Analysis and Discussion on Over-Squashing in GNNs: The Impact of Width, Depth, and Topology

Itaï Zehavi
ENS Paris Saclay - DER SIEN
itai.zehavi@ens-paris-saclay.fr

Elliot Merle
ENS Paris Saclay - DER Biologie
elliot.merle@ens-paris-saclay.fr

Abstract

We study over-squashing in Message Passing Neural Networks (MPNNs) by building on and extending the theoretical analysis of [3], who relate sensitivity bounds to model width, depth and graph topology via commute time and Cheeger constants. While their work focuses on how topology limits information diffusion, we adopt a complementary perspective and ask how topology also constrains the expressivity of Graph Neural Networks. To this end, we consider a controlled synthetic setting in which a Graph Attention Network (GATv2) [2] sits at the core of an autoencoder and communicates through a star-shaped graph with a variable number of central nodes. Our experiments show that the number of central nodes effectively bounds the dimensionality of the information subspace that can be faithfully reconstructed: as the bottleneck widens, reconstruction error systematically decreases and more target signals are individually recovered. This provides a concrete, data-driven illustration that graph topology not only governs how far information can propagate, but also how rich the representations can be, highlighting an expressivity bottleneck that complements the classical diffusion-based view of over-squashing.

Keywords

Graph Neural Networks, Message Passing, Over-squashing, Expressivity, Graph Topology, Sensitivity Analysis

ACM Reference Format:
Itaï Zehavi and Elliot Merle. 2025. Analysis and Discussion on Over-Squashing in GNNs: The Impact of Width, Depth, and Topology . In *Proceedings of Geometric Data Analysis (Geometric data analysis).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Context

Among the many data acquired in recent years, a significant part of them is graph-structured. This is useful for many application fields such as social media, electrical networks, meteorology, chemistry, or biology. To process such data, Graph Neural Networks (GNNs) [8] have emerged as the dominant paradigm for learning graph-structured information in the growing fields of artificial intelligence.

Geometric data analysis, Paris, France

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

At their core, message-passing neural networks (MPNNs) [5] propagate information through iterative neighbourhood aggregation, enabling the exploitation of the relational structure of graphs.

In particular, Graph Attention Networks (GATs) [6] introduced an attention mechanism into the message-passing process, allowing nodes to learn the relative importance of their neighbours during aggregation. Instead of treating all adjacent nodes equally, GATs compute attention coefficients that weigh each neighbour's contribution dynamically. This enables the model to focus on the most relevant parts of the local neighbourhood, improving both performance and interpretability.

However, despite their empirical success, GNNs face many theoretical and practical limitations that have motivated extensive research in recent years [3]. A foundational line of work has established that the expressive power of standard MPNNs is theoretically bounded by the 1-dimensional Weisfeiler–Leman graph isomorphism test [4]. This means that classic MPNNs cannot capture certain non-isomorphic graph structures and complex topological patterns. To this extent, a lot of effort has been devoted to creating more expressive MPNNs that extend beyond the 1-dimensional Weisfeiler–Leman test.

Another critical limitation, independent of expressivity, has been identified as the over-squashing phenomenon, where information from distant nodes is compressed when propagating through topological bottlenecks.

The article under consideration contributes to this landscape by providing a rigorous theoretical approach to understand and prevent the over-squashing phenomenon through the lens of sensitivity bounds, incorporating width, depth, and topological factors via commute time and Cheeger constants. Importantly, it extends beyond standard over-squashing analysis [1] by suggesting a novel perspective on expressivity bottlenecks, demonstrating that topology can fundamentally limit the capacity of GNNs and the MPNN algorithm. This view complements the previous literature and suggests that effective GNN design requires jointly considering the graph structure, the model architecture, and the intrinsic dimensionality of node features.

2 Main content of the article

As we previously mentioned, MPNNs are the core function of GNNs that allows them to use graph-structured data. Theoretically, they operate recursively as follows: for each layer $t = 1, \dots, m$, representations are updated such that

$$h_v^t = \mathrm{comb}_t\left(h_v^{t-1}, \mathrm{agg}_t\left(\{h_u^{t-1} \mid u \in \mathcal{N}(v)\}\right)\right), \tag{1}$$

where $\mathcal{N}(v)$ is the neighbourhood of node v, agg_t is an aggregation function (mostly linear), and comb_t is a non-linear combination function. In a simpler way, each node at each layer aggregates the

information of its neighbours from the previous layer together with its own previous information.

We can easily identify how the over-squashing phenomenon arises. Indeed, when information originates from one end of a graph, at each layer this information is combined with the current representation of the nodes it passes through. This means that when a topological bottleneck emerges (topologically, a region where information must funnel through a limited number of nodes), all subsequent node representations depend on the intermediary bottleneck nodes. Furthermore, because these nodes cannot fully encode all the information they receive, information is inevitably lost as the signal propagates through the bottleneck.

2.1 Identifying the over-squashing phenomenon

In order to understand the over-squashing phenomenon, the authors express it mathematically using the GNN and MPNN quantities [3]. Over-squashing can be indirectly measured by defining the sensitivity of each node v at layer m with respect to a distant node u at layer 0:

$$\left\| \frac{\partial h_v^m}{\partial h_u^0} \right\|. \tag{2}$$

Notation.

- c is the Lipschitz constant of the network's non-linearity;
- w bounds the model weights;
- p is the width (channel dimension) of the network;
- m is the number of message-passing layers;
- $\left(S_{r,a}\right)^m [v,u]$ is the (v,u) element of the matrix $S_{r,a}^m$ quantifying information flow after m layers;
- $h_v^{(m)}$ is the representation of node v at layer m;
- $r = d(v,u)$ is the geodesic distance between nodes u and v;
- $r^{v,u}$ is the number of shortest paths from u to v;
- $d_{\min}$ is the minimum degree in the graph;
- c_r, c_a are aggregation coefficients;
- $\kappa_{u,v}$ is the commute time between u and v;
- h_{Cheeger} is the Cheeger constant of the graph (the ratio between the number of edges crossing the sparsest cut and the size of the smaller partition);
- $\mathcal{L}$ is the loss function;
- $W^{(k)}$ are the parameters at layer k;
- $O_m(v,u)$ measures the obstruction to information propagation between v and u after m message-passing steps.

This table 1 summarizes the main results of the authors' work. The definition of sensitivity measures how node u is impacted by a small perturbation at the beginning of the message-passing process. A large Jacobian means the nodes are well-connected informationally; a small one means the perturbation is lost along the way. Over-squashing occurs precisely when this Jacobian becomes very small, distant nodes barely influence each other.

The authors demonstrate that sensitivity is bounded by parameters of the model such as width (Theorem 3.2), depth (Theorems 4.1 and 4.2), and also the topological structure of the graph represented by access time (expected steps of a random walk from u to v; Theorem 5.3) and commute time (expected time to go from u to v and back with a random walk; Theorem 5.5).

Table 1: Summary of key quantities in over-squashing analysis

Type of Quantity	Formula
Sensitivity	$\left\| \frac{\partial h_v^m}{\partial h_u^0} \right\|$
Sensitivity with width p (Theorem 3.2)	$\left\| \frac{\partial h_v^m}{\partial h_u^0} \right\| \leq c\, w\, p \times \left\| \left(S_{r,a}\right)^m [v,u] \right\|$
Sensitivity with depth m, $m \sim r \sim diameter$ (Theorem 4.1)	$\left\| \left(S_{r,a}\right)^m [v,u] \right\| \leq C\, r^{v,u} \times \left(\frac{2cwp}{d_{\min}} \right)^r$
Sensitivity with depth m, $m \gg diameter$ (Theorem 4.2)	$\left\| \frac{\partial \mathcal{L}}{\partial W^{(k)}} \right\| \leq C\, (c_r c_a)^m$
Obstruction Jacobian via commute time (Theorem 5.3)	$O_m(v,u) \leq \frac{4c_a}{(1-c_a)^2} \times \kappa_{u,v}$
Obstruction Jacobian via access time (Theorem 5.5)	$O^{(m)}(v,u) \geq \frac{\rho}{vc_a} \frac{t(u,v)}{2\lvert E \rvert} + o(m)$

We can deduce from this work that several factors control how information propagates. Constants c and w depend on the model, while $(S_{r,a})^m[v,u]$ is the topological factor describing how much information the graph's structure allows for flow after m steps. Understanding the graph topology is therefore crucial. The authors also show that depth influences information flow: when m is close to the diameter, the capacity to transmit information decreases exponentially with distance r. When $m \gg diameter$, the vanishing gradient phenomenon appears, meaning early layers cannot learn. This explains why standard GNNs rarely exceed 5–8 layers in practice.

To avoid over-squashing, the authors suggest graph rewiring techniques as a solution. These methods modify the original graph structure to improve information flow and address fundamental limitations of message-passing neural networks. Di Giovanni et al. [3] implement two types of rewiring: spatial and spectral. Spatial rewiring enhances connectivity by adding edges between distant nodes or introducing higher-order structures to reduce graph diameter. Spectral rewiring leverages spectral properties such as the Cheeger constant to optimize connectivity and reduce obstructions such as commute time. Both strategies mitigate the compression of distant information into fixed-size embeddings, improving sensitivity to long-range dependencies. However, it is important to preserve the meaningfulness of the original graph structure.

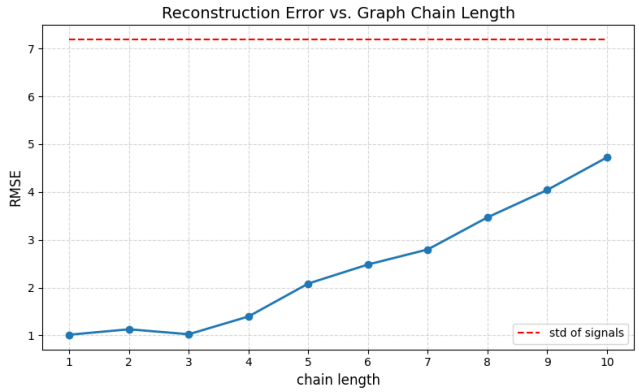

Figure 1: Evolution of the signal reconstruction error as the length of the graph chain increases.

All these conclusions confirm that the topological structure of the graph limits the capacity of the model through the over-squashing problem. To further illustrate the effect of graph topology, the authors define the obstruction Jacobian $O_m(v, u)$, which quantifies the bottleneck preventing information from flowing between u and v. Theorems 5.3 and 5.5 link commute time and access time to this obstruction measure. The results show that if two nodes have a long commute time (few indirect paths), the graph structure isolates them and suffers from over-squashing. On the other hand, small commute times indicate good connectivity. Similarly, large access time means information is difficult to propagate. These conclusions also relate to the Cheeger constant, which measures the presence of topological bottlenecks.

3 Limitations

While the authors address in a new way the problem of over-squashing in graph neural network, Di Gioanni et al. 's [3] work present several key limitations. Even if some progress have been made to avoid the over-squashing phenomenon, the problem of the theoretical limited expressive power of MPNN is still an issue. MPNNs are therotically boudned by the 1-dimensional WL graph isomorphism test [7], as we explained previously, meaning they cannot distinguish or capture certain non-isomorphic graph structures and complex topological patterns, which restricts their ability to fully represent these data. The solution of rewiring techniques presented by the authors only sweep under the rug the theoretical issue.

The easiest solution to implement, such as increasing the hidden size of models to alleviates over-impact severely impact the capacities of generalization and computational efficiency.

Some attention-based models like GATs can partially alleviate over-squashing by gating or pruning edges. However, the attention mechanism may fail if it cannot adequately identify which message-passing path to emphasize or ignore.

Finally, while graph rewiring methods have been shown to improve the over-squashing by modifying graph connectivity, the current theoretical framework does not specify when or why one rewiring approach is preferable over another. Furthermore, the relation between over-squashing and other phenomenons like over-smoothing remains an open-field investigation.

Following Di Giovanni team's analysis of the limitations imposed by graph topology on information diffusion and over-squashing in MPNNs, we investigate a complementary angle by directly exploring how topology affects the expressivity of GNMs. While prior work established that topological bottlenecks restrict sensitivity and the effective propagation of information, the current experiment highlights how these bottleneck also constrain the dimensionality of information that can be represented and reconstructed.

We study an autoencoder architecture where a GATv2 mediates communication through a star-shaped graph with a variable number of central nodes. Our results confirm that the number of central nodes bounds the maximum subspace dimension of the transmitted representations, which directly limits expressivity. As the number of central nodes grows, reconstruction errors declines systematically. We can hope to offer a more concrete and data-driven perspective on the theoretical observations noted regarding topological limitation.

4 Topology also affects the model's expressivity

4.1 Expressivity issue and experiment overview

In the original paper, the author aims to derive bounds on the capacity of information diffusion between two nodes, which depend on certain topological characteristics of the graph.

However, studying information diffusion in isolation can quickly lose its relevance, as it is often disconnected from the data on which it operates. In some cases, it is even desirable that certain nodes communicate very little with others, and vice versa, this strongly depends on the nature of the data and the task at hand.

Here, we propose a new line of investigation, this time focusing on the data themselves. We can observe that the topology of a graph can directly influence the expressivity of a node. For simplicity, we place ourselves in the framework of a Graph Attention Network (GATv2) [2]. To highlight this phenomenon, we designed the following experiment: first, we build a standard autoencoder that compresses and decompresses simple signals (trend + seasonality). The only difference is that, at its center, the information transmission block is replaced by a GATv2.

In a GATv2 [2] layer, node i updates its representation by attending to its neighbours $\mathcal{N}(i)$. Given input features h_i, the layer computes queries, keys, and values

$$q_i = \mathbf{W}_Q h_i, \qquad k_j = \mathbf{W}_K h_j, \qquad v_j = \mathbf{W}_V h_j.$$

The attention score from i to j is

$$e_{ij} = \mathrm{LeakyReLU}\big(a^\top [\, q_i \| k_j \,]\big),$$

normalized as

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{p \in \mathcal{N}(i)} \exp(e_{ip})}.$$

The new node representation is then

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}\, v_j \right).$$

GATv2 improves over the original GAT by making the attention fully learnable in both h_i and h_j, thus increasing expressivity [2].

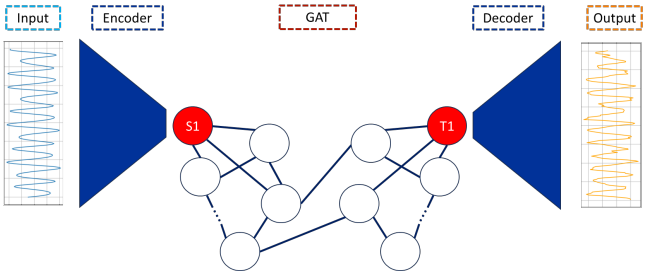

Figure 2: General scheme of the experiment

In our experiment, we replace this arbitrary graph with a star graph, in which we gradually increase the number of central nodes. The goal is to show that as this number grows, the reconstructed output signals become more expressive.

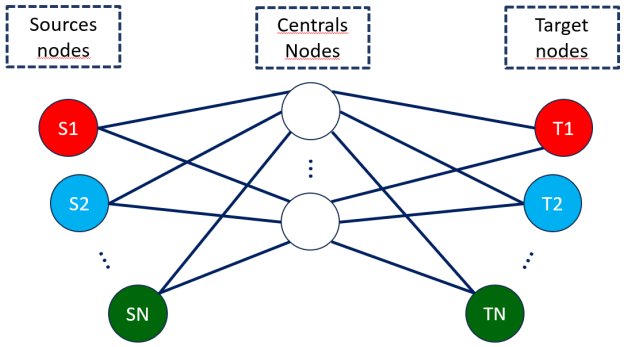

Figure 3: Star graph with a variable number of central nodes

The goal of this experiment is to observe how the topology of the graph, in particular, the number of central nodes in a star structure, affects the expressivity of a GATv2 placed at the core of an autoencoder.

Overall graph structure. We consider a fixed set of five source nodes and five target nodes. Each source node is paired with a unique target node, which must reconstruct its corresponding signal after it has travelled through the GATv2. Source and target nodes are not directly connected: they are linked only through the central nodes. The resulting graph is an *extended star*, where we vary the number of central nodes to modulate the severity of the bottleneck.

Data generation. Each source emits a synthetic signal consisting of:

- two cosine components, with randomly sampled frequencies in a continuous range;
- a degree-2 polynomial trend;
- a total length of 200 samples.

For each source node, we generate 10,000 independent signals, yielding 50,000 training examples per model. All signals are regenerated randomly for each run, ensuring that they are almost surely linearly independent.

Model architecture. The autoencoder comprises:

- a two-layer encoder projecting each signal into a 64-dimensional latent space;
- a symmetric two-layer decoder;
- a four-layer, single-head GATv2 responsible for transmitting the latent vector from the source node to its corresponding target node through the central nodes.

Thus, the GATv2 is the sole communication pathway between encoder and decoder.

Training protocol. For each number of central nodes, we train a full model for 500 epochs. Each training run takes approximately 50 minutes. In several cases, multiple attempts were required, as the model could remain trapped in local minima — especially when large batch sizes were used.

Optimization details. We train the model using the Adam optimizer with a learning rate of 5×10^{-4}. The loss function is the **RMSE** on the reconstructed signals. We use a batch size of 16: larger values tended to push the model into persistent local minima. No learning rate scheduler is employed in our setup.

Important detail. An essential but initially non-obvious point is that the nodes must be *explicitly labeled* in our experimental setup. We discovered this when trying to verify whether several source/target pairs could communicate simultaneously in a complete graph: without explicit node labeling, the model was unable to handle multiple pairs at once. The reason is straightforward: since all signals are generated from the same underlying process, the nodes are indistinguishable at the dataset level. As a result, the model cannot infer which node plays which role, and it simply outputs a reconstruction equal to the average of all source signals. To address this issue, we added a single scalar to each node's embedding, equal to its node index. This simple labeling allows the model to distinguish nodes from one another, to identify "who is who," and therefore to preserve the correct source/target associations.

Transition toward the theoretical analysis. Before presenting the full set of experimental results, it is useful to clarify the theoretical mechanism that governs the phenomenon we aim to study. The empirical setup introduced above suggests that the expressivity of the reconstructed signals strongly depends on the structure of the central bottleneck in the star graph. In particular, increasing the number of central nodes appears to give the GATv2 more freedom to transmit and combine latent information.

We develop a theoretical argument showing why this behaviour is expected: the topology of the graph itself imposes hard constraints on the dimensionality of the information that can be propagated through the central nodes. These constraints will allow us to interpret and justify the empirical results, which will be presented in detail in the following section.

Expressivity limitations in a star graph with k central nodes. Consider an *extended star graph* in which a set of source nodes $\{s_1, \dots, s_m\}$ is connected to a set of k central nodes $\{c_1, \dots, c_k\}$, which themselves connect to target nodes $\{t_1, \dots, t_n\}$. There is no direct edge between sources and targets, nor among targets.

Let $h_v^{(0)}$ denote the initial embedding of node v, produced by the encoder. The GATv2 updates embeddings according to:

$$h_v^{(l+1)} = \sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(l)} V^{(l)} h_u^{(l)},$$

where $V^{(l)}$ is a shared projection matrix and $\alpha_{vu}^{(l)}$ is an attention weight.

At the first layer, each central node receives information from all sources:

$$h_{c_i}^{(1)} = \sum_{j=1}^{m} \alpha_{c_i s_j}^{(0)} V^{(0)} h_{s_j}^{(0)}.$$

At the next layer, each target node receives information only from these k central nodes:

$$h_{t_j}^{(2)} = \sum_{i=1}^{k} \alpha_{t_j c_i}^{(1)} V^{(1)} h_{c_i}^{(1)}.$$

Since each $h_{c_i}^{(1)}$ is itself a linear combination of the source embeddings, we obtain:

$$h_{t_j}^{(2)} \in \mathrm{vect}\big(V^{(1)} h_{c_1}^{(1)}, \ldots, V^{(1)} h_{c_k}^{(1)}\big),$$

an at-most k-dimensional subspace.

Therefore, if the target vectors $\{h_{t_1}, \ldots, h_{t_n}\}$ span a space of dimension larger than k, perfect reconstruction is structurally impossible. In other words:

$$\dim\left(\mathrm{vect}(h_{t_1}, \ldots, h_{t_n})\right) \leq \dim\left(\mathrm{vect}(V^{(1)} h_{c_1}^{(1)}, \ldots, V^{(1)} h_{c_k}^{(1)})\right) \leq k.$$

This condition captures a topological expressivity bottleneck: all information must pass through a limited number of intermediate nodes.

In our experiment, the signals are sums of cosines plus a degree-2 polynomial trend. Since the cosine frequencies are sampled continuously within a range, the generated signals are almost always linearly independent. Hence:

$$\dim\left(\mathrm{vect}(h_{t_1}, \ldots, h_{t_n})\right) = n,$$

which implies that perfect reconstruction requires $k \geq n$.

4.2 Experiments

In this section, we present the results of the experiment described earlier. As discussed at the end of the previous subsection, in principle one would need n central nodes to faithfully represent information coming from the n target nodes without loss.

Our goal is to analyze several aspects:

- how the average reconstruction error of the target nodes varies with the number of central nodes;
- the reconstruction of a sample target signal for different numbers of central nodes;
- the evolution of the model's loss during training.

As clearly shown in Figure 4, the reconstruction error decreases as the number of central nodes increases, in line with the theoretical intuition discussed earlier.

During the training of each model, we observed an interesting phenomenon in the evolution of the loss. The loss curve frequently exhibits plateaus followed by sharp drops, as illustrated in Figure 5.

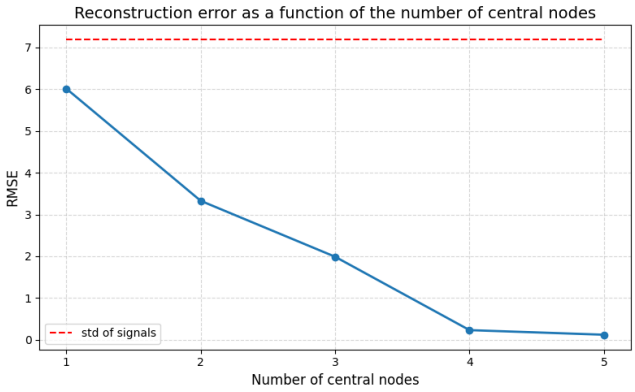

Figure 4: Evolution of the reconstruction error as a function of the number of central nodes.

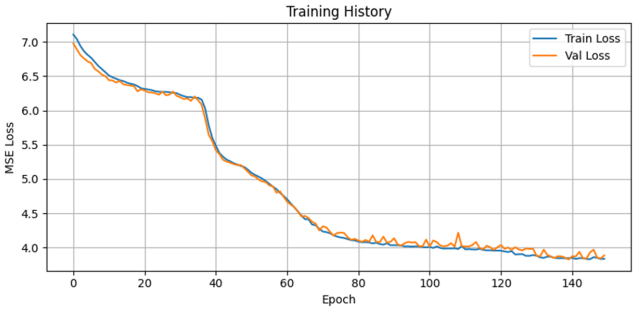

Figure 5: Evolution of the loss during training of the model with 3 central nodes.

In this example, a third drop occurs later in training, but for stability reasons I had to reduce the learning rate midway through training and rerun certain parts several times. What we observe overall is the emergence of several plateaus, approximately around 6.1, 4.8, 3.2, and finally 2.0. Interestingly, some of these plateau values align with those reported in Figure4.

To better understand this behaviour, we now visualize the reconstruction of the target nodes in the case where the graph contains three central nodes. In this configuration, we expect to see three signals reconstructed correctly, and an averaged reconstruction for the remaining two.

The figure clearly shows that three out of the five signals are accurately reconstructed (green arrows), while the remaining two share exactly the same reconstructed signal (red arrows), which corresponds to a kind of averaged shape and therefore a poor individual reconstruction. In the end, the number of correctly reconstructed signals matches the number of central nodes. This behaviour persists when varying the number of central nodes: with two central nodes, two signals are well reconstructed, and so on.

We hypothesize that these two observations are related: each plateau or drop in the loss may correspond to a learning phase during which the model assigns one central node to a specific

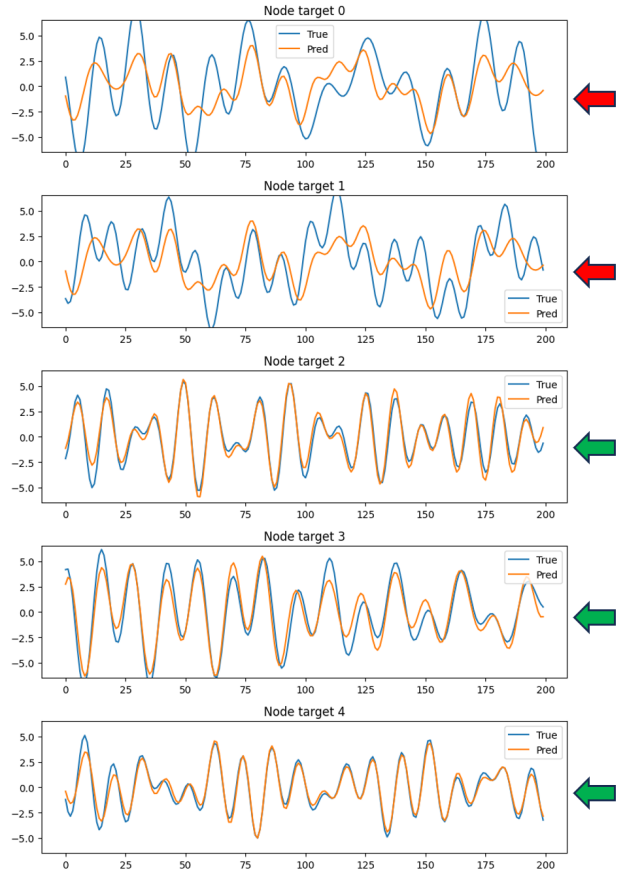

Figure 6: Signal reconstruction for the star graph with 3 central nodes.

source/target pair. Each such assignment would then create a visible break in the loss curve.

Overall, our experimental study strongly confirms the theoretical result: in order to reconstruct all signals, the model requires at least as many central nodes as the intrinsic dimensionality of the information being transmitted.

Limitations of our approach. However, in most practical situations, we do not explicitly know the true dimensionality of the information that nodes must exchange. This is precisely why most existing works focus more on information diffusion induced by the topology than on the expressivity constraints induced by the topology.

In this paper, we aim to show that topology affects not only how information diffuses, but also how expressive the model can be. This provides a complementary perspective on the same underlying problem. A coarse but conceptually simple alternative would be to estimate the dimensionality of the data itself rather than that of the information to be propagated, since propagated information cannot exceed the richness of the input data. However, this approximation is generally too crude: in most real scenarios, the intrinsic dimensionality of the relevant information is far lower than that of the raw input data. If one uses the raw dimensionality as a bound, one concludes that an excessively large number of central nodes is required, far beyond what the task actually needs. In other words, this bound is too loose to be useful in practice, but it highlights the fact that topological expressivity represents an additional effect, distinct from diffusion.

Another limitation is that we illustrated topology-induced expressivity in a deliberately simple setting. Identifying the same phenomenon in more complex (and therefore realistic) graphs is considerably more challenging. Additional work would be needed to generalize these findings and make them applicable to arbitrary graph structures.

5 Conclusion

In this work, we analyzed the over-squashing phenomenon in Message Passing Neural Networks (MPNNs), building on the theoretical framework proposed by Di Giovanni et al. Their results give a precise characterization of how width, depth, and especially graph topology restrict the propagation of information, and how topological bottlenecks fundamentally limit long-range interactions in GNNs. By connecting the obstruction Jacobian to spectral quantities such as commute time and the Cheeger constant, they showed that these limitations are structural, and not simply a consequence of training difficulty. Rewiring techniques offer partial mitigation by reducing graph diameter, but they do not remove the core expressivity constraints of message passing.

Our contribution complements this theoretical perspective by highlighting another effect of topology: beyond limiting information diffusion, topology also constrains how expressive the model can be. Through a controlled synthetic experiment, we illustrated that a narrow topological bottleneck restricts the dimensionality of the information that can be represented and transmitted. In our setting, the number of central nodes directly bounds the number of distinct signals that can be reconstructed, confirming the presence of a topological expressivity bottleneck in addition to the diffusion bottleneck.

However, our approach also has limitations. In practical applications, the true dimensionality of the information exchanged between nodes is rarely known, which makes it difficult to predict how many "channels" the topology should provide. Using the raw dimensionality of the data as a proxy is generally too coarse: real tasks often rely on a much lower intrinsic dimensionality than the input space. As a result, this bound is not directly usable in real-world scenarios. Moreover, our experiment was intentionally simple, and extending this analysis to more complex or realistic graph structures remains challenging. Additional work will be needed to understand how topological expressivity behaves in heterogeneous graphs, dynamic graphs, or domains where node features are high-dimensional and noisy.

Overall, our study suggests that understanding graph topology requires considering not only how information propagates through the network, but also how much information the topology allows the model to express in the first place. We hope that this perspective will help motivate future work exploring the joint role of structure,

model architecture, and intrinsic data dimensionality in the design of more robust and expressive GNNs.

References

[1] Singh Akansha. 2025. Over-Squashing in Graph Neural Networks: A Comprehensive survey. arXiv:2308.15568 [cs.AI] https://arxiv.org/abs/2308.15568

[2] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks?. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2105.14491

[3] Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio', and Michael Bronstein. 2023. On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology. arXiv:2302.02941 [cs.LG] https://arxiv.org/abs/2302.02941

[4] Ningyuan Teresa Huang and Soledad Villar. 2021. A Short Tutorial on The Weisfeiler-Lehman Test And Its Variants. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8533–8537. doi:10.1109/icassp39728.2021.9413523

[5] Qin Jiang, Chengjia Wang, Michael Lones, and Wei Pang. 2025. Demystifying MPNNs: Message Passing as Merely Efficient Matrix Multiplication. arXiv:2502.00140 [cs.LG] https://arxiv.org/abs/2502.00140

[6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML] https://arxiv.org/abs/1710.10903

[7] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks? arXiv:1810.00826 [cs.LG] https://arxiv.org/abs/1810.00826

[8] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2021. Graph Neural Networks: A Review of Methods and Applications. arXiv:1812.08434 [cs.LG] https://arxiv.org/abs/1812.08434