On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology

Francesco Di Giovanni [1] **Lorenzo Giusti** [2] **Federico Barbero** [3] **Giulia Luise** [4] **Pietro Liò** [1] **Michael Bronstein** [3]

Abstract

Message Passing Neural Networks (MPNNs) are instances of Graph Neural Networks that leverage the graph to send messages over the edges. This inductive bias leads to a phenomenon known as over-squashing, where a node feature is insensitive to information contained at distant nodes. Despite recent methods introduced to mitigate this issue, an understanding of the causes for over-squashing and of possible solutions are lacking. In this theoretical work, we prove that: (i) Neural network width can mitigate over-squashing, but at the cost of making the whole network more sensitive; (ii) Conversely, depth cannot help mitigate over-squashing: increasing the number of layers leads to over-squashing being dominated by vanishing gradients; (iii) The graph topology plays the greatest role, since over-squashing occurs between nodes at high commute time. Our analysis provides a unified framework to study different recent methods introduced to cope with over-squashing and serves as a justification for a class of methods that fall under graph rewiring.

1. Introduction

Learning on graphs with Graph Neural Networks (GNNs) (Sperduti, 1993; Goller & Kuchler, 1996; Gori et al., 2005; Scarselli et al., 2008; Bruna et al., 2014; Defferrard et al., 2016) has become an increasingly flourishing area of machine learning. Typically, GNNs operate in the *message-passing paradigm* by exchanging information between nearby nodes (Gilmer et al., 2017), giving rise to the class of Message-Passing Neural Networks (MPNNs). While message-passing has demonstrated to be a useful inductive bias, it has also been shown that the paradigm has some fundamental flaws, from expressivity (Xu et al., 2019; Morris et al., 2019), to over-smoothing (Nt & Maehara, 2019; Cai & Wang, 2020; Bodnar et al., 2022; Rusch et al., 2022; Di Giovanni et al., 2022b; Zhao et al., 2022) and over-squashing. The first two limitations have been thoroughly investigated, however *less is known about over-squashing*.

Alon & Yahav (2021) described over-squashing as an issue emerging when MPNNs propagate messages across distant nodes, with the exponential expansion of the receptive field of a node leading to many messages being 'squashed' into fixed-size vectors. Topping et al. (2022) formally justified this phenomenon via a sensitivity analysis on the Jacobian of node features and, partly, linked it to the existence of edges with high-negative curvature. However, some important **questions are left open** from the analysis in Topping et al. (2022): (i) What is the impact of *width* in mitigating over-squashing? (ii) Can over-squashing be avoided by sufficiently *deep* models? (iii) How does over-squashing relate to the graph-spectrum and the underlying *topology* beyond curvature bounds that only apply to 2-hop propagation? The last point is particularly relevant due to recent works trying to combat over-squashing via methods that depend on the graph spectrum (Arnaiz-Rodríguez et al., 2022; Deac et al., 2022; Karhadkar et al., 2022). However, it is yet to be clarified if and why these works alleviate over-squashing.

In this work, we aim to address all the questions that are left open in Topping et al. (2022) to provide a better theoretical understanding on the causes of over-squashing as well as on what can and cannot fix it.

Contributions and outline. An MPNN is generally constituted by two main parts: a choice of architecture, and an underlying graph over which it operates. In this work, we investigate how these factors participate in the over-squashing phenomenon. We focus on the width and depth of the MPNN, as well as on the graph-topology.

- In Section 3, we prove that the *width* can mitigate over-squashing (Theorem 3.2), albeit at the potential cost of generalization. We also verify this with experiments.
- In Section 4, we show that depth may not be able to alleviate over-squashing. We identify two regimes. In

[1]University of Cambridge [2]Sapienza University [3]University of Oxford [4]Microsoft Research. Correspondence to: Francesco Di Giovanni <fd405@cam.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the first one, the number of layers is comparable to the graph diameter, and we prove that over-squashing is likely to occur among distant nodes (Theorem 4.1). In fact, the distance at which over-squashing happens is strongly dependent on the graph topology – as we validate experimentally. In the second regime, we consider an arbitrary (large) number of layers. We prove that at this stage the MPNN is, generally, dominated by vanishing gradients (Theorem 4.2). This result is of independent interest, since it characterizes analytically conditions of vanishing gradients of the loss for a large class of MPNNs that also include residual connections.

- In Section 5 we show that the topology of the graph has the greatest impact on over-squashing. In fact, we show that over-squashing happens among nodes with high commute time (Theorem 5.5) and we validate this empirically. This provides a unified framework to explain why all spatial and spectral rewiring approaches (discussed in Section 2.3) do mitigate over-squashing.

Figure 1. Effect of different rewirings $\mathcal{R}$ on the graph connectivity. The colouring denotes Commute Time – defined in Section 5 – w.r.t. to the star node. From left to right, the graphs shown are: the base, spatially rewired and spectrally rewired. The added edges significantly reduce the Commute Time and hence mitigate over-squashing in light of Theorem 5.5.

2. Background and related work

2.1. The message-passing paradigm

Let $\mathcal{G}$ be a graph with nodes V and edges E. The connectivity is encoded in the adjacency matrix $A \in \mathbb{R}^{n \times n}$, with n the number of nodes. We assume that $\mathcal{G}$ is undirected, connected, and that there are features $\{h_v^{(0)}\}_{v \in V} \subseteq \mathbb{R}^p$. Graph Neural Networks (GNNs) are functions of the form $\mathrm{GNN}(\mathcal{G}, \{h_v^{(0)}\}) \mapsto y_{\mathcal{G}}$, with parameters estimated via training and whose output $y_{\mathcal{G}}$ is either a node-level or graph-level prediction. The most studied class of GNNs, known as the Message Passing Neural Network (MPNN) (Gilmer et al., 2017), compute node representations by stacking layers of the form:

$$h_v^{(t)} = \mathrm{com}^{(t)}(h_v^{(t-1)}, \mathrm{agg}^{(t)}(\{h_u^{(t-1)} : (v,u) \in E\})),$$

for $t = 1, \dots, m$, where $\mathrm{agg}^{(t)}$ is some aggregation function invariant to node permutation, while $\mathrm{com}^{(t)}$ combines the node's current state with messages from its neighbours. In this work, we usually assume agg to be of the form

$$\mathrm{agg}^{(t)}(\{h_u^{(t-1)} : (v,u) \in E\}) = \sum_u A_{vu} h_u^{(t-1)}, \qquad (1)$$

where $A \in \mathbb{R}^{n \times n}$ is a Graph Shift Operator (GSO), meaning that $A_{vu} \neq 0$ if and only if $(v,u) \in E$. Typically, A is a (normalized) adjacency matrix that we also refer to as message-passing matrix. While instances of MPNN differ based on the choices of A and com, they all aggregate messages over the neighbours, such that in a layer, only nodes connected via an edge exchange messages. This presents two advantages: MPNNs (i) can capture graph-induced 'short-range' dependencies well, and (ii) are efficient, since they can leverage the sparsity of the graph. Nonetheless, MPNNs have been shown to suffer from a few drawbacks, including limited expressive power and over-squashing. The problem of expressive power stems from the equivalence of MPNNs to the Weisfeiler-Leman graph isomorphism test (Xu et al., 2019; Morris et al., 2019). This framework has been studied extensively (Jegelka, 2022). On the other hand, the phenomenon of over-squashing, which is the main focus of this work, is more elusive and less understood. We review what is currently known about it in the next subsection.

2.2. The problem: introducing over-squashing

Since in an MPNN the information is aggregated over the neighbours, for a node v to be affected by features at distance r, an MPNN needs at least r layers (Barceló et al., 2019). It has been observed though that due to the expansion of the receptive field of a node, MPNNs may end up sending a number of messages growing exponentially with the distance r, leading to a potential loss of information known as over-squashing (Alon & Yahav, 2021). Topping et al. (2022) showed that for an MPNN with message-passing matrix A as in Eq. (1) and scalar features, given nodes v, u at distance r, we have $\left|\partial h_v^{(r)} / \partial h_u^{(0)}\right| \leq c\,(A^r)_{vu}$; with c a constant depending on the Lipschitz regularity of the model. If $(A^r)_{vu}$ decays exponentially with r, then the feature of v is insensitive to the information contained at u. Moreover, Topping et al. (2022) showed that over-squashing is related to the existence of edges with high negative curvature. Such characterization though only applies to propagation of information up to 2 hops.

2.3. Related work

Multiple solutions to mitigate over-squashing have already been proposed. We classify them below; in Section 5, we provide a unified framework that encompasses all such solu-

tions. We first introduce the following notion:

Definition 2.1. Consider an MPNN, a graph G with adjacency A, and a map $R : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$. We say that G has been rewired by R, if the messages are exchanged on $R(G)$ instead of G, with $R(G)$ the graph with adjacency $R(A)$.

Recent approaches to combat over-squashing share a common idea: replace the graph G with a rewired graph $R(G)$ enjoying better connectivity Figure 1. We then distinguish these works based on the choice of the rewiring R.

Spatial methods. Since MPNNs fail to propagate information to distant nodes, a solution consists in replacing G with $R(G)$ such that $\mathrm{diam}(R(G)) \ll \mathrm{diam}(G)$. Typically, this is achieved by either explicitly adding edges (possibly attributed) between distant nodes (Gutteridge et al., 2022; Abboud et al., 2022; Gutteridge et al., 2023) or by allowing distant nodes to communicate through higher-order structures (e.g., cellular or simplicial complexes, (Bodnar et al., 2021a;b), which requires additional domain knowledge and incurs a computational overhead). Graph-Transformers can be seen as an extreme example of rewiring, where $R(G)$ is a complete graph with edges weighted via attention (Kreuzer et al., 2021; Mialon et al., 2021; Ying et al., 2021; Rampasek et al., 2022). While these methods do alleviate over-squashing, since they bring all pair of nodes closer, they come at the expense of making the graph $R(G)$ much denser. In turn, this has an impact on computational complexity and introduces the risk of mixing local and non-local interactions.

We include in this group Topping et al. (2022) and Banerjee et al. (2022), where the rewiring is surgical – but requires specific pre-processing – in the sense that G is replaced by $R(G)$ where edges have only been added to 'mitigate' bottlenecks as identified, for example, by negative curvature (Ollivier, 2007; Di Giovanni et al., 2022a).

We finally mention that spatial rewiring, intended as accessing information beyond the 1-hop when updating node features, is common to many existing frameworks (Abu-El-Haija et al., 2019; Klicpera et al., 2019; Chen et al., 2020; Ma et al., 2020; Wang et al., 2020; Nikolentzos et al., 2020). However, this is usually done via powers of the adjacency matrix, which is the main culprit for over-squashing (Topping et al., 2022). Accordingly, although the diffusion operators A^k allow to aggregate information over non-local hops, they are not suited to mitigate over-squashing.

Spectral methods. The connectedness of a graph G can be measured via a quantity known as the Cheeger constant, defined as follows (Chung & Graham, 1997):

Definition 2.2. For a graph G, the Cheeger constant is

$$h_{\mathrm{Cheeg}} = \min_{U \subset V} \frac{|\{(u,v) \in E : u \in U, v \in V \setminus U\}|}{\min(\mathrm{vol}(U), \mathrm{vol}(V \setminus U))},$$

where $\mathrm{vol}(U) = \sum_{u \in U} d_u$, with d_u the degree of node u.

The Cheeger constant h_{Cheeg} represents the energy required to disconnect G into two communities. A small h_{Cheeg} means that G generally has two communities separated by only few edges – over-squashing is then expected to occur here if information needs to travel from one community to the other. While h_{Cheeg} is generally intractable to compute, thanks to the Cheeger inequality we know that $h_{\mathrm{Cheeg}} \geq \lambda_1$, where λ_1 is the positive, smallest eigenvalue of the graph Laplacian. Accordingly, a few new approaches have suggested to choose a rewiring that depends on the spectrum of G and yields a new graph satisfying $h_{\mathrm{Cheeg}}(R(G)) > h_{\mathrm{Cheeg}}(G)$. This strategy includes Arnaiz-Rodríguez et al. (2022); Deac et al. (2022); Karhadkar et al. (2022). It is claimed that sending messages over such a graph $R(G)$ alleviates over-squashing, however this has not been shown analytically yet.

The goal of this work. The analysis of Topping et al. (2022), which represents our current theoretical understanding of the over-squashing problem, leaves some important open questions which we address in this work: (i) We study the role of the *width* in mitigating over-squashing; (ii) We analyse what happens when the *depth* exceeds the distance among two nodes of interest; (iii) We prove how over-squashing is related to the graph structure (beyond local curvature-bounds) and its spectrum. As a consequence of (iii), we provide a unified framework to explain how spatial and spectral approaches alleviate over-squashing. We reference here the concurrent work of Black et al. (2023), who, similarly to us, drew a strong connection between over-squashing and Effective Resistance (see Section 5).

Notations and conventions to improve readability. In the following, to prioritize readability we often leave sharper bounds with more optimal and explicit terms to the Appendix. From now on, p always denotes the width (hidden dimension) while m is the depth (i.e. number of layers). The feature of node v at layer t is written as $h_v^{(t)}$. Finally, we write $[\ell] = \{1, \dots, \ell\}$ for any integer ℓ. All proofs can be found in the Appendix.

3. The impact of width

In this Section we assess whether the width of the underlying MPNN can mitigate over-squashing and the extent to which this is possible. In order to do that, we extend the sensitivity analysis in Topping et al. (2022) to higher-dimensional node features. We consider a class of MPNNs parameterised by neural networks, of the form:

$$h_v^{(t+1)} = c_r W_r^{(t)} h_v^{(t)} + c_a W_a^{(t)} \sum_u A_{vu} h_u^{(t)}; \quad (2)$$

where σ is a pointwise-nonlinearity, $W_r^{(t)}, W_a^{(t)} \in \mathbb{R}^{p \times p}$ are learnable weight matrices and A is a graph shift operator. Note that Eq. (2) includes common MPNNs such as GCN (Kipf & Welling, 2017), SAGE (Hamilton et al., 2017), and GIN (Xu et al., 2019), where A is one of $D^{-1/2}AD^{-1/2}$, $D^{-1}A$ and A, respectively, with D the diagonal degree matrix. In Appendix B, we extend the analysis to a more general class of MPNNs (see Theorem B.1), which includes stacking multiple nonlinearities. We also emphasize that the positive scalars c_r, c_a represent the weighted contribution of the residual term and of the aggregation term, respectively. To ease the notations, we introduce a family of message-passing matrices that depend on c_r, c_a.

Definition 3.1. For a graph shift operator A and constants $c_r, c_a > 0$, we define $S_{r,a} := c_r I + c_a A \in \mathbb{R}^{n \times n}$ to be the message-passing matrix adopted by the MPNN.

As in Xu et al. (2018) and Topping et al. (2022), we study the propagation of information in the MPNN via the Jacobian of node features after m layers.

Theorem 3.2 (Sensitivity bounds). Consider an MPNN as in Eq. (2) for m layers, with c the Lipschitz constant of the nonlinearity σ and w the maximal entry-value over all weight matrices. For $v, u \in V$ and width p, we have

$$\left\| \frac{\partial h_v^{(m)}}{\partial h_u^{(0)}} \right\|_1 \leq \underbrace{(c\, w p)^m}_{\text{model}} \underbrace{(S_{r,a}^m)_{vu}}_{\text{topology}}; \tag{3}$$

with $S_{r,a}^m$ the m^{th}-power of $S_{r,a}$ introduced in Definition 3.1.

Over-squashing occurs if the right hand side of (3) is too small – this will be related to the distance among v and u in Section 4.1. A small derivative of $h_v^{(m)}$ with respect to $h_u^{(0)}$ means that after m layers, the feature at v is mostly insensitive to the information initially contained at u, and hence that messages have not been propagated effectively. Theorem 3.2 clarifies how the model can impact over-squashing through (i) its Lipschitz regularity, c and w, and (ii) its width p. In fact, given a graph G such that $(S_{r,a}^m)_{vu}$ decays exponentially with m, the MPNN can compensate by increasing the width p and the magnitude of w and c. This confirms analytically the discussion in Alon & Yahav (2021): a larger hidden dimension p does mitigate over-squashing. However, this is not an optimal solution since increasing the contribution of the model (i.e. the term cwp) may lead to over-fitting and poorer generalization (Bartlett et al., 2017). Taking larger values of c, w, p affects the model globally and does not target the sensitivity of specific node pairs induced by the topology via $S_{r,a}$.

Validating the theoretical results. We validate empirically the message from Theorem 3.2: if the task presents long-range dependencies, increasing the hidden dimension mitigates over-squashing and therefore has a positive impact on the performance. We consider the following 'graph transfer' task, building upon Bodnar et al. (2021a): given a graph, consider source and target nodes, placed at distance r from each other. We assign a one-hot encoded label to the target and a constant unitary feature vector to all other nodes. The goal is to assign to the source node the feature vector of the target. Partly due to over-squashing, performance is expected to degrade as r increases.

To validate that this holds irrespective of the graph structure, we evaluate across three graph topologies, called CrossedRing, Ring and CliquePath - see Appendix E for further details. While the topology is also expected to affect the performance (as confirmed in Section 4), given a fixed topology, we expect the model to benefit from an increase of hidden dimension.

To verify this behaviour, we evaluate GCN (Kipf & Welling, 2017) on the three graph transfer tasks increasing the hidden dimension, but keeping the number of layers equal to the distance between source and target, as shown in Figure 2. The results verify the intuition from the theorem that a higher hidden dimension helps the GCN model solve the task to larger distances across the three graph-topologies.

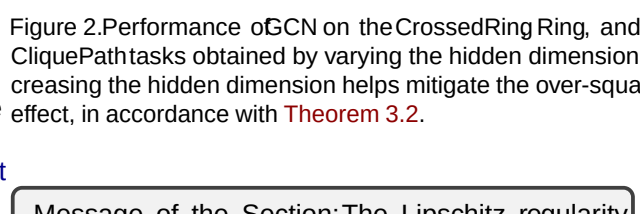

Figure 2. Performance of GCN on the CrossedRing, Ring, and CliquePath tasks obtained by varying the hidden dimension. Increasing the hidden dimension helps mitigate the over-squashing effect, in accordance with Theorem 3.2.

Message of the Section: The Lipschitz regularity, weights, and width of the underlying MPNN can help mitigate the effect of over-squashing. However, this is a remedy that comes at the expense of generalization and does not address the real culprit behind over-squashing: the graph-topology.

4. The impact of depth

Consider a graph G and a task with `long-range' dependencies, meaning that there exists (at least) a node v whose embedding has to account for information contained at some node u at distance $r \gg 1$. One natural attempt at resolving over-squashing amounts to increasing the number of layers m to compensate for the distance. We prove that the depth of the MPNN will, generally, not help with over-squashing. We show that: (i) When the depth m is comparable to the distance, over-squashing is bound to occur among distant nodes – in fact, the distance at which over-squashing happens, is strongly dependent on the underlying topology; (ii) If we take a large number of layers to cover the long-range interactions, we rigorously prove under what exact conditions, MPNNs incur the vanishing gradients problem.

4.1. The shallow-diameter regime: over-squashing occurs among distant nodes

Consider the scenario above, with two nodes v, u, whose interaction is important for the task, at distance r. We first focus on the regime $m \approx r$. We refer to this as the shallow-diameter regime, since the number of layers m is comparable to the diameter of the graph.

From now on, we set $\hat{A} = D^{-1/2} A D^{-1/2}$, where we recall that A is the adjacency matrix and D is the degree matrix. This is not restrictive, but allows us to derive more explicit bounds and, later, bring into the equation the spectrum of the graph. We note that results can be extended easily to $D^{-1}A$, given that this matrix is similar to $\hat{A}$, and, in expectation, to A by normalizing the Jacobian as in Xu et al. (2019) and Section A in the Appendix of Topping et al. (2022).

Theorem 4.1 (Over-squashing among distant nodes). *Given an MPNN as in Eq. (2), with $c_a \leq 1$, let $v, u \in V$ be at distance r. Let c be the Lipschitz constant of ψ, w the maximal entry-value over all weight matrices, d_{min} the minimal degree of G, and $\ell(v,u)$ the number of walks from v to u of maximal length r. For any $0 \leq k < r$, there exists $C_{k,r+k} > 0$ independent of r and of the graph, such that*

$$\left\| \frac{\partial h_v^{(r+k)}}{\partial h_u^{(0)}} \right\|_{L_1} \leq C_{k,r+k}(v,u) \left(\frac{2c\,wp}{d_{min}} \right)^r. \tag{4}$$

To understand the bound above, take $k < r$ and assume that nodes v, u are `badly' connected, meaning that the number of walks $\ell_{r+k}(v,u)$ of length at most $r+k$, is small. If $2c\,wp < d_{min}$, then the bound on the Jacobian in Eq. (4) decays exponentially with the distance r. Note that the bound above considers d_{min} and ℓ_{r+k} as a worst case scenario. If one has a better understanding of the topology of the graph, sharper bounds can be derived by estimating $(S^r_{r;a})_{vu}$. Theorem 4.1 implies that, when the depth m is comparable to the diameter of G, over-squashing becomes an issue if the task depends on the interaction of nodes v, u at `large' distance r. In fact, Theorem 4.1 shows that the distance at which the Jacobian sensitivity falls below a given threshold, depends on both the model, via c, w, p, and on the graph, through d_{min} and $\ell_{r+k}(v,u)$. We finally observe that Theorem 4.1 generalizes the analysis in Topping et al. (2022) in multiple ways: (i) it holds for any width $p > 1$; (ii) it includes cases where $m > r$; (iii) it provides explicit estimates in terms of number of walks and degree information.

Remark. What if $2c\,wp > d_{min}$? Taking larger weights and hidden dimension increases the sensitivity of node features. However, this occurs everywhere in the graph the same. Accordingly, nodes at shorter distances will, on average, still have sensitivity exponentially larger than nodes at large distance. This is validated in our synthetic experiments below, where we do not have constraints on the weights.

Validating the theoretical results. From Theorem 4.1, we derive a strong indication of the difficulty of a task by calculating an upper bound on the Jacobian. We consider the same graph transfer tasks introduced above, namely CrossedRing, Ring, and CliquePath. For these special cases, we can derive a refined version of the r.h.s in Eq. (4): in particular, $k = 0$ (i.e. the depth coincides with the distance among source and target) and the term $\ell_r(v,u)(d_{min})^{-r}$ can be replaced by the exact quantity $(S^r_{r;a})_{vu}$. Fixing a distance r between source u and target v then, if we consider for example the GCN-case, we have $S_{r;a} = A$ so that the term $(S^r_{r;a})_{vu}$ can be computed explicitly:

$$(S^r_{r;a})_{vu} = (3/2)^{(r-1)} \quad \text{for CrossedRing}$$
$$(S^r_{r;a})_{vu} = 2^{(r-1)} \quad \text{for Ring}$$
$$(S^r_{r;a})_{vu} = 2^{(r-2)} \binom{r}{\frac{r}{2}-2} \quad \text{for CliquePath}$$

Given an MPNN, terms like c, w, p entering Theorem 4.1 are independent of the graph-topology and hence can be assumed to behave, roughly, the same across different graphs. As a consequence, we can expect over-squashing to be more problematic for CliquePath, followed by Ring, and less prevalent comparatively in CrossedRing. Figure 3 shows the behaviour of GIN (Xu et al., 2019), SAGE (Hamilton et al., 2017), GCN (Kipf & Welling, 2017), and GAT (Veličković et al., 2018) on the aforementioned tasks. We verify the conjectured difficulty. CliquePath is the consistently hardest task, followed by Ring, and CrossedRing. Furthermore, the decay of the performance to random guessing for the same architecture across different graph topologies highlights that this drop cannot be simply labelled as `vanishing gradients' since for certain topologies the same model can, in fact, achieve perfect accuracy. This validates that the underlying topology has a strong impact on the distance at which over-squashing is expected to happen. Moreover, we confirm

that in the regime where the depth m is comparable to the distance r, over-squashing will occur if m is large enough.

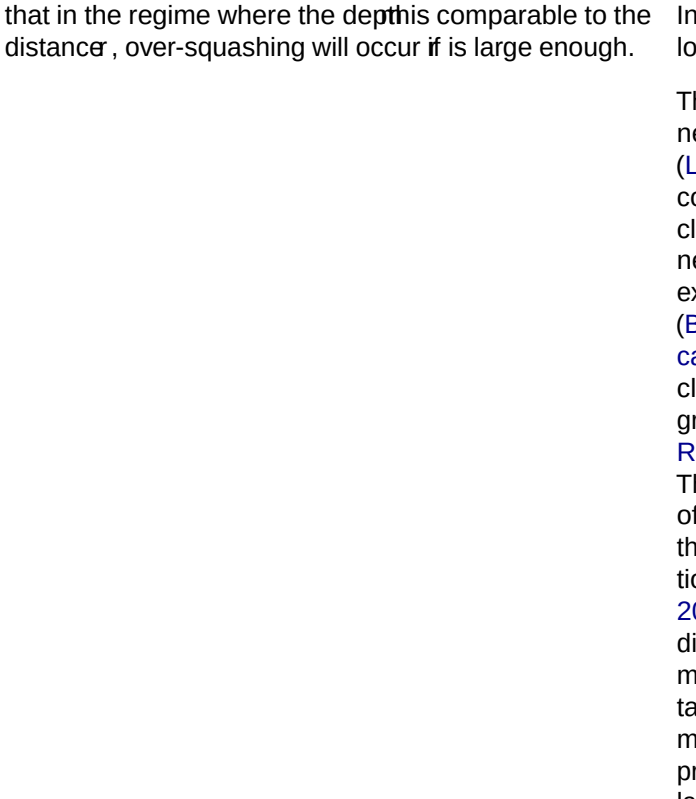

Figure 3. Performance of GIN, SAGE, GCN, and GAT on the CliquePath, Ring, and CrossedRing tasks. In the case where depth and distance are comparable, over-squashing highly depends on the topology of the graph as we increase the distance.

4.2. The deep regime: vanishing gradients dominate

We now focus on the regime where the number of layers $m \gg r$ is large. We show that in this case, vanishing gradients can occur and make the entire model insensitive. Given a weight $W^{(k)}$ entering a layer k, one can write the gradient of the loss after m layers as (Pascanu et al., 2013)

$$\frac{\partial \mathcal{L}}{\partial W^{(k)}} = \sum_{v,u \in V} \frac{\partial \mathcal{L}}{\partial h_v^{(m)}} \frac{\partial h_v^{(m)}}{\partial W^{(k)}} \underbrace{\frac{\partial h_v^{(m)}}{\partial h_u^{(k)}}}_{\text{sensitivity}} \tag{5}$$

We provide exact conditions for MPNNs to incur the vanishing gradient problem, intended as the gradients of the loss decaying exponentially with the number of layers m.

Theorem 4.2 (Vanishing gradients). Consider an MPNN as in Eq. (2) for m layers with a quadratic loss $\mathcal{L}$. Assume that (i) ϕ has Lipschitz constant c_ϕ and $\phi(0) = 0$, and (ii) weight matrices have spectral norm bounded by $\gamma > 0$. Given any weight $W^{(k)}$ entering a layer k, there exists a constant $C > 0$ independent of m, such that

$$\left\| \frac{\partial \mathcal{L}}{\partial W^{(k)}} \right\| \leq C \, (c_\phi (c_r + c_a))^{m-k} \, (1 + (c_\phi (c_r + c_a))^m). \tag{6}$$

In particular, if $c_\phi (c_r + c_a) < 1$, then the gradients of the loss decay to zero exponentially fast with m.

The problem of vanishing gradients for graph convolutional networks have been studied from an empirical perspective (Li et al., 2019; 2021). Theorem 4.2 provides sufficient conditions for the vanishing of gradients to occur in a large class of MPNNs that also include (a form of) residual connections through the contribution of c_r in Eq. (2). This extends a behaviour studied for Recurrent Neural Networks (Bengio et al., 1994; Hochreiter & Schmidhuber, 1997; Pascanu et al., 2013; Rusch & Mishra, 2021a;b) to the MPNN class. We also mention that some discussion on vanishing gradients for MPNNs can be found in Ruiz et al. (2020) and Rusch et al. (2022). A few final comments are in order. (i) The bound in Theorem 4.2 seems to 'hide' the contribution of the graph. This is, in fact, because the spectral norm of the graph operator $\mathbf{S}_{r;a}$ is $c_r + c_a$ – we reserve the investigation of more general graph shift operators (Dasoulas et al., 2021) to future work. (ii) Theorem 4.1 shows that if the distance r is large enough and we take the number of layers m s.t. $m \gg r$, over-squashing arises among nodes at distance r. Taking the number of layers large enough though, may incur the vanishing gradient problem Theorem 4.2. In principle, there might be an intermediate regime where m is larger than r, but not too large, in which the depth could help with over-squashing before it leads to vanishing gradients. Given a graph $\mathcal{G}$, and bounds on the Lipschitz regularity and width, we conjecture though that there exists $r_\mathcal{G}$, depending on the topology of $\mathcal{G}$, such that if the task has interactions at distance $r > r_\mathcal{G}$, no number of layers can allow the MPNN class to solve it. This is left for future work.

> Message of the Section: Increasing the depth m will, in general, not fix over-squashing. As we increase m, MPNNs transition from over-squashing (Theorem 4.1) to vanishing gradients (Theorem 4.2).

5. The impact of topology

We finally discuss the impact of the graph topology, and in particular of the graph spectrum, on over-squashing. This allows us to draw a unified framework that shows why existing approaches manage to alleviate over-squashing by either spatial or spectral rewiring (Section 2.3).

5.1. On over-squashing and access time

Throughout the section we relate over-squashing to well-known properties of random walks on graphs. To this aim, we first review basic concepts about random walks.

Access and commute time. A Random Walk (RW) on $\mathcal{G}$ is a Markov chain which, at each step, moves from a node v to one of its neighbours with probability $1/d_v$. Several

properties about RWs have been studied. We are particularly interested in the notion of access time $t(v;u)$ and of commute time $(v;u)$ (see Figure 1). The access time $t(v;u)$ (also known as hitting time) is the expected number of steps before node u is visited for a RW starting from node v. The commute time instead, represents the expected number of steps in a RW starting v to reach node u and come back. A high access (commute) time means that nodes $v;u$ generally struggle to visit each other in a RW – this can happen if nodes are far-away, but it is in fact more general and strongly dependent on the topology.

Some connections between over-squashing and the topology have already been derived (Theorem 4.1), but up to this point `topology' has entered the picture through `distances' only. In this section, we further link over-squashing to other quantities related to the topology of the graph, such as access time, commute time and the Cheeger constant. We ultimately provide a uni ed framework to understand how existing approaches manage to mitigate over-squashing via graph-rewiring.

Integrating information across different layers. We consider a family of MPNNs of the form

$$h_v^{(t)} = \mathrm{ReLU}\left(W^{(t)}\left(c_r h_v^{(t-1)} + c_a (A h^{(t-1)})_v\right)\right) \quad (7)$$

Similarly to Kawaguchi (2016); Xu et al. (2018), we require the following:

Assumption 5.1. All paths in the computation graph of the model are activated with the same probability of success

Take two nodes $v \neq u$ at distance $r \geq 1$ and imagine we are interested in sending information from u to v. Given a layer $k < m$ of the MPNN, by Theorem 4.1 we expect that $h_v^{(m)}$ is much more sensitive to the information contained at the same node v at an earlier layer k, i.e. $h_v^{(k)}$, rather than to the information contained at a distant node u, i.e. $h_u^{(k)}$. Accordingly, we introduce the following quantity:

$$J_k^{(m)}(v;u) := \frac{1}{d_v}\frac{\partial h_v^{(m)}}{\partial h_v^{(k)}} \quad \frac{1}{\sqrt{d_v d_u}}\frac{\partial h_v^{(m)}}{\partial h_u^{(k)}}.$$

We note that the normalization by degree stems from our choice $A = D^{-1/2}AD^{-1/2}$. We provide an intuition for this term. Say that node v at layer m of the MPNN is mostly insensitive to the information sent from u at layer k. Then, on average, we expect $\|\partial h_v^{(m)}/\partial h_u^{(k)}\| \ll \|\partial h_v^{(m)}/\partial h_v^{(k)}\|$. In the opposite case instead, we expect, on average, that $\|\partial h_v^{(m)}/\partial h_u^{(k)}\| \gg \|\partial h_v^{(m)}/\partial h_v^{(k)}\|$. Therefore $\|J_k^{(m)}(v;u)\|$ will be larger when v is (roughly) independent of the information contained at u at layer k. We extend the same argument by accounting for messages sent at each layer $k \leq m$.

De nition 5.2. The Jacobian obstruction of node v with respect to node u after m layers is $O^{(m)}(v;u) = \sum_{k=0}^{m} \|J_k^{(m)}(v;u)\|$.

As motivated above, a larger $O^{(m)}(v;u)$ means that, after m layers, the representation of node v is more likely to be insensitive to information contained at u and conversely, a small $O^{(m)}(v;u)$ means that node v is, on average, able to receive information from u. Differently from the Jacobian bounds of the earlier sections, here we consider the contribution coming from all layers $k \leq m$ (note the sum over layers k in De nition 5.2).

Theorem 5.3 (Over-squashing and access-time). Consider an MPNN as in Eq. (7) and let Assumption 5.1 hold. If α is the smallest singular value across all weight matrices and $c_r; c_a$ are such that $|\alpha(c_r + c_a)| = 1$, then, in expectation, we have

$$O^{(m)}(v;u) \quad \frac{1}{c_a}\frac{t(u;v)}{2|E|} + o(m);$$

with $o(m) \to 0$ exponentially fast with m.

We note that an exact expansion of the term $o(m)$ is reported in the Appendix. We also observe that more general bounds are possible if $|\alpha(c_r + c_a)| < 1$ – however, they will progressively become less informative in the limit $|\alpha(c_r + c_a)| \to 0$. Theorem 5.3 shows that the obstruction is a function of the access time $t(u;v)$; high access time, on average, translates into high obstruction for node v to receive information from node u inside the MPNN. This resonates with the intuition that access time is a measure of how easily a `diffusion' process starting at u reaches v. We emphasize that the obstruction provided by the access time cannot be xed by increasing the number of layers and in fact this is independent of the number of layers, further corroborating the analysis in Section 4. Next, we relate over-squashing to commute time, and hence, to effective resistance.

5.2. On over-squashing and commute time

We now restrict our attention to a slightly more special form of over-squashing, where we are interested in nodes $v; u$ exchanging information both ways – differently from before where we looked at node v receiving information from node u. Following the same intuition described previously, we introduce the symmetric quantity:

$$\bar{J}_k^{(m)}(v;u) := \frac{1}{d_v}\frac{\partial h_v^{(m)}}{\partial h_v^{(k)}} \quad \frac{1}{\sqrt{d_v d_u}}\frac{\partial h_v^{(m)}}{\partial h_u^{(k)}} + \frac{1}{d_u}\frac{\partial h_u^{(m)}}{\partial h_u^{(k)}} \quad \frac{1}{\sqrt{d_v d_u}}\frac{\partial h_u^{(m)}}{\partial h_v^{(k)}}.$$

Once again, we expect that $\|\bar{J}_k^{(m)}(v;u)\|$ is larger if nodes $v; u$ are failing to communicate in the MPNN, and con-

versely to be smaller whenever the communication is sufficiently robust. Similarly, we integrate the information collected at each layer $k \leq m$.

Definition 5.4. The symmetric Jacobian obstruction of nodes v, u after m layers is $\mathcal{O}^{(m)}(v,u) = \sum_{k=0}^{m} \|J_k^{(m)}(v,u)\|$.

The intuition of comparing the sensitivity of a node v with a different node u and to itself, and then swapping the roles of v and u, resembles the concept of commute time $\kappa(v,u)$. In fact, this is not a coincidence:

Theorem 5.5 (Over-squashing and commute-time)**.** Consider an MPNN as in Eq. (7) with c_r the maximal spectral norm of the weight matrices and c_a the minimal singular value. Let Assumption 5.1 hold. If $(c_r + c_a) \leq 1$, then there exists γ_G, independent of nodes v, u, such that in expectation, we have

$$\gamma_G(1 - o(m)) \frac{1}{c_a} \frac{\kappa(v,u)}{2|E|} \leq \mathcal{O}^{(m)}(v,u) \leq \frac{1}{c_a} \frac{\kappa(v,u)}{2|E|},$$

with $o(m) \to 0$ exponentially fast with m increasing.

We note that an explicit expansion of the $o(m)$-term is reported in the proof of the Theorem in the Appendix. By the previous discussion, a smaller $\mathcal{O}^{(m)}(v,u)$ means v is more sensitive to u in the MPNN (and vice versa when $\mathcal{O}^{(m)}(v,u)$ is large). Therefore, Theorem 5.5 implies that nodes at small commute time will exchange information better in an MPNN and conversely for those at high commute time. This has some important consequences:

(i) When the task only depends on local interactions, the property of MPNN of reducing the sensitivity to messages from nodes with high commute time can be beneficial since it decreases harmful redundancy.

(ii) Over-squashing is an issue when the task depends on the interaction of nodes with high commute time.

(iii) The commute time represents an obstruction to the sensitivity of an MPNN which is independent of the number of layers, since the bounds in Theorem 5.5 are independent of m (up to errors decaying exponentially fast with m).

We note that the very same comments hold in the case of access time as well if, for example, the task depends on node v receiving information from node u but not on u receiving information from v.

5.3. A unified framework

Why spectral-rewiring works. First, we justify why the spectral approaches discussed in Section 2.3 mitigate over-squashing. This comes as a consequence of Lovász (1993) and Theorem 5.5:

Corollary 5.6. Under the assumptions of Theorem 5.5, for any $v, u \in V$, we have

$$\mathcal{O}^{(m)}(v,u) \leq \frac{4}{c_a} \frac{1}{h_{\text{Cheeg}}^2}.$$

Corollary 5.6 essentially tells us that the obstruction among all pairs of nodes decreases (so better information flow) if the MPNN operates on a graph G with larger Cheeger constant. This rigorously justifies why recent works like Arnaiz-Rodríguez et al. (2022); Deac et al. (2022); Karhadkar et al. (2022) manage to alleviate over-squashing by propagating information on a rewired graph $R(G)$ with larger Cheeger constant h_{Cheeg}. Our result also highlights why bounded-degree expanders are particularly suited - as leveraged in Deac et al. (2022) – given that their commute time is only $O(|E|)$ (Chandra et al., 1996), making the bound in Theorem 5.5 scale as $O(1)$ w.r.t. the size of the graph. In fact, the concurrent work of Black et al. (2023) leverages directly the effective resistance of the graph $\text{Res}(v,u) = \kappa(v,u)/2|E|$ to guide a rewiring that improves the graph connectivity and hence mitigates over-squashing.

Why spatial-rewiring works. Chandra et al. (1996) proved that the commute time satisfies $\kappa(v,u) = 2|E|\text{Res}(v,u)$, with $\text{Res}(v,u)$ the effective resistance of nodes v, u. $\text{Res}(v,u)$ measures the voltage difference between nodes v, u if a unit current flows through the graph from v to u and we take each edge to represent a unit resistance (Thomassen, 1990; Ollivier et al., 2018), and has also been used in Velingker et al. (2022) as a form of structural encoding. Therefore, we emphasize that Theorem 5.5 can be equivalently rephrased as saying that nodes at high-effective resistance struggle to exchange information in an MPNN and vice versa for node at low effective resistance. We recall that a result known as Rayleigh's monotonicity principle (Thomassen, 1990), asserts that the total effective resistance $\text{Res}_G = \sum_{v,u} \text{Res}(v,u)$ decreases when adding new edges – which offer a new interpretation as to why spatial methods help combat over-squashing.

What about curvature? Our analysis also sheds further light on the relation between over-squashing and curvature derived in Topping et al. (2022). If the effective resistance is bounded from above, this leads to lower bounds for the resistance curvature introduced in Devriendt & Lambiotte (2022) and hence, under some assumptions, for the Ollivier curvature too (Ollivier, 2007; 2009). Our analysis then recovers why preventing the curvature from being `too' negative has benefits in terms of reducing over-squashing.

About the Graph Transfer task. We finally note that the results in Figure 3 also validate the theoretical findings of Theorem 5.5. If v, u represent target and source nodes on the different graph-transfer topologies, then $\mathrm{Reff}(v, u)$ is highest for the CliquePath and lowest for the CrossedRing. Once again, the distance is only a partial information. Effective resistance provides a better picture for the impact of topology to over-squashing and hence the accuracy on the task; in Appendix F we further validate that via a synthetic experiment where we study how the propagation of a signal in a MPNN is affected by the effective resistance of $\mathcal{G}$.

> Message of the Section: MPNNs struggle to send information among nodes with high commute (access) time (equivalently, effective resistance). This connection between over-squashing and commute (access) time provides a unified framework for explaining why spatial and spectral-rewiring approaches manage to alleviate over-squashing.

6. Conclusion and discussion

What did we do? In this work, we investigated the role played by width, depth, and topology in the over-squashing phenomenon. We have proved that, while width can partly mitigate this problem, depth is, instead, generally bound to fail since over-squashing spills into vanishing gradients for a large number of layers. In fact, we have shown that the graph-topology plays the biggest role, with the commute (access) time providing a strong indicator for whether over-squashing is likely to happen independently of the number of layers. As a consequence of our analysis, we can draw a unified framework where we can rigorously justify why all recently proposed rewiring methods do alleviate over-squashing.

Limitations. Strictly speaking, the analysis in this work applies to MPNNs that weigh each edge contribution the same, up to a degree normalization. In the opposite case, which, for example, includes GAT (Veličković et al., 2018) and GatedGCN (Bresson & Laurent, 2017), over-squashing can be further mitigated by pruning the graph, hence alleviating the dispersion of information. However, the attention (gating) mechanism can fail if it is not able to identify which branches to ignore and can even amplify over-squashing by further reducing `useful' pathways. In fact, GAT still fails on the Graph Transfer task of Section 4, albeit it seems to exhibit slightly more robustness. Extending the Jacobian bounds to this case is not hard, but will lead to less transparent formulas: a thorough analysis of this class, is left for future work. We also note that determining when the sensitivity is `too' small is generally also a function of the resolution of the readout, which we have not considered. Finally, Theorem 5.5 holds in expectation over the nonlinearity and, generally, Definition 5.2 encodes an average type of behaviour: a more refined (and exact) analysis is left for future work.

Where to go from here. We believe that the relation between over-squashing and vanishing gradient deserves further analysis. In particular, it seems that there is a phase transition that MPNNs undergo from over-squashing of information between distant nodes, to vanishing of gradients at the level of the loss. In fact, this connection suggests that traditional methods that have been used in RNNs and GNNs to mitigate vanishing gradients, may also be beneficial for over-squashing. On a different note, this work has not touched on the important problem of over-smoothing; we believe that the theoretical connections we have derived, based on the relation between over-squashing, commute time, and Cheeger constant, suggest a much deeper interplay between these two phenomena. Finally, while this analysis confirms that both spatial and spectral-rewiring methods provably mitigate over-squashing, it does not tell us which method is preferable, when, and why. We hope that the theoretical investigation of over-squashing we have provided here, will also help tackle this important methodological question.

Acknowledgements

We are grateful to Adrián Arnaiz, Johannes Lutzeyer, and Ismail Ceylan for providing insightful and detailed feedback and suggestions on an earlier version of the manuscript. We are also particularly thankful to Jacob Bamberger for helping us fix a technical assumption in one of our arguments. Finally, we are grateful to the anonymous reviewers for their input. This research was supported in part by ERC Consolidator grant No. 274228 (LEMAN) and by the EU and Innovation UK project TROPHY.

References

Abboud, R., Dimitrov, R., and Ceylan, I. I. Shortest path networks for graph property prediction. InThe First Learning on Graphs Conference, 2022. URLhttps://openreview.net/forum?id=mWzWvMxuFg1 .

Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Ver Steeg, G., and Galstyan, A. Mixhop: Higher-order graph convolutional architectures via sparsi ed neighborhood mixing.In International conference on machine learning, pp. 21–29. PMLR, 2019.

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. InInternational Conference on Learning Representations, 2021.

Arnaiz-Rodríguez, A., Begga, A., Escolano, F., and Oliver, N. DiffWire: Inductive Graph Rewiring via the Lováz Bound. In The First Learning on Graphs Conference, 2022. URL https://openreview.net/pdf?id=IXvfIex0mX6f .

Banerjee, P. K., Karhadkar, K., Wang, Y. G., Alon, U., and Montúfar, G. Oversquashing in gnns through the lens of information contraction and graph expansion.In Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1–8. IEEE, 2022.

Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J., and Silva, J. P. The logical expressiveness of graph neural networks. InInternational Conference on Learning Representations, 2019.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks.Advances in neural information processing systems, 30, 2017.

Bengio, Y., Simard, P., and Frasconi, P. Learning long-term dependencies with gradient descent is dif cult.IEEE transactions on neural networks, 5(2):157–166, 1994.

Black, M., Nayyeri, A., Wan, Z., and Wang, Y. Understanding oversquashing in gnns through the lens of effective resistance.arXiv preprint arXiv:2302.06835, 2023.

Bodnar, C., Frasca, F., Otter, N., Wang, Y., Lio, P., Montufar, G. F., and Bronstein, M. Weisfeiler and lehman go cellular: Cw networks. InAdvances in Neural Information Processing Systems, volume 34, pp. 2625–2640, 2021a.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montúfar, G. F., Lió, P., and Bronstein, M. M. Weisfeiler and lehman go topological: Message passing simplicial networks. In International Conference on Machine Learning, pp. 1026–1037, 2021b.

Bodnar, C., Giovanni, F. D., Chamberlain, B. P., Lió, P., and Bronstein, M. M. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In Advances in Neural Information Processing Systems, 2022.

Bresson, X. and Laurent, T. Residual gated graph convnets. arXiv preprint arXiv:1711.07553, 2017.

Brüel-Gabrielsson, R., Yurochkin, M., and Solomon, J. Rewiring with positional encodings for graph neural networks. arXiv preprint arXiv:2201.12674, 2022.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. In International Conference on Learning Representations, 2014.

Cai, C. and Wang, Y. A note on over-smoothing for graph neural networks.arXiv preprint arXiv:2006.13318, 2020.

Chandra, A. K., Raghavan, P., Ruzzo, W. L., Smolensky, R., and Tiwari, P. The electrical resistance of a graph captures its commute and cover times.computational complexity, 6(4):312–340, 1996.

Chen, Z., Li, L., and Bruna, J. Supervised community detection with line graph neural networks. InInternational conference on learning representations, 2020.

Chung, F. R. and Graham, F. C.Spectral graph theory. American Mathematical Soc., 1997.

Dasoulas, G., Lutzeyer, J. F., and Vazirgiannis, M. Learning parametrised graph shift operators. InInternational Conference on Learning Representations, 2021.

Deac, A., Lackenby, M., and Veličković, P. Expander graph propagation. InThe First Learning on Graphs Conference, 2022.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral ltering. InAdvances in neural information processing systems, volume 29, 2016.

Devriendt, K. and Lambiotte, R. Discrete curvature on graphs from the effective resistance.Journal of Physics: Complexity, 2022.

Di Giovanni, F., Luise, G., and Bronstein, M. Heterogeneous manifolds for curvature-aware graph embedding. In International Conference on Learning Representations Workshop on Geometrical and Topological Representation Learning 2022a.

Di Giovanni, F., Rowbottom, J., Chamberlain, B. P., Markovich, T., and Bronstein, M. M. Graph neural networks as gradient ows. arXiv preprint arXiv:2206.10991, 2022b.

Dör er, F., Simpson-Porco, J. W., and Bullo, F. Electrical networks and algebraic graph theory: Models, properties, and applicationsProceedings of the IEEE106(5):977–1005, 2018.

Ellens, W., Spieksma, F. M., Van Mieghem, P., Jamakovic, A., and Kooij, R. E. Effective graph resistanceLinear algebra and its applications435(10):2491–2506, 2011.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In International Conference on Machine Learning pp. 1263–1272. PMLR, 2017.

Goller, C. and Kuchler, A. Learning task-dependent distributed representations by backpropagation through structure. InProceedings of International Conference on Neural Networks (ICNN'96)volume 1, pp. 347–352. IEEE, 1996.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. InProceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005, volume 2, pp. 729–734. IEEE, 2005.

Gutteridge, B., Dong, X., Bronstein, M., and Di Giovanni, F. Drew: Dynamically rewired message passing with delay arXiv preprint arXiv:2305.08018, 2023.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. InAdvances in Neural Information Processing Systems pp. 1025–1035, 2017.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. Neural computation9(8):1735–1780, 1997.

Jegelka, S. Theory of graph neural networks: Representation and learning.arXiv preprint arXiv:2204.07697, 2022.

Karhadkar, K., Banerjee, P. K., and Montúfar, G. Fosr: First-order spectral rewiring for addressing oversquashing in gnns.arXiv preprint arXiv:2210.11790, 2022.

Kawaguchi, K. Deep learning without poor local minima. In Advances in neural information processing systems volume 29, 2016.

Kipf, T. N. and Welling, M. Semi-Supervised Classi cation with Graph Convolutional Networks. InInternational Conference on Learning Representations2017.

Klicpera, J., Weissenberger, S., and Günnemann, S. Diffusion improves graph learning. InAdvances in Neural Information Processing Systems2019.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. InAdvances in Neural Information Processing Systemsvolume 34, pp. 21618–21629, 2021.

Li, G., Muller, M., Thabet, A., and Ghanem, B. Deepgcns: Can gcns go as deep as cnns? InProceedings of the IEEE/CVF international conference on computer vision pp. 9267–9276, 2019.

Li, G., Müller, M., Ghanem, B., and Koltun, V. Training graph neural networks with 1000 layers. InInternational conference on machine learning pp. 6437–6449. PMLR, 2021.

Lovász, L. Random walks on graphsCombinatorics, Paul erdos is eighty2(1-46):4, 1993.

Ma, Z., Xuan, J., Wang, Y. G., Li, M., and Liò, P. Path integral based convolution and pooling for graph neural networks. InAdvances in Neural Information Processing Systemsvolume 33, pp. 16421–16433, 2020.

Mialon, G., Chen, D., Selosse, M., and Mairal, J. Graphit: Encoding graph structure in transformers.CoRRabs/2106.05667, 2021.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. InAAAI Conference on Arti cial Intelligencepp. 4602–4609. AAAI Press, 2019.

Nikolentzos, G., Dasoulas, G., and Vazirgiannis, M. k-hop graph neural networksNeural Networks130:195–205, 2020.

Nt, H. and Maehara, T. Revisiting graph neural networks: All we have is low-pass lters.arXiv preprint arXiv:1905.09550, 2019.

Ollivier, Y. Ricci curvature of metric spacesComptes Rendus Mathematique345(11):643–646, 2007.

Ollivier, Y. Ricci curvature of markov chains on metric spacesJournal of Functional Analysis256(3):810–864, 2009.

Pascanu, R., Mikolov, T., and Bengio, Y. On the dif culty of training recurrent neural networks. InInternational conference on machine learning pp. 1310–1318. PMLR, 2013.

Rampasek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. InAdvances in Neural Information Processing Systems2022.

Ruiz, L., Gama, F., and Ribeiro, A. Gated graph recurrent neural networksIEEE Transactions on Signal Processing, 68:6303–6318, 2020.

Rusch, T. K. and Mishra, S. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In International Conference on Learning Representations 2021a. URLhttps://openreview.net/forum?id=F3s69XzWOia .

Rusch, T. K. and Mishra, S. Unicornn: A recurrent model for learning very long time dependencies. InInternational Conference on Machine Learning, pp. 9168–9178. PMLR, 2021b.

Rusch, T. K., Chamberlain, B. P., Rowbottom, J., Mishra, S., and Bronstein, M. M. Graph-coupled oscillator networks. In International Conference on Machine Learning, 2022.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network modelIEEE transactions on neural networks, 20(1):61–80, 2008.

Sperduti, A. Encoding labeled graphs by labeling raam. In Advances in Neural Information Processing Systems volume 6, 1993.

Thomassen, C. Resistances and currents in in nite electrical networks.Journal of Combinatorial Theory, Series B, 49 (1):87–102, 1990.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. InInternational Conference on Learning Representations, 2022.

Velingker, A., Sinop, A. K., Ktena, I., Veličković, P., and Gollapudi, S. Af nity-aware graph networksarXiv preprint arXiv:2206.11941, 2022.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In International Conference on Learning Representations 2018.

Wang, G., Ying, R., Huang, J., and Leskovec, J. Multi-hop attention graph neural networkarXiv preprint arXiv:2009.14332, 2020.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. InInternational Conference on Machine Learning, pp. 5453–5462. PMLR, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? InInternational Conference on Learning Representations, 2019.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform badly for graph representation? InAdvances in Neural Information Processing Systems, volume 34, pp. 28877–28888, 2021.

Zhao, W., Wang, C., Han, C., and Guo, T. Analysis of graph neural networks with theory of markov chains. 2022.

A. General preliminaries

We first introduce important quantities and notations used throughout our proofs. We take a Graph G with nodes V and edges $E \subseteq V \times V$, to be simple, undirected, and connected. We let $|V| = n$ and write $[n] := \{1, \dots, n\}$. We denote the adjacency matrix by $A \in \mathbb{R}^{n \times n}$. We compute the degree of $v \in V$ by $d_v = \sum_u A_{vu}$ and write $D = \mathrm{diag}(d_1, \dots, d_n)$. One can take different normalizations of A, so we write $\mathbf{A} \in \mathbb{R}^{n \times n}$ for a Graph Shift Operator (GSO), i.e. an $n \times n$ matrix satisfying $A_{vu} \neq 0$ if and only if $(v, u) \in E$; typically, we have $\mathbf{A} \in \{A, D^{-1}A, D^{-1/2}AD^{-1/2}\}$. Finally, $d_G(v, u)$ is the shortest walk (geodesic) distance between nodes v and u.

Graph spectral properties: the eigenvalues. The (normalized) graph Laplacian is defined as $\mathbf{L} = I - D^{\frac{1}{2}}AD^{\frac{1}{2}}$. This is a symmetric, positive semi-definite operator on G. Its eigenvalues can be ordered as $0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$. The smallest eigenvalue λ_0 is always zero, with multiplicity given by the number of connected components of G (Chung & Graham, 1997). Conversely, the largest eigenvalue λ_{n-1} is always strictly smaller than 2 whenever the graph is not bipartite. Finally, we recall that the smallest, positive, eigenvalue λ_1 is known as the *spectral gap*. In several of our proofs we rely on this quantity to provide convergence rates. We also recall that the spectral gap is related to the Cheeger constant – introduced in Definition 2.2 – of G via the Cheeger inequality:

$$2h_{\mathrm{Cheeg}} \quad {}_1 > \frac{h^2_{\mathrm{Cheeg}}}{2}. \tag{8}$$

Graph spectral properties: the eigenvectors. Throughout the appendix, we let $\{\mathbf{v}_i\}$ be a family of orthonormal eigenvectors of $\mathbf{L}$. In particular, we note that the eigenspace associated with λ_0 represents the space of signals that respect the graph topology the most (i.e. the smoothest signals), so that we can write $\mathbf{v}_0(v) = \sqrt{d_v}/\sqrt{2|E|}$, for any $v \in V$.

As common, from now on we assume that the graph is bipartite, so that $\lambda_{n-1} < 2$. We let $H^{(0)} \in \mathbb{R}^{n \times p}$ be the matrix representation of node features, with p denoting the hidden dimension. Features of node v produced by layer t of an MPNN are denoted by $h_v^{(t)}$ and we write their components as $(h_v^{(t)}) := h_v^{(t),\ell}$, for $\ell \in [p]$.

Einstein summation convention. To ease notations when deriving the bounds on the Jacobian, in the proof below we often rely on Einstein summation convention, meaning that, unless specified otherwise, we always sum across repeated indices: for example, when we write terms like $x_k y_k$, we are tacitly omitting the symbol $\sum_k$.

B. Proofs of Section 3

In this Section we demonstrate the results in Section 3. In fact, we derive a sensitivity bound far more general than Theorem 3.2 that, in particular, extends to MPNNs that can stack multiple layers (MLPs) in the aggregation phase. We introduce a class of MPNNs of the form:

$$h_v^{(t)} = \mathrm{up}^{(t)}\left(\mathrm{rs}^{(t)}(h_v^{(t-1)}) + \mathrm{mp}^{(t)}\Big(\sum_u A_{vu}\, h_u^{(t-1)}\Big)\right) \tag{9}$$

for learnable update, residual, and message-passing maps $\mathrm{up}^{(t)}, \mathrm{rs}^{(t)}, \mathrm{mp}^{(t)} : \mathbb{R}^p \to \mathbb{R}^p$. Note that Eq. (9) includes common MPNNs like GCN (Kipf & Welling, 2017), SAGE (Hamilton et al., 2017), and GIN (Xu et al., 2019), where $\mathbf{A}$ is $D^{-1/2}AD^{-1/2}$, $D^{-1}A$ and A, respectively. An MPNN usually has Lipschitz maps, with Lipschitz constants typically depending on regularization of the weights to promote generalization. We say that an MPNN as in Eq. (9) is (c_{up}, c_{rs}, c_{mp})-regular, if for $t \in [m]$ and $\ell \in [p]$, we have

$$\|\nabla(\mathrm{up}^{(t)})_\ell\|_{L_1} \leq c_{up}, \quad \|\nabla(\mathrm{rs}^{(t)})_\ell\|_{L_1} \leq c_{rs}, \quad \|\nabla(\mathrm{mp}^{(t)})_\ell\|_{L_1} \leq c_{mp}.$$

As in Xu et al. (2018); Topping et al. (2022), we study the propagation of information in MPNNs via the Jacobian of node features after m layers. A small derivative of $h_v^{(m)}$ with respect to $h_u^{(0)}$ means that – at the first-order – the representation at node v is mostly insensitive to the information contained at u (e.g. its atom type, if G is a molecule).

Theorem B.1. Given a (c_{up}, c_{rs}, c_{mp})-regular MPNN for m layers and nodes $v, u \in V$, we have

$$\left\|\frac{\partial h_v^{(m)}}{\partial h_u^{(0)}}\right\|_{L_1} \leq c_{up}^m \left((c_{rs} I + c_{mp}\mathbf{A})^m\right)_{vu}. \tag{10}$$

Proof. We prove the result above by induction on the number of layers m. We fix x ; 2 [p]. In the case of $m = 1$, we get (omitting to write the arguments where we evaluate the maps, and using the Einstein summation convention over repeated indices):

$$\frac{@h_v^{(1)\,;}}{@h_u^{(0)\,;}} = @_p up^{(0)\,;} \quad @_r s^{(0)\,;p} \frac{@h_v^{(0)\,;r}}{@h_u^{(0)\,;}} + @_q mp^{(0)\,;p} A_{vz} \frac{@h_z^{(0)\,;q}}{@h_u^{(0)\,;}} \quad ;$$

which can be readily reduced to

$$\frac{@h_v^{(1)\,;}}{@h_u^{(0)\,;}} = @_p up^{(0)\,;} \quad @_r s^{(0)\,;p}{}_{vu} + @_m p^{(0)\,;p} A_{vu} \qquad c_{up} (c_{rs} I + c_{mp} A)_{vu} ;$$

thanks to the Lipschitz bounds on the MPNN, which con rms the case of a single layer (i.e. $m = 1$). We now assume the bound to be satis ed for m layers and use induction to derive

$$\begin{aligned}
\frac{@h_v^{(m+1)\,;}}{@h_u^{(0)\,;}} &= @_p up^{(m)\,;} \quad @_r s^{(m)\,;p} \frac{@h_v^{(m)\,;r}}{@h_u^{(0)\,;}} + @_q mp^{(m)\,;p} A_{vz} \frac{@h_z^{(m)\,;q}}{@h_u^{(0)\,;}} \\
&\quad @_p up^{(m)\,;} \quad @_r s^{(m)\,;p} \quad c_{up}^m ((c_{rs} I + c_{mp} A)^m)_{vu} + @_m p^{(m)\,;p} A_{vz} \quad c_{up}^m ((c_{rs} I + c_{mp} A)^m)_{zu} \\
&\quad @_p up^{(m)\,;} \quad c_{rs} \quad c_{up}^m ((c_{rs} I + c_{mp} A)^m)_{vu} + c_{mp} A_{vz} \quad c_{up}^m ((c_{rs} I + c_{mp} A)^m)_{zu} \\
&\quad c_{up}^{m+1} \quad c_{rs} ((c_{rs} I + c_{mp} A)^m)_{vu} + c_{mp} A_{vz} ((c_{rs} I + c_{mp} A)^m)_{vu} \\
&= c_{up}^{m+1} \quad (c_{rs} I + c_{mp} A)^{m+1}{}_{vu} ;
\end{aligned}$$

where we have again using the Lipschitz bounds on the maps up, rs, mp. This completes the induction argument. □

From now on we focus on the class of MPNN adopted in the main document, whose layer we report below for convenience:

$$h_v^{(t+1)} = \quad c_r W_r^{(t)} h_v^{(t)} + c_a W_a^{(t)} \sum_u A_{vu} h_u^{(t)} \quad ;$$

We can adapt easily the general argument to derive Theorem 3.2.

Proof of Theorem 3.2. One can follow the steps in the proof of Theorem B.1 and, again, proceed by induction. The case $m = 1$ is straightforward, so we move to the inductive step and assume the bound to hold for m arbitrary. Given $;$ 2 [p], we have

$$\begin{aligned}
\frac{@h_v^{(m+1)\,;}}{@h_u^{(0)\,;}} &\quad j \quad {}^0 j \quad c_r \quad (W_r)^{(m)} \quad \frac{@h_v^{(m)\,;}}{@h_u^{(0)\,;}} + c_a \quad (W_a)^{(m)} \quad A_{vz} \frac{@h_z^{(m)\,;}}{@h_u^{(0)\,;}} \quad ! \\
&\quad c \quad w \quad c_r \frac{@h_v^{(m)}}{@h_u^{(0)}}_{L_1} + c_a A_{vz} \frac{@h_z^{(m)}}{@h_u^{(0)}}_{L_1} \\
&\quad c \quad w (c \quad wp)^m ((c_r I + c_a A)^m)_{vu} + c_a A_{vz} ((c_r I + c_a A)^m)_{zu}) \\
&\quad c \quad w (c \quad wp)^m \quad (c_r I + c_a A)^{m+1}{}_{vu} :
\end{aligned}$$

We can nally sum over $$ on the left and conclude the proof (this will generate an extra factor on the right hand side).

□

C. Proofs of Section 4

Convention: From now on we always let $A = D^{-1=2} A D^{-1=2}$. The bounds in this Section extend easily to $D^{-1} A$ in light of the similarity of the two matrices since $A^k = D^{1=2} D^{-1} A^k D^{-1=2}$. For the unnormalized matrix A instead, things are slightly more subtle. In principle, this matrix is not normalized, and in fact, the entry $(A^k)_{vu}$ coincides with the number of walks from v to u of length k. In general, this will not lead to bounds decaying exponentially with the distance. However, if

we go in expectation over the computational graph as in Xu et al. (2018), Appendix A of Topping et al. (2022) and Section 5, one finds that nodes at smaller distance will still have sensitivity exponentially larger than nodes at large distance. This is also confirmed by our Graph Transfer synthetic experiments, where GIN struggles with long-range dependencies (in fact, even slightly more than GCN, which uses the symmetrically normalized adjacency $\mathbf{A}$).

We prove a sharper bound for Eq. (4) which contains Theorem 4.1 as a particular case.

Theorem C.1. *Given an* MPNN *as in Eq. (2), let $v; u \in V$ be at distance r. Let c be the Lipschitz constant of ϕ, w the maximal entry-value over all weight matrices, $d_{\min}$ be the minimal degree, and $\ell(v; u)$ be the number of walks from v to u of maximal length ℓ. For any $0 \leq k < r$, we have*

$$\left\| \frac{\partial h_v^{(r+k)}}{\partial h_u^{(0)}} \right\|_{L_1} \leq \ell_{r+k}(v; u)(c\phi(c_r + c_a)w p(k+1))^k \left(\frac{2c\phi w p c_a}{d_{\min}} \right)^r. \tag{11}$$

Proof. Fix $v; u \in V$ as in the statement and let $0 \leq k < r$. We can use the sensitivity bounds in Theorem 3.2 and write that

$$\left\| \frac{\partial h_v^{(r+k)}}{\partial h_u^{(0)}} \right\|_{L_1} \leq (c\phi w p)^{r+k} \left(c_r \mathrm{I} + c_a \mathbf{A} \right)^{r+k}_{vu} = (c\phi w p)^{r+k} \sum_{i=0}^{r+k} \binom{r+k}{i} c_r^{r+k-i} c_a^i (\mathbf{A}^i)_{vu}.$$

Since nodes $v; u$ are at distance r, the first r terms of the sum above vanish. Once we recall that $\mathbf{A} = D^{-1/2} A D^{-1/2}$, we can bound the polynomial in the previous equation by

$$\begin{aligned}
\sum_{i=0}^{r+k} \binom{r+k}{i} c_r^{r+k-i} c_a^i (\mathbf{A}^i)_{vu} &= \sum_{i=r}^{r+k} \binom{r+k}{i} c_r^{r+k-i} c_a^i (\mathbf{A}^i)_{vu} \leq \ell_{r+k}(v; u) \sum_{i=r}^{r+k} \binom{r+k}{i} c_r^{r+k-i} \left(\frac{c_a}{d_{\min}} \right)^i \\
&= \ell_{r+k}(v; u) \sum_{q=0}^{k} \binom{r+k}{r+q} c_r^{k-q} \left(\frac{c_a}{d_{\min}} \right)^{r+q} \\
&= \ell_{r+k}(v; u) \left(\frac{c_a}{d_{\min}} \right)^r \sum_{q=0}^{k} \binom{r+k}{r+q} c_r^{k-q} \left(\frac{c_a}{d_{\min}} \right)^q.
\end{aligned}$$

We can now provide a simple estimate for

$$\begin{aligned}
\binom{r+k}{r+q} &= \frac{(r+k)(r-1+k)\cdots(1+k)}{(r+q)(r-1+q)\cdots(1+q)} \binom{k}{q} \leq \frac{(r+k)(r-1+k)\cdots(1+k)}{r!} \binom{k}{q} \\
&\leq \left(1 + \frac{k}{r}\right)\cdots(1+k) \binom{k}{q} \leq \left(1 + \frac{k}{k+1}\right)^r (1+k)^k \binom{k}{q}.
\end{aligned}$$

Accordingly, the polynomial above can be expanded as

$$\begin{aligned}
\sum_{i=0}^{r+k} \binom{r+k}{i} c_r^{r+k-i} c_a^i (\mathbf{A}^i)_{vu} &\leq \ell_{r+k}(v; u) \left(1 + \frac{k}{k+1}\right)^r (1+k)^k \left(\frac{c_a}{d_{\min}} \right)^r \sum_{q=0}^{k} \binom{k}{q} c_r^{k-q} \left(\frac{c_a}{d_{\min}} \right)^q \\
&= \ell_{r+k}(v; u) \left(\frac{(1+k)^2}{2k+1} \right)^k \left(c_r + \frac{c_a}{d_{\min}} \right)^k \left(1 + \frac{k}{k+1} \right)^r \left(\frac{c_a}{d_{\min}} \right)^r \\
&\leq \ell_{r+k}(v; u) \left(\frac{(1+k)^2}{2k+1} \right)^k \left(c_r + \frac{c_a}{d_{\min}} \right)^k \left(\frac{2c_a}{d_{\min}} \right)^r.
\end{aligned}$$

We can then put all the ingredients together, and write the bound

$$\left\| \frac{\partial h_v^{(r+k)}}{\partial h_u^{(0)}} \right\|_{L_1} \leq \alpha_{r+k}(v,u)\,(c\ \sigma w)^{r+k} \left(\frac{(1+k)^2}{2k+1}\right)^k \left(c_r + \frac{c_a}{d_{\min}}\right)^k \left(\frac{2c_a}{d_{\min}}\right)^r$$
$$= \alpha_{r+k}(v,u)\ \left(c\ \sigma w \frac{(1+k)^2}{2k+1}\right)^k \left(c_r + \frac{c_a}{d_{\min}}\right)^k \left(\frac{2c\ \sigma w c_a}{d_{\min}}\right)^r$$
$$\leq \alpha_{r+k}(v,u)\,(c\ (c_r + c_a)\,\sigma w(1+k))^k \left(\frac{2c\ \sigma w c_a}{d_{\min}}\right)^r;$$

which completes the proof. We finally note that this also proves Theorem 4.1. □

C.1. Vanishing gradients result

We now report and demonstrate a more explicit version of Theorem 4.2.

Theorem C.2 (**Vanishing gradients**). *Consider an* MPNN *as in Eq. (2) for m layers with a quadratic loss L. Assume that (i) has Lipschitz constant c and $(0) = 0$, and (ii) that all weight matrices have spectral norm bounded by > 0. Given any weight entering a layer k, there exists a constant $C > 0$ independent of m, such that*

$$\left\| \frac{\partial L}{\partial} \right\| \leq C\,(c\ (c_r + c_a))^{m-k}\,(1 + (c\ (c_r + c_a))^m); \tag{12}$$

where $\|H^{(0)}\|_F$ is the Frobenius norm of the input node features.

Proof. Consider a quadratic loss L of the form

$$L(H^{(m)}) = \frac{1}{2} \sum_{v \in V} \|h_v^{(m)} - y_v\|^2;$$

and we let Y represent the node ground-truth values. Given a weight entering layer $k < m$, we can write the gradient of the loss as

$$\frac{\partial L(H^{(m)})}{\partial} = \sum_{v,u \in V} \sum_{[p]} \frac{\partial L}{\partial h_v^{(m),}} \frac{\partial h_v^{(m),}}{\partial h_u^{(k),}} \frac{\partial h_u^{(k),}}{\partial};$$

Once we fix k, the term $\|\partial h_u^{(k),}/\partial\|$ is independent of m and we can bound it by some constant C. Since we have a quadratic loss, to bound $\|\partial L/\partial h_v^{(m),}\|$, it suffices to bound the solution of the MPNN after m layers. First, we use the Kronecker product formalism to rewrite the MPNN-update in matricial form as

$$H^{(m)} = \sigma\left(c_r^{(m)} I + c_a W^{(m)} \mathbf{A} H^{(m-1)}\right); \tag{13}$$

Thanks to the Lipschitzness of and the requirement $(0) = 0$, we derive

$$\|H^{(m)}\|_F \leq c\ \|c_r^{(m)} I + c_a W^{(m)} \mathbf{A}\|_2 \|H^{(m-1)}\|_F;$$

where F indicates the Frobenius norm. Since the largest singular value of $B C$ is bounded by the product of the largest singular values, we deduce that – recall that the largest eigenvalue of $\mathbf{A} = D^{-1/2} A D^{-1/2}$ is 1:

$$\|H^{(m)}\|_F \leq c\ (c_r + c_a)\|H^{(m-1)}\|_F \leq (c\ (c_r + c_a))^m \|H^{(0)}\|_F; \tag{14}$$

which affords a control of the gradient of the loss w.r.t. the solution at the final layer being the loss quadratic. We then find

$$\begin{aligned}
\frac{@L(\mathbf{H}^{(m)})}{@} &\quad C \sum_{v;u2\mathcal{V}\ ;\ 2[p]} \frac{@L}{@h_v^{(m);}} \frac{@h_v^{(m);}}{@h_u^{(k);}} \\
&\quad C \sum_{v;u2\mathcal{V}\ \ 2[p]} \frac{@L}{@\mathrm{h}_v^{(m)}} \frac{@\mathrm{h}_v^{(m)}}{@h_u^{(k);}} \\
&\quad C \sum_{v;u2\mathcal{V}\ \ 2[p]} k\mathbf{H}^{(m)}k_F + k\mathbf{Y}k_F \frac{@\mathrm{h}_v^{(m)}}{@h_u^{(k);}} \\
&\quad C \sum_{v;u2\mathcal{V}\ \ 2[p]} (c \quad (c_\mathrm{r} + c_\mathrm{a}))^m k\mathbf{H}^{(0)}k_F + k\mathbf{Y}k_F \frac{@\mathrm{h}_v^{(m)}}{@h_u^{(k);}}
\end{aligned} \tag{15}$$

where in the last step we have used Eq. (14). We now provide a **new** bound on the sensitivity – *differently from the analysis in earlier Sections, here we no longer account for the topological information depending on the choice of $v; u$ given that we need to integrate over all possible pairwise contributions to compute the gradient of the loss.* The idea below, is to apply the Kronecker product formalism to derive a single operator in the tensor product of feature and graph space acting on the Jacobian matrix – this allows us to derive much sharper bounds. Note that, once we fix a node u and a $\ 2\ [p]$, we can write

$$\begin{aligned}
\frac{@\mathbf{H}^{(m)}}{@h_u^{(k);}}^2 &\quad \sum_{v2\mathcal{V}\ \ 2[p]} c^2 \ c_\mathrm{r} \ ^{(m)} \frac{@h_v^{(m\ 1);}}{@h_u^{(k);}} + c_\mathrm{a}\mathbf{W}^{(m)}\mathbf{A}_{vz} \frac{@h_z^{(m\ 1);}}{@h_u^{(k);}}^2 \\
&= c^2 \sum_{v2\mathcal{V}\ \ 2[p]} c_\mathrm{r} \ ^{(m)} \quad \mathbf{I} + c_\mathrm{a}\mathbf{W}^{(m)} \quad \mathbf{A} \frac{@\mathbf{H}^{(m\ 1)}}{@h_u^{(k);}}^2_{v;} \\
&\quad c^2 k c_\mathrm{r} \ ^{(m)} \quad \mathbf{I} + c_\mathrm{a}\mathbf{W}^{(m)} \quad \mathbf{A}k_2^2 \frac{@\mathbf{H}^{(m\ 1)}}{@h_u^{(k);}}^2_F
\end{aligned}$$

meaning that

$$\frac{@\mathbf{H}^{(m)}}{@h_u^{(k);}} \quad (c \quad (c_\mathrm{r} + c_\mathrm{a}))^{m\ \ k};$$

where we have used that (i) the largest singular value of the weight matrices is , (ii) that the largest eigenvalue of $c_\mathrm{r}\mathbf{I} + c_\mathrm{a}\mathbf{A}$ is $c_\mathrm{r} + c_\mathrm{a}$ (as follows from $\mathbf{A} = \mathbf{I}$, and the spectral analysis of), (iii) that $k@\mathbf{H}^{(k)}=@h_u^{(k);}\ k = 1$. Once we absorb the term $k\mathbf{Y}k$ in the constant C in (15), we conclude the proof.

□

D. Proofs of Section 5

In this Section we consider the convolutional family of MPNN in Eq. (7). Before we prove the main results of this Section, we comment on the main assumption on the nonlinearity and formulate it more explicitly. Let us take $k < m$. When we compute the sensitivity of $\mathrm{h}_v^{(m)}$ to $\mathrm{h}_u^{(k)}$, we obtain a sum of different terms over all possible paths from v to u of length $m \quad k$. In this case, the derivative of ReLU acts as a Bernoulli variable evaluated along all these possible paths. Similarly to Kawaguchi (2016); Xu et al. (2018), we require the following:

Assumption D.1. Assume that all paths in the computation graph of the model are activated with the same probability of success . When we take the expectation $\mathbb{E}[@\mathrm{h}_v^{(m)}=@\mathrm{h}_u^{(k)}]$, we mean that we are taking the average over such Bernoulli variables.

Thanks to Assumption D.1, we can follow the very same argument in the proof of Theorem 1 in (Xu et al., 2018) to derive

$$\mathbb{E}\left[\frac{@\mathrm{h}_v^{(m)}}{@\mathrm{h}_u^{(k)}}\right] = \prod_{s=k+1}^{m} \mathbf{W}^{(s)}(\mathbf{S}_{\mathrm{r};\mathrm{a}}^{m\ \ k})_{vu};$$

We can now proceed to prove the relation between sensitivity analysis and access time.

Proof of Theorem 5.3. Under Assumption D.1, we can write the term $J_k^{(m)}(v;u)$ as

$$\mathbb{E}\left[J_k^{(m)}(v;u)\right] = \mathbb{E}\left[\frac{1}{d_v}\frac{\partial h_v^{(m)}}{\partial h_v^{(k)}} \quad p\frac{1}{d_v d_u}\frac{\partial h_v^{(m)}}{\partial h_u^{(k)}}\right]$$
$$= \sum_{s=k+1}^{m} W^{(s)} \quad \frac{1}{d_v}(S_{r;a}^{m\ k})_{vv} \quad p\frac{1}{d_v d_u}(S_{r;a}^{m\ k})_{vu} \ :$$

Since $S_{r;a} = c_r I + c_a D^{-1=2} A D^{-1=2}$, we can rely on the spectral decomposition of the graph Laplacian – see the conventions and notations introduced in Appendix A – to write

$$S_{r;a} = \sum_{\ell=0}^{n-1} (c_r + c_a(1 - \lambda_\ell))\, u_\ell u_\ell^\top;$$

where we recall that $\lambda_\ell = 1 - \lambda_\ell$. We can then bound (in **expectation**) the Jacobian obstruction by

$$O^{(m)}(v;u) = \sum_{k=0}^{m} |J_k^{(m)}(v;u)| \sum_{k=0}^{m} \quad m\ k \quad \frac{1}{d_v}(S_{r;a}^{m\ k})_{vv} \quad p\frac{1}{d_v d_u}(S_{r;a}^{m\ k})_{vu}$$
$$= \sum_{k=0}^{m} \quad m\ k \quad \frac{1}{d_v}(S_{r;a}^{m\ k})_{vv} \quad p\frac{1}{d_v d_u}(S_{r;a}^{m\ k})_{vu}$$
$$= \sum_{k=0}^{m} \quad m\ k \sum_{\ell=0}^{n-1} c_r + c_a(1 - \lambda_\ell)^{\quad m\ k} \quad \frac{u_\ell^2(v)}{d_v} \quad p\frac{u_\ell(v)\, u_\ell(u)}{d_u d_v}$$
$$= \sum_{\ell=0}^{n-1} \sum_{k=0}^{m} \quad m\ k\, (c_r + c_a(1 - \lambda_\ell))^{m\ k} \quad \frac{u_\ell^2(v)}{d_v} \quad p\frac{u_\ell(v)\, u_\ell(u)}{d_u d_v}$$
$$= \sum_{\ell=1}^{n-1} \sum_{k=0}^{m} \quad (\quad (c_r + c_a(1 - \lambda_\ell)))^{m\ k} \quad \frac{u_\ell^2(v)}{d_v} \quad p\frac{u_\ell(v)\, u_\ell(u)}{d_u d_v} \ ;$$

where in the last equality we have used that $u_0(v) = \sqrt{d_v}=(2|E|)$ for each $v \in V$. We can then expand the geometric sum by using our assumption $(c_r + c_a) = 1$ and write

$$O^{(m)}(v;u) \quad \sum_{\ell=1}^{n-1} \frac{1 \quad (\quad (c_r + c_a(1 - \lambda_\ell)))^{m+1}}{1 \quad (c_r + c_a) + \quad c_a\ \lambda_\ell} \quad \frac{u_\ell^2(v)}{d_v} \quad p\frac{u_\ell(v)\, u_\ell(u)}{d_u d_v} \ :$$

Since $(c_r + c_a) = 1$, we can simplify the lower bound as

$$O^{(m)}(v;u) \quad \sum_{\ell=1}^{n-1} \frac{1}{c_a\ \lambda_\ell} \quad \frac{u_\ell^2(v)}{d_v} \quad p\frac{u_\ell(v)\, u_\ell(u)}{d_u d_v} \qquad \sum_{\ell=1}^{n-1} \frac{(\quad (c_r + c_a(1 - \lambda_\ell)))^{m+1}}{c_a\ \lambda_\ell} \quad \frac{u_\ell^2(v)}{d_v} \quad p\frac{u_\ell(v)\, u_\ell(u)}{d_u d_v} \ :$$

By Lovász (1993, Theorem 3.1), the first term is equal to $|(\ c_a)^{-1} t(u;v)=2|E||$ which is a positive number. Concerning the second term, we recall that the eigenvalues of the graph Laplacian are ordered from smallest to largest and that u_ℓ is a unit vector, so

$$O^{(m)}(v;u) \quad \frac{t(u;v)}{c_a} \frac{}{2|E|} \quad \frac{(1 \quad c_a\ )^{m+1}}{c_a\ \lambda_1} \frac{n \quad 1}{d_{\min}};$$

with λ_1 such that $|1 \quad \lambda_1| = \max_{\lambda_\ell > 0} |1 \quad \lambda_\ell|$ which completes the proof.

□

Proof of Theorem 5.5. We follow the same strategy used in the proof of Theorem 5.3. Under Assumption D.1, we can write

the term $\mathsf{J}_k^{(m)}(v;u)$ as

$$\mathsf{E}\left[\mathsf{J}_k^{(m)}(v;u)\right] = \mathsf{E}\left[\frac{1}{d_v}\frac{\partial \mathsf{h}_v^{(m)}}{\partial \mathsf{h}_v^{(k)}} \quad \frac{1}{\sqrt{d_v d_u}}\frac{\partial \mathsf{h}_v^{(m)}}{\partial \mathsf{h}_u^{(k)}} + \frac{1}{d_u}\frac{\partial \mathsf{h}_u^{(m)}}{\partial \mathsf{h}_u^{(k)}} \quad \frac{1}{\sqrt{d_v d_u}}\frac{\partial \mathsf{h}_u^{(m)}}{\partial \mathsf{h}_v^{(k)}}\right]$$
$$= \sum_{s=k+1}^{m} \mathsf{W}^{(s)} \quad \frac{1}{d_v}(\mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k})_{vv} + \frac{1}{d_u}(\mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k})_{uu} \quad 2(\mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k})_{vu}$$

where we have used the symmetry of $\mathbf{S}_{\mathsf{r};\mathsf{a}}$. We note that the term within brackets can be equivalently reformulated as

$$\frac{1}{d_v}(\mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k})_{vv} + \frac{1}{d_u}(\mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k})_{uu} \quad 2(\mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k})_{vu} = \left\langle \frac{\mathsf{e}_v}{\sqrt{d_v}} \quad \frac{\mathsf{e}_u}{\sqrt{d_u}}; \mathbf{S}_{\mathsf{r};\mathsf{a}}^{m\ k}\left[\frac{\mathsf{e}_v}{\sqrt{d_v}} \quad \frac{\mathsf{e}_u}{\sqrt{d_u}}\right]\right\rangle$$

where e_v is the vector with 1 at entry v, and zero otherwise. In particular, we note an **important fact**: since, by assumption, $c_{\mathsf{r}} \quad c_{\mathsf{a}}$ and $\quad_{n\ 1} < 2$, whenever G is not bipartite, we derive that $\mathbf{S}_{\mathsf{r};\mathsf{a}}$ is a *positive definite operator*. We can then bound (in **expectation**) the Jacobian obstruction by

$$\mathcal{O}^{(m)}(v;u) = \sum_{k=0}^{m} \left|\mathsf{J}_k^{(m)}(v;u)\right| \sum_{k=0}^{m} \quad {}^{m\ k} \prod_{\ell=0}^{n-1} (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_\ell))^{m\ k} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2$$
$$= \sum_{\ell=0}^{n-1} \sum_{k=0}^{m} \quad {}^{m\ k} (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_\ell))^{m\ k} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2$$
$$= \sum_{\ell=1}^{n-1} \sum_{k=0}^{m} \quad {}^{m\ k} (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_\ell))^{m\ k} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2;$$

where in the last equality we have used that $\quad_0(v) = \sqrt{d_v}/(2|\mathsf{E}|)$. We can then expand the geometric sum by using our assumption $(c_{\mathsf{r}} + c_{\mathsf{a}}) \quad 1$ and write

$$\mathcal{O}^{(m)}(v;u) \quad \sum_{\ell=1}^{n-1} \frac{1 \quad (\quad (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_\ell)))^{m+1}}{1 \quad (c_{\mathsf{r}} + c_{\mathsf{a}}) + \quad c_{\mathsf{a}} \quad \lambda_\ell} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2$$
$$\sum_{\ell=1}^{n-1} \frac{}{c_{\mathsf{a}} \quad \lambda_\ell} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2$$
$$= \frac{}{c_{\mathsf{a}}} \sum_{\ell=1}^{n-1} \frac{1}{\quad \lambda_\ell} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2$$
$$= \frac{}{c_{\mathsf{a}}} \mathsf{Res}(v;u)$$

where in the last step we used the spectral characterization of the effective resistance derived in Lovász (1993) – which was also leveraged in Arnaiz-Rodríguez et al. (2022) to derive a novel rewiring algorithm. Since by Chandra et al. (1996) we have $2\mathsf{Res}(v;u)|\mathsf{E}| = \quad(v;u)$, this completes the proof of the upper bound. The lower bound case follows by a similar argument. In fact, one arrives at the estimate

$$\mathcal{O}^{(m)}(v;u) \quad \sum_{\ell=1}^{n-1} \frac{1 \quad (\quad (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_\ell)))^{m+1}}{1 \quad (c_{\mathsf{r}} + c_{\mathsf{a}}) + \quad c_{\mathsf{a}} \quad \lambda_\ell} \quad \left\langle \frac{p_\ell(v)}{\sqrt{d_v}} \quad \frac{p_\ell(u)}{\sqrt{d_u}} \right\rangle^2.$$

We derive

$$1 \quad (\quad (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_\ell)))^{m+1} \quad 1 \quad (\quad (c_{\mathsf{r}} + c_{\mathsf{a}}(1 \quad \lambda_{\max})))^{m+1};$$

where $|1 \quad \lambda_{\max}| = \max_{\ell>0}|1 \quad \lambda_\ell|$. Next, we also find that

$$\frac{1}{1 \quad (c_{\mathsf{r}} + c_{\mathsf{a}}) + \quad c_{\mathsf{a}} \quad \lambda_\ell} \quad \frac{}{c_{\mathsf{a}} \quad \lambda_\ell} \quad (\quad) \quad \lambda_\ell \quad \frac{1}{1} \quad \frac{1 \quad (c_{\mathsf{r}} + c_{\mathsf{a}})}{c_{\mathsf{a}}}.$$

Since the eigenvalues are ordered from smallest to largest, it suffices that

$$1 \quad \frac{1}{1} \frac{1}{c_a} \frac{(c_r + c_a)}{} \quad ( ) \qquad {}_G := \frac{1}{1 + \frac{1}{c_a}(c_r + c_a)}.$$

This completes the proof. □

We emphasize that without the degree normalization, the bound would have an extra-term (potentially diverging with the number of layers) and simply proportional to the degrees of nodes $v; u$. The extra-degree normalization is off-setting this uninteresting contribution given by the steady state of the Random Walks.

E. Graph Transfer

The goal in the three graph transfer tasks - Ring, CrossedRing, and CliquePath - is for the MPNN to 'transfer' the features contained at the target node to the source node. Ring graphs are cycles of size n, in which the target and source nodes are placed at a distance of $bn{=}2c$ from each other. CrossedRing graphs are also cycles of size n, but now include 'crosses' between the auxiliary nodes. Importantly, the added edges do not reduce the minimum distance between the source and target nodes, which remains $bn{=}2c$. CliquePath graphs contain a $bn{=}2c$-clique and a path of length $bn{=}2c$. The source node is placed on the clique and the target node is placed at the end of the path. The clique and path are connected in such a way that the distance between the source and target nodes is $bn{=}2c + 1$, in other words the source node requires one hop to gain access the path.

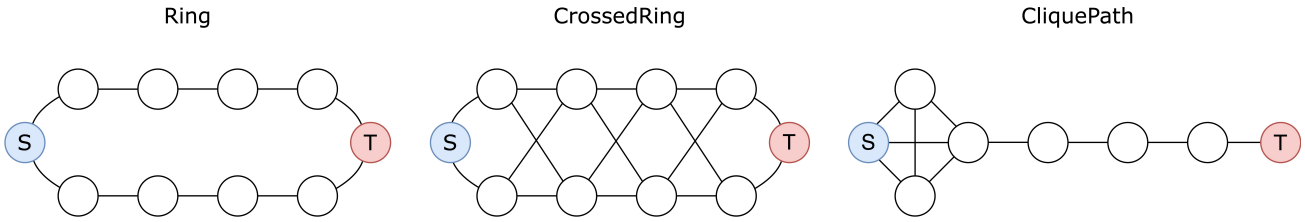

Figure 4. Topological structure of RingTransfer, CrossedRingTransfer, and CliquePath. The nodes marked with an S are the source nodes, while the nodes with a T are the target nodes. All tasks are shown for a distance between the source and target nodes of $r = 5$.

Figure 4 shows examples of the graphs contained in the Ring, CrossedRing, and CliquePath tasks, for when the distance between the source and target nodes is $r = 5$. In our experiments we take as input dimension $p = 5$ and assign to the target node a randomly one-hot encoded feature vector - for this reason the random guessing baseline obtains 20% accuracy. The source node is assigned a vector of all 0s and the auxiliary nodes are instead assigned vectors of 1s. Following (Bodnar et al., 2021a), we generate 5000 graphs for the training set and 500 graphs for the test set for each task. In our experiments, we report the mean accuracy over the test set. We train for 100 epochs, with depth of the MPNN equal to the distance between the source and target nodes r. Unless specified otherwise, we set the hidden dimension to 64. During training and testing, we apply a mask over all the nodes in order to focus only on the source node to compute losses and accuracy scores.

F. Signal Propagation

In this section we provide synthetic experiments on the PROTEINS, NCI1, PTC, ENZYMES datasets with the aim to provide empirical evidence to the fact that the total effective resistance of a graph, $\mathrm{Res}_G = \sum_{v;u} \mathrm{Res}(v;u)$ (Ellens et al., 2011), is related to the ease of information propagation in an MPNN. The experiment is designed as follows: we first fix a source node $v \in V$ assigning it a p-dimensional unitary feature vector, and assigning the rest of the nodes zero-vectors. We then consider the quantity

$$h^{(m)} = \frac{1}{p \max_{u \in V} d_G(v;u)} \sum_{f=1}^{p} \sum_{u \in V} \frac{h_u^{(m);f}}{k h_u^{(m);f} k} d_G(v;u);$$

to be the amount of signal (or 'information') that has been propagated through G by an MPNN with m layers. Intuitively, we measure the (normalized) propagation distance over G, and average it over all the p output channels. By propagation

Figure 5. Decay of the amount of information propagated through the graphs w.r.t. the normalized total effective resistance (commute time) for: (a) PROTEINS; (b) NCI1; (c) PTC; (d) ENZYMES. For each dataset we report the decay for: (i) GIN (top-left); (ii) Sage (top-right), (iii) GCN (bottom-left) and (iv) GAT(bottom-right).

distance we mean the average distance to which the initial 'unit mass' has been propagated to - a larger propagation distance means that on average the unit mass has travelled further w.r.t. to the source node. The goal is to show that $h^{(m)}$ is *inversely proportional to* $\mathsf{Res}_{\mathsf{G}}$. In other words, we expect graphs with *lower* total effective resistance to have a *larger* propagation distance. The experiment is repeated for each graph G that belongs to the dataset D. We start by randomly choosing the source node v, we then set $\mathbf{h}_v$ to be an arbitrary feature vector with unitary mass (i.e. $k\mathbf{h}_v k_{L_1} = 1$) and assigning the zero-vector to all other nodes (i.e. $\mathbf{h}_u = 0;\ u \neq v$). We use MPNNs with a number of layers m close to the average diameter of the graphs in the dataset, input and hidden dimensions $p = 5$ and ReLU activations. In particular, we estimate the resistance of G by sampling 10 nodes with uniform probability for each graph and we report $h^{(m)}$ accordingly. In Figure 5 we show that MPNN*s are able to propagate information further when the effective resistance is low*, validating empirically the impact of the graph topology on over-squashing phenomena. It is worth to emphasize that in this experiment, the parameters of the MPNN are randomly initialized and there is no underlying training task. This implies that in this setup we are isolating the problem of propagating the signal throughout the graph, separating it from vanishing gradient phenomenon.