit-B	5.4e-4 ± 1.2e-4† •	5.3e-3 ± 5.6e-3 •	3.8e-3 ± 3.1e-3 •	2.5e-2 ± 2.0e-2 •	7.8 ± 12.7
	Avg ATE	−1.4e-2 ± 1.7e-2 •	−3.6e-2 ± 5.2e-2 •	−2.2e-2 ± 1.7e-2 •	−3.3e-2 ± 3.9e-2 •	5.6 ± 2.9
DIGL	Mutag	−5.0e-1 ± 1.9e-1 •	−1.0e-1 ± 1.3e-2 •	−9.9e-2 ± 1.3e-2 •	−4.2e-1 ± 6.6e-2 •	0.7 ± 3.0
	Proteins	−3.4e-1 ± 1.1e-1 •	−9.1e-2 ± 3.2e-2 •	−8.5e-2 ± 3.2e-2 •	−3.3e-1 ± 1.5e-1 •	−0.2 ± 0.9
	Enzymes	−2.8e-1 ± 1.7e-1 •	−6.7e-2 ± 4.6e-2 •	−6.4e-2 ± 4.8e-2 •	−2.7e-1 ± 2.3e-1 •	0.0 ± 1.4
	IMDB-B	−6.3e-1 ± 0.0e0 •	−3.1e-1 ± 0.0e0 •	−1.6e-1 ± 0.0e0 •	−5.5e-1 ± 0.0e0 •	−2.9 ± 3.3
	Collab	−5.4e-1 ± 1.5e-2 •	−2.6e-1 ± 5.8e-4 •	−1.9e-1 ± 1.1e-3 •	−9.0e-1 ± 1.3e-2 •	−18.2 ± 1.1
	Reddit-B	−9.4e-2 ± 5.9e-2 •	5.7e-2 ± 1.2e-2 •	6.2e-2 ± 1.1e-2 •	6.3e-1 ± 9.0e-2 •	−13.3 ± 3.6
	Avg ATE	−4.0e-1 ± 1.8e-1 •	−1.3e-1 ± 1.2e-1 •	−8.9e-2 ± 8.8e-2 •	−3.1e-1 ± 4.7e-1 •	−5.6 ± 7.4
SDRF	Mutag	−1.0e-2 ± 0.0e0 •	2.3e-3 ± 0.0e0 •	2.9e-3 ± 0.0e0 •	2.4e-2 ± 0.0e0 •	−0.5 ± 1.4
	Proteins	−2.3e-2 ± 6.1e-3 •	−3.9e-4 ± 1.6e-4† •	3.6e-3 ± 2.0e-3 •	1.2e-2 ± 3.8e-3 •	−0.3 ± 0.5
	Enzymes	−1.3e-2 ± 0.0e0 •	−3.1e-4 ± 0.0e0† •	−2.4e-3 ± 0.0e0 •	1.3e-2 ± 0.0e0† •	2.0 ± 2.0
	IMDB-B	−3.6e-2 ± 2.6e-2 •	−4.4e-2 ± 2.8e-2 •	1.9e-2 ± 8.7e-3† •	1.3e-1 ± 2.3e-2 •	1.0 ± 1.9
	Collab	5.5e-3 ± 1.2e-3 •	−1.7e-2 ± 9.8e-3 •	−5.6e-3 ± 3.4e-3 •	−3.7e-2 ± 1.6e-2 •	8.8 ± 17.2
	Reddit-B	−3.1e-3 ± 2.9e-3† •	−1.1e-4 ± 1.1e-3† •	−6.5e-4 ± 7.7e-4† •	−2.9e-4 ± 2.7e-3† •	−2.5 ± 4.3
	Avg ATE	−1.3e-2 ± 1.5e-2 •	−9.9e-3 ± 1.8e-2 •	2.8e-3 ± 8.6e-3 •	2.4e-2 ± 5.6e-2 •	1.4 ± 3.6
GTR	Mutag	−1.4e-3 ± 4.6e-2 •	8.9e-2 ± 3.6e-2 •	1.2e-2 ± 2.2e-2 •	5.4e-2 ± 6.4e-2 •	6.5 ± 7.1
	Proteins	−7.6e-3 ± 1.8e-2 •	−2.9e-3 ± 6.6e-3 •	−4.0e-2 ± 7.4e-3 •	−9.4e-2 ± 3.2e-2 •	3.8 ± 2.2
	Enzymes	1.1e-2 ± 1.3e-2 •	1.3e-2 ± 4.5e-3 •	−3.2e-2 ± 4.7e-3 •	−8.0e-2 ± 2.0e-2 •	5.1 ± 7.9
	IMDB-B	−2.3e-2 ± 6.3e-3 •	−1.5e-2 ± 5.5e-3 •	−2.7e-2 ± 9.4e-3 •	−3.9e-2 ± 2.1e-2 •	4.7 ± 6.9
	Collab	1.5e-2 ± 1.8e-3 •	9.0e-4 ± 1.3e-3 •	−8.8e-4 ± 2.3e-4 •	1.6e-2 ± 1.0e-2 •	0.4 ± 1.1
	Reddit-B	1.9e-2 ± 5.0e-4 •	1.4e-2 ± 3.0e-3 •	9.2e-3 ± 1.4e-3 •	4.5e-2 ± 1.0e-2 •	8.4 ± 14.6
	Avg ATE	2.2e-3 ± 1.5e-2 •	1.7e-2 ± 3.4e-2 •	−1.3e-2 ± 2.1e-2 •	−1.6e-2 ± 5.8e-2 •	4.8 ± 2.5
BORF	Mutag	−8.0e-2 ± 1.2e-2 •	−2.8e-2 ± 2.5e-2 •	−2.6e-3 ± 3.3e-3 •	−1.3e-1 ± 1.3e-1 •	2.5 ± 1.8
	Proteins	1.5e-2 ± 6.3e-3 •	−6.4e-3 ± 2.8e-3 •	−2.4e-2 ± 1.1e-2 •	−1.3e-1 ± 3.1e-2 •	0.4 ± 0.1
	Enzymes	2.1e-2 ± 2.5e-2 •	−3.5e-2 ± 1.3e-3 •	−2.9e-2 ± 1.4e-4 •	−1.4e-1 ± 1.8e-2 •	1.0 ± 1.1
	IMDB-B	−5.1e-2 ± 0.0e0 •	−7.6e-2 ± 0.0e0 •	−3.7e-1 ± 0.0e0 •	−2.1e-1 ± 9.9e-2 •	0.9 ± 1.0
	Avg ATE	−2.4e-2 ± 4.5e-2 •	−3.6e-2 ± 2.9e-2 •	−1.1e-1 ± 1.8e-1 •	−1.5e-1 ± 3.9e-2 •	1.2 ± 0.9

Rewiring vs. Performance (Q3): To assess which rewiring strategy best improves performance, we compute Spearman's correlation coefficient ρ between each ATE metric and performance gain for every method (see Figure 1a). A $\rho < 0$ indicates a desirable outcome, where reduced over-squashing (i.e., lower ATE) aligns with improved generalization. FoSR is the most effective at translating over-squashing mitigation into performance gains, with mostly moderate correlations: prevalence ($\rho = -0.49$, moderate), intensity ($\rho = -0.38$, moderate), variability ($\rho = -0.34$, moderate), and extremity ($\rho = -0.25$, weak).[8] BORF also shows negative (but comparatively weaker) correlations. SDRF and GTR show largely negligible and mixed correlations. Although DIGL achieves the greatest over-squashing reduction (see Q2), three metrics show *positive* correlations with performance gains, indicating that lower ATE values do not translate into higher performance. This paradox might be explained by DIGL's heavy edge addition, thus disrupting the graph's original topology, weakening the local-message-passing

[8] Correlation strengths follows Cohen's convention [12]: *weak* for $\rho < 0.30$, *moderate* for $0.30 \leq \rho < 0.50$, and *strong* for $\rho \geq 0.50$.

Table 3: Graph dataset responsiveness to rewiring (percentages). Negatives are desirable mitigation (metric decreases), whereas positives indicate increases. Text color denote most, second-most, second-worst, and worst responsiveness.

Dataset	Prevalence	Intensity	Variability	Extremity
Mutag	−20.2	−27.3	−22.7	−22.0
Proteins	−13.2	−19.0	−27.6	−21.0
Enzymes	−9.1	−16.9	−20.2	−18.5
IMDB-B	−23.9	−38.5	−70.1	−25.5
Collab	−23.3	−28.1	−28.4	−25.3
Reddit-B	−4.0	+96.9	+101.1	+125.9

Table 4: Statistics of node-classification datasets. Color coding: weak/low/small, moderate, and strong/high/large.

	Statistic	Cornell	Texas	Wiscon.	Cora	Citeseer	Chamel.
Topology	#Nodes	140	135	184	2485	2120	832
	#Edges	219	251	362	5096	3679	12355
	Diameter	8	8	8	19	28	11
Over-Squashing	Prevalence	5.47e-1	5.02e-1	5.46e-1	1.52e-2	1.84e-3	2.03e-1
	Intensity	8.99e-3	6.20e-3	7.95e-3	3.63e-2	3.09e-4	1.42e-1
	Variability	8.02e-3	5.72e-3	8.01e-3	2.59e-2	1.76e-3	8.37e-2
	Extremity	1.14e-1	8.04e-2	1.09e-1	2.04e-1	1.84e-2	3.94e-1

inductive bias, and inducing over-smoothing.[9] Overall, only FoSR—and, to a lesser extent, BORF—translated reduced over-squashing into performance gains, whereas SDRF and GTR have negligible impact and DIGL's notable reductions fail to improve performance, underscoring its susceptibility to over-smoothing and the disruption of the graph's original topology and inductive bias.

Dataset Treatment Responsiveness (Q4): Table 3 presents the dataset-level responsiveness to rewiring—defined as the ratio of average treatment effects (across methods) to the original dataset over-squashing measurement. Negative values indicate over-squashing mitigation, where positive values show an increase in the over-squashing metric. Social network datasets IMDB-B and Collab are the most responsive to rewiring. IMDB-B records the largest reductions in all metrics: prevalence by −23.9%, intensity by −38.5%, variability by −70.1%, and extremity by −25.5%. Collab follows closely, ranking second in all metrics. In contrast, Reddit-B resists mitigation the most: it ranks last across all metrics and is the only dataset where rewiring worsens over-squashing . We hypothesize that this is due to disconnected components within each graph, where rewiring inadvertently introduces new bottlenecks. Among the bioinformatic datasets,Mutag exhibits the most consistent responsiveness. Proteins is slightly less responsive, especially in prevalence (−13.2%) and intensity (−19.0%). Enzymes presents the weakest responsiveness among the three, with smaller and second-worst reductions in all four metrics: −9.1% in prevalence, −16.9% in intensity, −20.2% in variability, and −18.5% in extremity .Overall, these results suggest that connected social graphs with dense community structure (Collab and IMDB-B) benefit most from rewiring, while large, disconnected networks such as Reddit-B—and to a lesser extent molecular graphs—pose greater challenges for over-squashing mitigation.[10]

4.3 Results on Node Classification Tasks

We report our results for Q1–Q4 of node classification.

Dataset Over-Squashing Levels (Q1): Table 4 shows that over-squashing is generally weak across datasets and metrics. Cornell, Texas, and Wisconsin display the highest prevalence (0.50–0.55, large), but with low intensity (0.006–0.009), variability (0.005–0.008), and extremity (0.08–0.11), indicating widespread yet mild compression. Chameleon has the highest intensity (0.14, moderate) and extremity (0.39, strong), but with low prevalence (20%, small), suggesting severe, uneven bottlenecks over a small subset of pairs—making it a suitable benchmark for mitigation studies. Cora and Citeseer show the lowest prevalence (0.015 and 0.002, respectively), along with the lowest intensity and variability, indicating minimal over-squashing. Comparing Tables 1 and 4, graph-task datasets are more susceptible to over-squashing than node-task datasets.

Rewiring Effectiveness (Q2): Table 5 shows in node-classification tasks, rewiring more often *increases* over-squashing (red markers, •) than reduces it (green markers, •)—the opposite of the trend observed in graph-classification benchmarks. FoSR generally raises the metrics, with only a few exceptions all of which are weak treatment effects. DIGL also increases over-squashing in five of the six datasets; Wisconsin is the exception with the weak treatment effects. SDRF has a near-zero impact, with changes mostly on the order of 10^{-6} to 10^{-2}). These negligible effects have improved over-squashing just for Wisconsin and Chameleon. BORF exhibits mixed behavior, improving over-squashing for some metrics in some datasets while worsening others.

By treatment-effect ranking, DIGL performs the worst in most datasets/metrics—except in Wisconsin, where it ranks best across all metrics. BORF ranks best for intensity, variability, and extremity in Citeseer, Texas, and Chameleon, while FoSR is best for these metrics in Cora and Cornell. In Wisconsin, FoSR is the worst overall metric.

Aggregating effects across datasets (AVG ITE in Table 5), DIGL has the strongest adverse effects (prevalence +0.15, intensity +0.1, variability +0.09, and extremity +1.1 on average), being worst in all metrics except variability. SDRF has a near-negligible impact: it slightly increases all four metrics (worsens over-squashing), yet its increments in intensity, variability, and extremity are the smallest among the other methods, making it the "least harmful" of the rewiring options. BORF shows a mixed pattern: it slightly lowers prevalence (-0.011, the best among all methods) but sharply increases extremity (+0.37) and variability (+0.11, the worst), indicating reduced global compression at the cost of new local bottlenecks.

Overall, as node-classification benchmarks are structurally less prone to over-squashing (Table 4), aggressive (e.g., DIGL) or even moderate (e.g, SDRF and BORF) rewiring is often ineffective or counterproductive. While added connectivity relieves bottlenecks in graph-classification benchmarks with high over-squashing, it

[9] Previous work has also linked DIGL to over-smoothing [11, 20].
[10] See the number of connected components of each dataset in Table 1, which supports this argument.

Table 5: Treatment effects (ITEs) for node classification, averaged over each GNN baseline's hyperparameter configuration. For each dataset–metric combination, the background highlights the best and worst rewiring method. Desirable negative ITEs are marked with •, and undesirable positive ITEs with •. Gain is the percentage change in classification accuracy after rewiring. † marks not statistically significant results. "Avg ITE" summarizes each method's average effect across all datasets.

Rew.	Dataset	Individual Treatment Effect				Gain (%)
		Prevalence	Intensity	Variability	Extremity	
FoSR	Cora	6.0e-4 ± 2.2e-2 •	−1.0e-2 ± 3.7e-2 •	−4.5e-3 ± 2.8e-2 •	−6.1e-2 ± 1.9e-1 •	−0.9 ± 0.1
	Citeseer	−8.1e-5 ± 3.0e-5 •	9.4e-4 ± 4.9e-4 •	3.2e-3 ± 1.2e-3 •	2.0e-2 ± 4.5e-3 •	1.2 ± 1.7
	Texas	7.5e-2 ± 5.0e-2 •	1.6e-2 ± 1.3e-3 •	1.2e-2 ± 9.9e-4 •	9.1e-2 ± 5.1e-2 •	−2.3 ± 5.9
	Cornell	4.5e-2 ± 6.3e-2† •	2.3e-2 ± 4.4e-3 •	1.9e-2 ± 9.7e-2 •	1.4e-1 ± 9.4e-2 •	−1.1 ± 0.3
	Wiscon.	4.4e-2 ± 5.1e-2 •	1.3e-2 ± 1.9e-2 •	1.1e-2 ± 1.7e-2 •	1.4e-1 ± 1.7e-1 •	1.9 ± 2.6
	Chamel.	1.8e-1 ± 1.0e-2 •	8.5e-3 ± 3.8e-3 •	2.8e-2 ± 3.5e-4 •	1.1e-1 ± 1.6e-2 •	−0.9 ± 1.2
	Avg ITE	5.7e-2 ± 6.6e-2 •	8.6e-3 ± 1.1e-2 •	1.1e-2 ± 1.1e-2 •	7.3e-2 ± 7.9e-2 •	−0.4 ± 1.4
DIGL	Cora	6.4e-1 ± 0.0e0 •	2.0e-1 ± 0.0e0 •	1.6e-1 ± 0.0e0 •	2.2e0 ± 0.0e0 •	1.3 ± 0.0
	Citeseer	1.8e-1 ± 0.0e0 •	1.4e-1 ± 0.0e0 •	1.5e-2 ± 0.0e0 •	1.3e0 ± 0.0e0 •	1.0 ± 0.0
	Texas	3.3e-2 ± 0.0e0 •	2.4e-2 ± 0.0e0 •	3.8e-2 ± 0.0e0 •	3.0e-1 ± 0.0e0 •	−0.8 ± 0.0
	Cornell	−3.0e-2 ± 0.0e0 •	2.0e-1 ± 0.0e0 •	2.1e-1 ± 0.0e0 •	1.7e0 ± 0.0e0 •	5.0 ± 0.0
	Wiscon.	−4.0e-1 ± 0.0e0 •	−7.4e-3 ± 0.0e0 •	−7.4e-3 ± 0.0e0 •	−1.0e-1 ± 0.0e0 •	−2.4 ± 0.0
	Chamel.	4.6e-1 ± 0.0e0 •	5.2e-2 ± 0.0e0 •	1.1e-1 ± 0.0e0 •	1.1e0 ± 0.0e0 •	−0.7 ± 0.0
	Avg ITE	1.5e-1 ± 3.7e-1 •	1.0e-1 ± 9.1e-2 •	8.8e-2 ± 8.6e-2 •	1.1e0 ± 7.8e-1 •	0.6 ± 2.3
SDRF	Cora	5.5e-6 ± 0.0e0 •	5.3e-5 ± 0.0e0 •	3.2e-6 ± 0.0e0 •	1.9e-5 ± 0.0e0 •	0.3 ± 0.9
	Citeseer*	N/A	N/A	N/A	N/A	N/A
	Texas	1.2e-1 ± 0.0e0 •	2.2e-3 ± 0.0e0 •	5.3e-3 ± 0.0e0 •	8.7e-2 ± 0.0e0 •	−1.8 ± 2.1
	Cornell*	N/A	N/A	N/A	N/A	N/A
	Wiscon.	−5.2e-2 ± 1.2e-1 •	−1.2e-3 ± 2.9e-3 •	−5.1e-4 ± 2.2e-3 •	6.8e-3 ± 4.0e-2 •	0.4 ± 0.4
	Chamel.	−2.7e-4 ± 1.2e-5 •	−2.0e-4 ± 9.7e-5 •	−1.9e-4 ± 1.6e-4 •	5.8e-4 ± 9.4e-5 •	0.2 ± 0.1
	Avg ITE	1.7e-2 ± 6.3e-2 •	2.1e-4 ± 1.2e-3 •	1.1e-3 ± 2.8e-3 •	2.4e-2 ± 3.7e-2 •	−0.2 ± 0.9
BORF	Cora	−6.8e-5 ± 1.4e-6 •	−8.0e-4 ± 1.0e-4 •	1.7e-4 ± 2.7e-5 •	1.2e-2 ± 2.1e-4 •	2.3 ± 2.1
	Citeseer	2.0e-6 ± 9.9e-7† •	−3.3e-6 ± 2.3e-6† •	−9.2e-6 ± 6.4e-6 •	0.0e0 ± 0.0e0 •	2.7 ± 1.6
	Texas	3.7e-2 ± 6.8e-2 •	7.6e-4 ± 8.7e-4 •	8.4e-4 ± 1.5e-3 •	2.6e-2 ± 4.7e-2 •	7.4 ± 3.1
	Cornell	−6.0e-2 ± 2.7e-2 •	6.8e-2 ± 1.5e-2 •	6.9e-1 ± 3.9e-1 •	2.3e0 ± 7.8e-2 •	10.7 ± 2.0
	Wiscon.	−3.1e-2 ± 1.1e-2 •	3.3e-4 ± 1.0e-4 •	−9.5e-4 ± 5.4e-4 •	−9.9e-3 ± 2.9e-3 •	6.1 ± 0.5
	Chamel.	−8.2e-3 ± 1.4e-5 •	−3.3e-2 ± 5.7e-4 •	−2.1e-2 ± 2.8e-4 •	−9.9e-2 ± 1.5e-3 •	4.8 ± 3.5
	Avg ITE	−1.1e-2 ± 3.4e-2 •	5.8e-3 ± 3.0e-2 •	1.1e-1 ± 2.8e-1 •	3.7e-1 ± 8.6e-1 •	5.7 ± 2.9

* No edges are added by SDRF on Citeseer and Cornell; consequently, no treatment effect can be computed (entries marked "N/A"). The performance change reported in the SDRF paper stems from a different hyperparameter set rather than the rewiring itself.

often disrupts local structure in node-classification benchmarks with low over-squashing, creating new compression pathways.

Rewiring vs. Performance (Q3): Figure 1b shows the correlation coefficient ρ between each rewiring method's treatment effect and its performance changes. DIGL shows strong, significant positive correlations for three metrics ($\rho \geq 0.83$), suggesting its performance gains coincide with increased over-squashing. SDRF shows moderate, non-significant negative correlations ($\rho = -0.37$); while in the "right" direction, effects are too small to yield meaningful gains. FoSR exhibits weak, non-significant negative correlations for most metrics, indicating little potential performance gain. BORF mostly shows non-significant positive correlations, implying its mitigation may drop the performance. Overall, in node-classification benchmarks, rewiring rarely improves performance by reducing over-squashing. On the contrary, performance gains—particularly in DIGL—often coincide with increased compression, suggesting that other mechanisms (e.g., altered propagation patterns or smoothing behavior) drive the improvements.

Dataset Treatment Responsiveness (Q4): Table 6 shows that no dataset exhibit "true" responsiveness to over-squashing mitigation. All values (except one) are positive, indicating that rewiring methods fail to reduce over-squashing and often worsen it. Citeseer is the most extreme case,suggests followed by Cornell. These results suggest that rewiring often introduces new bottlenecks in node classification tasks, rather than relieving them.

4.4 Discussion: Graph vs. Node Classification

Dataset Over-Squashing Levels (Q1): Graph-classification datasets exhibit substantially higher levels of over-squashing than node-classification ones (compare Tables 1 and 4). Hence, over-squashing

Table 6: Responsiveness Node datasets to rewiring (percentages). Negatives are desirable mitigation (metric decreases), whereas positives indicate increases. Text color denote most, second-most, second-worst, and worst responsiveness.

Dataset	Prevalence	Intensity	Variability	Extremity
Cornell	2.7	1079.0	3865.3	1228.1
Texas	13.1	177.4	244.8	161.7
Wisconsin	−20.1	15.1	6.6	8.4
Cora	1052.6	129.5	150.6	264.7
Citeseer	3260.9	15210.4	346.6	2391.3
Chameleon	78.8	4.8	34.6	71.1

is a core obstacle in graph classification datasets, but a more negligible issue in node datasets—implying that mitigation efforts may be more impactful for the former.

Rewiring Effectiveness (Q2): Rewiring mitigates over-squashing in graph benchmarks, but is often harmful in node datasets. In graph datasets, added connectivity—especially via DIGL's dense rewiring—consistently reduces over-squashing metrics. In node datasets, the same interventions frequently worsen over-squashing, as already balanced or mildly compressed structures (i.e., graphs with minimal over-squashing) are disrupted, and new bottlenecks emerge. This suggests that rewiring is beneficial only when over-squashing is severe or moderate (e.g., graph datasets) and can be counterproductive when over-squashing is low (most node datasets).

Rewiring vs. Performance (Q3): In graph-level benchmarks, FoSR and BORF show negative correlations between reduced over-squashing and improved accuracy, confirming that alleviating information bottlenecks enhances generalization. DIGL, despite large reductions, fails to improve performance—likely due to aggressive edge additions erasing topological information. In node-level tasks, DIGL and BORF often improve performance while surprisingly increasing over-squashing, suggesting that other factors (e.g., smoothing or altered message propagation) drive the gains. SDRF and FoSR show no correlations. Overall, rewiring mitigation helps performance when over-squashing is pronounced (as in most graph datasets) and not overcorrecting (as in FoSR or BORF). By contrast, when over-squashing is mild—as in typical node datasets—rewiring rarely converts metric improvements into accuracy gains.

Dataset Responsiveness (Q4): Dataset responsiveness to rewiring is pronounced in graph classification datasets but not in node classification datasets. In graph datasets, well-connected social networks (i.e., Collab and IMDB-B) show the strongest responsiveness, and bioinformatic graphs are moderately responsive. In node datasets, rewiring rarely helps and often hurts. These findings suggest that rewiring pays off only when over-squashing is severe and global, but has limited or negative impact when compression is mild, underscoring the need for dataset-aware interventions.

5 Conclusion and Future Work

We proposed topology-focused measurements for over-squashing, built on its formal characterization by modeling the exponential decay of node-pair sensitivity with increasing network depth. We extend our measurements to the graph-level measures and integrate them into a causal inference framework to evaluate the effect of rewiring on over-squashing. Our extensive empirical analyses show that graph-classification datasets (except Reddit-B) suffer substantial over-squashing and are generally responsive: rewiring lowers our metrics and often boosts accuracy. Node-classification benchmarks show little over-squashing; rewiring often *increases* compression, and its performance effects are largely unrelated to over-squashing. Our findings underscore the importance of applying rewiring selectively, based on the presence of over-squashing. Future work includes extending our experiments to dynamic rewiring methods, exploring the relationship between negative decay rates and over-smoothing, and designing novel rewiring methods guided by our over-squashing measurement framework.

A Normalized Jacobian Norm Approximation

Proof. We derive an approximation of the relative Jacobian norm $\tilde{\mathcal{J}}_\ell(v,u)$ under the assumption of a linear message-passing GNN. Let the ℓ-th layer of a *linear* message-passing GNN be

$$\mathbf{H}^{(\ell)} = \tilde{\mathbf{A}}\mathbf{H}^{(\ell-1)}\mathbf{W}^{\ell} \tag{9}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the self-loop–augmented adjacency matrix, $\mathbf{H}^{(\ell-1)}$ stacks node embeddings of the layer $\ell - 1$ as rows, and $\mathbf{W}^{\ell}$ is the learnable weight matrix of the ℓ-th layer. Iterating from the initial features $\mathbf{H}^0$ gives

$$\mathbf{H}^{(\ell)} = \tilde{\mathbf{A}}^{\ell}\mathbf{H}^{(0)}\mathbf{W}, \qquad \mathbf{W} := \mathbf{W}^{(1)}\mathbf{W}^{(2)}\dots\mathbf{W}^{(\ell)}. \tag{10}$$

For any node v, its representation after ℓ layers is:

$$\mathbf{h}_v^{(\ell)} = \sum_{u=1}^{n} (\tilde{\mathbf{A}}^{\ell})_{uv}\mathbf{h}_u^{(0)}\mathbf{W}, \tag{11}$$

where $\mathbf{h}_u^{(0)}$ is the input feature row of node u. The Jacobian of $\mathbf{h}_v^{(\ell)}$ with respect to $\mathbf{h}_u^{(0)}$ is then

$$\frac{\partial \mathbf{h}_v^{(\ell)}}{\partial \mathbf{h}_u^{(0)}} = (\tilde{\mathbf{A}}^{\ell})_{uv}\mathbf{W} \in \mathbb{R}^{d_0 \times d_\ell}, \tag{12}$$

Given that the matrix norms are homogeneous—for any scaler c, $\|c\,\mathbf{W}\| = |c|\|\mathbf{W}\|$—we compute the Frobenius norm

$$\left\| \frac{\partial \mathbf{h}_v^{(\ell)}}{\partial \mathbf{h}_u^{(0)}} \right\| = \|(\tilde{\mathbf{A}}^{\ell})_{uv}\mathbf{W}\| = (\tilde{\mathbf{A}}^{\ell})_{uv}\|\mathbf{W}\|. \tag{13}$$

To compute the relative Jacobian norm, we normalize by the total sensitivity of v to all input nodes:

$$\sum_k \left\| \frac{\partial \mathbf{h}_v^{(\ell)}}{\partial \mathbf{h}_k^{(0)}} \right\| = \|\mathbf{W}\| \sum_k (\tilde{\mathbf{A}}^{\ell})_{kv}. \tag{14}$$

Canceling the common factor $\|\mathbf{W}\|$, the relative Jacobian norm simplifies to the exact expression:

$$\tilde{\mathcal{J}}_\ell(v,u) = \frac{(\tilde{\mathbf{A}}^{\ell})_{uv}}{\sum_k (\tilde{\mathbf{A}}^{\ell})_{kv}}. \tag{15}$$

□

GenAI Usage Disclosure

We used Generative AI tools in a limited capacity to support this work. During the writing process, GenAI was employed occasionally to improve the clarity and readability of the text, such as for grammar checking and suggesting alternative word choices. In terms of coding, GenAI tools were used primarily to assist with writing documentation and resolving debugging issues. All core research contributions, including the development of the methodology, design of experiments, analysis of results, and interpretation of findings, were conceived and executed independently by the authors without the involvement of GenAI systems.

References

[1] Ralph Abboud, Radoslav Dimitrov, and Ismail Ilkan Ceylan. 2022. Shortest Path Networks for Graph Property Prediction. In *Learning on Graphs Conference*. 5–1.

[2] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *International Conference on Machine Learning*. 21–29.

[3] Uri Alon and Eran Yahav. 2021. On the Bottleneck of Graph Neural Networks and Its Practical Implications. In *International Conference on Learning Representations*.

[4] Adrián Arnaiz-Rodríguez, Ahmed Begga, Francisco Escolano, and Nuria Oliver. 2022. DiffWire: Inductive Graph Rewiring via the Lov 'asz Bound. *arXiv preprint arXiv:2206.07369* (2022).

[5] Federico Barbero, Ameya Velingker, Amin Saberi, Michael Bronstein, and Francesco Di Giovanni. 2023. Locality-Aware Graph-Rewiring in GNNs. *arXiv preprint arXiv:2310.01668* (2023).

[6] Pablo Barceló, Egor V Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan-Pablo Silva. 2020. The Logical Expressiveness of Graph Neural Networks. In *8th International Conference on Learning Representations*.

[7] Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. 2023. Understanding Oversquashing in GNNs Through the Lens of Effective Resistance. In *International Conference on Machine Learning*. 2528–2547.

[8] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. 2021. Weisfeiler and Lehman Go Cellular: CW Networks. *Advances in Neural Information Processing Systems* 34 (2021), 2625–2640.

[9] Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. 2021. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *International Conference on Machine Learning*. 1026–1037.

[10] Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. 2023. On the Connection Between MPNN and Graph Transformer. In *International Conference on Machine Learning*. 3408–3430.

[11] Jeongwhan Choi, Sumin Park, Hyowon Wi, Sung-Bae Cho, and Noseong Park. 2024. PANDA: Expanded Width-Aware Message Passing Beyond Rewiring. *arXiv preprint arXiv:2406.03671* (2024).

[12] Jacob Cohen. 2013. *Statistical Power Analysis for the Behavioral Sciences*. Routledge.

[13] Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M Bronstein. 2023. On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology. In *International Conference on Machine Learning*. 7865–7885.

[14] Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. 2020. SIGN: Scalable Inception Graph Neural Networks. *arXiv preprint arXiv:2004.11198* (2020).

[15] Rickard Brüel Gabrielsson, Mikhail Yurochkin, and Justin Solomon. 2022. Rewiring with Positional Encodings for GNNs. (2022).

[16] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion Improves Graph Learning. *Advances in Neural Information Processing Systems* 32 (2019).

[17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*. 1263–1272.

[18] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A New Model for Learning in Graph Domains. In *Proceedings of IEEE International Joint Conference on Neural Networks.*, Vol. 2. 729–734.

[19] Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. 2023. DREW: Dynamically Rewired Message Passing with Delay. In *International Conference on Machine Learning*. 12252–12267.

[20] Kedar Karhadkar, Pradeep Kr Banerjee, and Guido Montúfar. 2022. FoSR: First-Order Spectral Rewiring for Addressing Oversquashing in GNNs. *arXiv preprint arXiv:2210.11790* (2022).

[21] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. 2021. Rethinking Graph Transformers with Spectral Attention. *Advances in Neural Information Processing Systems* 34 (2021), 21618–21629.

[22] Paul Louis, Shweta Ann Jacob, and Amirali Salehi-Abari. 2023. Simplifying Subgraph Representation Learning for Scalable Link Prediction. *arXiv preprint arXiv:2301.12562* (2023).

[23] Quinn McNemar. 1947. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika* 12, 2 (1947), 153–157.

[24] Alessio Micheli. 2009. Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Transactions on Neural Networks* 20, 3 (2009), 498–511.

[25] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. TUDataset: A Collection of Benchmark Datasets for Learning with Graphs. *arXiv preprint arXiv:2007.08663* (2020).

[26] Khang Nguyen, Nong Minh Hieu, Vinh Duc Nguyen, Nhat Ho, Stanley Osher, and Tan Minh Nguyen. 2023. Revisiting Over-Smoothing and Over-Squashing Using Ollivier–Ricci Curvature. In *International Conference on Machine Learning*. 25956–25979.

[27] Giannis Nikolentzos, George Dasoulas, and Michalis Vazirgiannis. 2020. K-Hop Graph Neural Networks. *Neural Networks* 130 (2020), 195–205.

[28] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems* 35 (2022), 14501–14515.

[29] Danial Saber and Amirali Salehi-Abari. 2024. Scalable Expressiveness Through Preprocessed Graph Perturbations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 4020–4025.

[30] Danial Saber and Amirali Salehi-Abari. 2025. Over-Squashing in GNNs and Causal Inference of Rewiring Strategies. *arXiv preprint arXiv:2508.09265* (2025).

[31] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[32] Joshua Southern, Francesco Di Giovanni, Michael Bronstein, and Johannes F Lutzeyer. 2024. Understanding Virtual Nodes: Oversmoothing, Oversquashing, and Node Heterogeneity. *arXiv preprint arXiv:2405.13526* (2024).

[33] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. 2021. Understanding Over-Squashing and Bottlenecks on Graphs via Curvature. *arXiv preprint arXiv:2111.14522* (2021).

[34] Guangtao Wang, Rex Ying, Jing Huang, and Jure Leskovec. 2020. Multi-Hop Attention Graph Neural Network. *arXiv preprint arXiv:2009.14332* (2020).

[35] Eric W Weisstein. 2004. Bonferroni Correction. (2004).

[36] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying Graph Convolutional Networks. In *International Conference on Machine Learning*. 6861–6871.

[37] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *International Conference on Machine Learning*. 5453–5462.

[38] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation? *Advances in Neural Information Processing Systems* 34 (2021), 28877–28888.