Understanding Oversquashing in GNNs through the Lens of Effective Resistance

Mitchell Black [1] **Zhengchao Wan** [2] **Amir Nayyeri** [1] **Yusu Wang** [2]

Abstract

Message passing graph neural networks (GNNs) are a popular learning architectures for graph-structured data. However, one problem GNNs experience is oversquashing, where a GNN has difficulty sending information between distant nodes. Understanding and mitigating oversquashing has recently received significant attention from the research community. In this paper, we continue this line of work by analyzing oversquashing through the lens of the *effective resistance* between nodes in the input graph. Effective resistance intuitively captures the "strength" of connection between two nodes by paths in the graph, and has a rich literature spanning many areas of graph theory. We propose to use *total effective resistance* as a bound of the total amount of oversquashing in a graph and provide theoretical justification for its use. We further develop an algorithm to identify edges to be added to an input graph to minimize the total effective resistance, thereby alleviating oversquashing. We provide empirical evidence of the effectiveness of our total effective resistance based rewiring strategies for improving the performance of GNNs.

1. Introduction

Graph neural networks (GNNs) are powerful tools for graph learning and optimization tasks (Scarselli et al., 2008). One major framework for GNNs is *message passing*, where node and edge features are repeatedly aggregated locally through node neighborhoods. While it has proven successful, message passing also suffers from several problem related to the topology of the graph. The number of layers of a GNN defines the radius of the neighborhood of a node from which information will be aggregated. When the number of layers is too small, the message passing will only be done locally, and the GNN will not be able to capture information from nodes outside this neighborhood. This problem is known as *underreaching*. On the other hand, choosing a large number of layers can lead to *oversmoothing*, where node features might be smoothed out and become indistinguishable (Cai & Wang, 2020; Oono & Suzuki, 2020). A third issue is *oversquashing* (Alon & Yahav, 2021), where as larger neighborhoods are considered, information from long-range interactions passing through certain bottlenecks of the graph will have negligible impact on the training of GNNs. This behaviour was named oversquashing as information from potentially exponentially many (with respect to the number of layers) nodes will be squashed into fixed-sized node vectors.

Understanding when oversquashing occurs is an active area of research. Recently, oversquashing has been analyzed using different techniques such as graph curvature (Topping et al., 2021) and information theory (Banerjee et al., 2022). Moreover, various *rewiring techniques* have been proposed to alleviate oversquashing, where edges are added or removed or edge weights are changed to decrease bottlenecks in the graph before applying GNNs (Arnaiz-Rodríguez et al., 2022; Deac et al., 2022; Karhadkar et al., 2022; Topping et al., 2021).

In this paper, we propose to analyze oversquashing through the lens of *effective resistance*. The concept of effective resistance originates from Electrical Engineering (Kirchhoff, 1847), where the effective resistance between two nodes u and v in an electrical network is the difference in voltage between u and v when a unit of current is inserted at u and removed at v. Since then, effective resistance has taken on a new life in Graph Theory, where effective resistance has been shown to be tied to many properties of the graph underlying the electrical network (Doyle & Snell, 1984; Lyons & Peres, 2017). For example, the effective resistance between a pair of vertices is proportional to the *commute time* between two vertices—the expected number of steps in a random walk from one vertex to the other and back (Chandra et al., 1996). The effective resistance between the end points of an edge is proportional to the probability of the edge being included in a random spanning tree of the graph (Biggs, 1997). Furthermore, effective resistance is closely

[1]School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon, USA [2]Halıcıoğlu Data Science Institute, University of California San Diego, San Diego, California, USA. Correspondence to: Mitchell Black <blackmit@oregonstate.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

related to the Cheeger constant for graphs that measures bottlenecks in graphs (Mémoli et al., 2022). Because of its various connections to many other objects (e.g., random walks and Laplacians), effective resistance has been widely used in practice; e.g., (Spielman & Srivastava, 2011; Alev et al., 2018; Ahmad et al., 2021).

These properties suggest that the effective resistance is a measure of how "well-connected" two nodes are (see Section 3). In this paper, we will show that the effective resistance can also be used to bound the amount oversquashing between two nodes in a GNN. In particular, the lower the effective resistance between a pair of nodes, the less oversquashing is experienced by a graph neural network sending messages between these nodes.

Contributions. In this paper, we propose to use effective resistance as a way to quantify oversquashing in graph neural networks. We then show how this perspective can be used to modify input graphs to alleviate oversquashing.

- In Section 3, we prove that the information passed from one node to another by any number of layers of a GNN is upper bounded by a quantity related to the effective resistance between the nodes.
- In Section 4, we utilize total effective resistance as a global measure of oversquashing and develop a rewiring algorithm for minimizing total effective resistance by adding edges to the graphs.
- In Section 5, we empirically demonstrate that our rewiring technique is effective in alleviating oversquashing. Our method outperforms the curvature based method SDRF from (Topping et al., 2021) and has similar performance compared to the spectral gap based method FoSR from (Karhadkar et al., 2022).

All missing technical details and proofs are in the Appendix.

More on related work. Alon & Yahav (2021) were the first to study the oversquashing problem in GNNs, although they did not provide a theoretical analysis of the problem. Topping et al. (2021) were the first to introduce a method for quantitatively analyzing the oversquashing problem. Inspired by Xu et al. (2018), Topping et al. proposed using norm of the Jacobian between node features at different levels of a GNN as a measure of oversquashing. Intuitively the norm of the Jacobian represents the ability of the features at one node to influence the features at another. They proved an upper bound on the norm of the Jacobian for certain nodes by the Balanced Forman Curvature of an edge. However, their theoretical analysis has the limitation that their final upper bound of the Jacobian via curvature only applies to nodes within 2-hop neighborhoods. In contrast, our analysis (Lemma 3.2 and Theorem 3.3) applies to any two nodes at any layer of the GNN. Banerjee et al. (2022) proposed an approach for analyzing the oversquashing problem using techniques from information theory.

Di Giovanni et al. (2023) also analyzed oversquashing using the commute time between a pair of nodes in a concurrent work. Both ours and their papers use similar approaches and reach the conclusion that large effective resistance between a pair of nodes results in more oversquashing. Additionally, they provide an analysis of how the width and depth of a GNN affect oversquashing.

In addition to analyzing the oversquashing problem, there has also been a line of research on ways to alleviate oversquashing. One of the most popular approaches is *rewiring* the graph: adding, removing, or reweighting the edges of the graph to improve the topology of the graph. For example, Alon & Yahav (2021) proposed using a fully connected graph in the last layer of a GNN.

A popular, generic approach to rewiring is to optimize some quantity measuring the graph topology. For example, Topping et al. (2021) proposed a rewiring technique to alleviate the oversquashing problem by increasing the curvature of edges in the graph. However, the most common approach has been to try to increase the *spectral gap* of the graph: the smallest eigenvalue of the Laplacian. Intuitively, the spectral gap is proportional to bottlenecks of graphs through the Cheeger inequality (Chung, 1996), so increasing the spectral gap decreases the bottleneck. However, there was previously no theoretical work directly tying the spectral gap to oversquashing (see Section 3.2). Some approaches to decrease the spectral gap have been to add edges (Karhadkar et al., 2022), flip edges (Banerjee et al., 2022), reweight edges (Arnaiz-Rodríguez et al., 2022), or use an expander to perform a GNN layer (Deac et al., 2022). Our rewiring technique is most similar to the approach of Karhadkar et al. (2022): we add edges to minimize the total effective resistance. Conceptually speaking, however, our approach may lead to better results as the total effective resistance reflects the entire spectrum of the graph Laplacian, including the spectral gap. See our discussion in Section 3.2.

Particularly relevant to this paper are rewiring techniques that incorporate information about effective resistance (Arnaiz-Rodríguez et al., 2022; Banerjee et al., 2022). These papers observe that edges with high effective resistance often appear in the bottleneck of the graph, so they target these edges in different ways. Banerjee et al. (2022) flip edges with probability proportional to their effective resistance to increase the spectral gap. Arnaiz-Rodríguez et al. (2022) reweight edges proportionally to their effective resistance. While our paper and these papers both study effective resistance as it relates to oversquashing, we make different observations about the relationship between oversquashing and effective resistance. In short, these papers observes that

edges of high effective resistance are important to the global topology of the graph so propose to target these edges. In contrast, our paper observes that oversquashing is in part the result of pairs of vertices with high effective resistance so propose to decrease total resistance. In particular, while the approach of Arnaiz-Rodríguez et al. (2022) is effective, its effectiveness can not be attributed to decreasing total resistance, as the reweighted graph will have approximately the same effective resistance between all pairs of nodes as the original graph (see Theorem 1 of (Arnaiz-Rodríguez et al., 2022).)

Additionally, while not a rewiring technique, Velingker et al. (2022) propose node and edge features based on effective resistance as a way of incorporating information about the graph topology into GNNs.

2. Background

This section reviews some definitions from Spectral Graph Theory; see books by Chung (1997) and Spielman (2019) for a more thorough introduction.

2.1. Matrices and Spectra of Graphs.

Let $G = (V, E)$ be a connected, undirected, unweighted graph with n vertices and m edges. Let A be the ***adjacency matrix*** and D be the ***degree matrix***. The ***Laplacian*** is $L = D - A$. Additionally, let $\hat{A} = D^{-1/2}AD^{-1/2}$ be the ***normalized adjacency matrix*** and $\hat{L} = I - \hat{A} = D^{-1/2}LD^{-1/2}$ be the ***normalized Laplacian***.

The matrices $\hat{L}$ and $\hat{A}$ have the same orthonormal basis of eigenvectors $\{z_i : 1 \leq i \leq n\}$ (up to choice of basis) but different eigenvalues. The eigenvalues λ_i of $\hat{L}$ are in the range $[0, 2]$, and the eigenvalues of $\hat{A}$ are $\mu_i = 1 - \lambda_i$, which are in the range $[-1, 1]$. The matrix $\hat{A}$ always has eigenvalue 1 and has eigenvalue -1 if and only if G is bipartite. We use the notational convention that $\lambda_n \geq \cdots \geq \lambda_2 > \lambda_1 = 0$ and $\mu_n \leq \cdots \leq \mu_2 < \mu_1 = 1$. z_1, the μ_1-eigenvector of $\tilde{A}$ satisfies $z_1(v) = \sqrt{d_v/2m}$, where d_v is the degree of a vertex v.

2.2. Graph Neural Networks

Consider a graph G with node features $X \in \mathbb{R}^{n \times d}$. We let $x_v \in \mathbb{R}^d$ denote the row in X corresponding to the vertex $v \in V$. A ***Graph Neural Network*** (GNN) updates the node features by iteratively aggregating features of nodes in the neighborhood. More precisely, the feature vectors at each layer are iteratively computed by

$$h_v^{(0)} := x_v, \; h_v^{(l+1)} = \phi_l \left(h_v^{(l)}, \sum_{u \in \mathcal{N}(v)} \hat{A}_{uv} \psi_l \left(h_u^{(l)} \right) \right)$$

for learnable functions ϕ_l and ψ_l. Note that this is a strict subset of the more general class of Message-Passing Neural Networks (Gilmer et al., 2017).

Relational GNNs. In the process of graph rewiring, the structure of the underlying graph will be changed. In order to retain information of the original graph and also exploit the new graph structure induced from graph rewiring, we use ***relational GNNs*** (R-GNNs) (Battaglia et al., 2018) to accommodate both information. The idea of using R-GNNs for rewired graphs was introduced in (Karhadkar et al., 2022). In the framework of R-GNNs, for a graph G, there exists a set $\mathcal{R}$ of relation types such that each edge $\{u, v\} \in E$ is associated with an edge type $r \in \mathcal{R}$. For each $v \in V$ and $r \in \mathcal{R}$, we let $\mathcal{N}_r(v) \subseteq \mathcal{N}(v)$ denote the collection of all neighbors of v incident to an edge of type r. An R-GNN is a function of the form

$$h_v^{(l+1)} = \phi_l \left(h_v^{(l)}, \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_r(v)} \hat{A}_{uv} \psi_l^r \left(h_u^{(l)} \right) \right)$$

for learnable functions ϕ_l and ψ_l^r.

3. Effective Resistance and Oversquashing

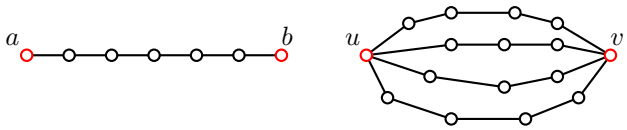


Figure 1. Two examples where effective resistance can be easily computed. For vertices u and v connected by several vertex-disjoint paths p, $R_{u,v} = (\sum_{uv\text{-paths } p} \text{length}(p)^{-1})^{-1}$. Left: $R_{a,b} = 6$, the length of the path. Right: $R_{u,v} = 10/9$.

Let u and v be vertices of G. The ***effective resistance*** between u and v is defined

$$R_{u,v} = (1_u - 1_v)^T L^+ (1_u - 1_v),$$

where 1_v is the indicator vector of the vertex v and L^+ is the pseudoinverse of L. The effective resistance can also be computed using the normalized Laplacian $\hat{L}$. This follows from a formula for effective resistance given by Lovász (1993, Corollary 3.2), but is somewhat non-standard. We provide a different proof in Appendix A.1 for completeness.

Lemma 3.1. *Let G be a connected graph. Let u and v be two vertices. Then*

$$R_{u,v} = \left(\frac{1}{\sqrt{d_u}} 1_u - \frac{1}{\sqrt{d_v}} 1_v \right)^T \hat{L}^+ \left(\frac{1}{\sqrt{d_u}} 1_u - \frac{1}{\sqrt{d_v}} 1_v \right).$$

Intuitively, the effective resistance is a measure of how "well-connected" two vertices u and v are. While "well-connected"-ness is informal, there are many theorems which

suggest such a connection. For example, if u and v are connected by k edge-disjoint paths of length at most l, then the effective resistance $R_{u,v}$ is a most l/k. Therefore, the more and shorter paths connecting u and v, the smaller the effective resistance between u and v. See the Introduction for more intuition behind effective resistance.

3.1. Effective Resistance and the Jacobian of GNNs.

As a way of measuring oversquashing in graph neural networks, Topping et al. (2021) proposed upper bounding the 2-norm of the Jacobian between node features $\|\partial h_u^{(r)}/\partial x_v\|$; here, both $h_u^{(r)}$ and x_v are vectors, so $\partial h_u^{(r)}/\partial x_v$ is the Jacobian matrix. The Jacobian captures the influence of initial feature vector x_v at vertex v upon the feature vector $h_u^{(r)}$ at vertex u at the rth layer of the GNN. A smaller upper bound on the partial derivative indicates that that the features at the node v can have less influence on the features at the node u. We adopt this way of analysis and establish a bound on the norm of the Jacobian matrix via the effective resistance.

First, we show how the norm of the Jacobian is upper bounded by the powers of the normalized adjacency matrix.

Lemma 3.2. *Let $u, v \in V$ and let $r \in \mathbb{N}$. Assume that $\|\nabla\phi_l\| \leq \alpha$ and $\max\{\|\nabla\psi_l\|, 1\} \leq \beta$ for all $l = 0, \ldots, r$, where ∇f denotes the Jacobian of a map f. Then*

$$\left\|\frac{\partial h_u^{(r)}}{\partial x_v}\right\| \leq (2\alpha\beta)^r \sum_{l=0}^{r} (\hat{A}^l)_{uv}.$$

This result is different from Lemma 1 in (Topping et al., 2021) in which the two vertices u and v are required to be **exactly** distance r apart from each other; while our result is for any two vertices.

We can now use Lemma 3.2 to establish a new bound via effective resistance. Recall that $\mu_n \leq \cdots \leq \mu_2 < \mu_1 = 1$ denote the eigenvalues of $\hat{A}$.

Theorem 3.3. *Let G be a non-bipartite graph. Let $u, v \in V$. Let $\|\nabla\phi_l\| \leq \alpha$ and $\max\{\|\nabla\psi_l\|, 1\} \leq \beta$. Let $d_{\min} = \min\{d_u, d_v\}$ and $d_{\max} = \max\{d_u, d_v\}$. Let $\max\{|\mu_2|, |\mu_n|\} \leq \mu$. Then*

$$\left\|\frac{\partial h_u^{(r)}}{\partial x_v}\right\| \leq (2\alpha\beta)^r \frac{d_{\max}}{2}\left(\frac{2}{d_{\min}}\left(r+1+\frac{\mu^{r+1}}{1-\mu}\right) - R_{u,v}\right)$$

Theorem 3.3 intuitively suggests that vertices with low effective resistance have a better influence over each other in message passing; that is, the node feature $h_u^{(r)}$ at node u in level r is more affected by the initial node feature x_v at node v. Intuitively this makes sense, as effective resistance is tied to the number and length of paths connecting u and v. The more and shorter paths connecting u and v, the lower the effective resistance between u and v is. This implies that there are more ways for a GNN to send messages between u and v, and indeed, by Theorem 3.3, the less oversquashing between u and v.

Sketch of proof of Theorem 3.3. Lemma 3.2 allows us to bound the Jacobian by a sum of entries of powers of the adjacency matrix. Therefore, we need a way of connecting powers of the adjacency matrix to effective resistance. For this, we use the following two lemmas, which themselves may be of independent interest. Detailed proofs of the theorem and the lemmas can be found in Appendix A.3.

Let $\hat{A}_r$ denote the restriction of $\hat{A}$ to the space orthogonal to the eigenvector z_1, i.e. $\hat{A}_r = \sum_{i=2}^{n} \mu_i z_i z_i^T$. Recall that the eigenvalues of $\hat{A}_r$ are in the range $[-1, 1)$, and $(-1, 1)$ if G is not bipartite. The pseudoinverse of $\hat{L}$ can be characterized as follows.

Lemma 3.4. *Let G be a connected, non-bipartite graph. Then $\hat{L}^+ = \sum_{j=0}^{\infty} \hat{A}_r^j$.*

This characterization of $\hat{L}^+$ allows us to prove the following relationship between the effective resistance and powers of the normalized adjacency matrix $\hat{A}$ (not just $\hat{A}_r$.)

Lemma 3.5. *Let G be a non-bipartite graph. Let u and v be two vertices in G. Then*

$$R_{u,v} = \sum_{i=0}^{\infty}\left(\frac{1}{d_u}(\hat{A}^i)_{uu} + \frac{1}{d_v}(\hat{A}^i)_{vv} - \frac{2}{\sqrt{d_v d_u}}(\hat{A}^i)_{uv}\right).$$

The upper bound in Theorem 3.3 follows from Lemma 3.2 and Lemma 3.5. □

Total Resistance We now take our analysis one step further and summarize message passing rate between **all pairs** of nodes at any given layer of GNN using the notion of ***total effective resistance*** R_{tot}—the sum of the effective resistance between all pairs of vertices.

As the partial derivative between a pair of vertices is bounded above by a function of the effective resistance, the total resistance bounds the sum of the Jacobian between all pairs of vertices in the graph. The following corollary follows immediately from Theorem 3.3.

Corollary 3.6. *Let G be a non-bipartite graph. Let $\|\nabla\phi_l\| \leq \alpha$ and $\max\{\|\nabla\psi_l\|, 1\} \leq \beta$. Let $d_{\min} = \min_{v\in V} d_v$ and $d_{\max} = \max_{v\in V} d_v$. Let $\max\{|\mu_2|, |\mu_n|\} \leq \mu$. Then*

$$\sum_{u\neq v\in V}\left\|\frac{\partial h_u^{(r)}}{\partial x_v}\right\| \leq (2\alpha\beta)^r \frac{d_{\max}}{2}\left(\frac{n\cdot(n-1)}{d_{\min}}\left(r+1+\frac{\mu^{r+1}}{1-\mu}\right) - R_{tot}\right).$$

Comparison with Curvature Bounds. Theorem 3.3 and Corollary 3.6 are inspired by Theorem 4 in (Topping et al., 2021), which bounds the Jacobian matrix between vertex features by the Balanced Forman curvature of an edge. In some ways, the effective resistance and Balanced Forman curvature of an edge are similar, as both measure how connected the endpoints are. However, our analysis generalizes the previous bound in several important ways.

(1) Our analysis can be applied to any pair of vertices in a graph, not just those vertices at distance 2.

(2) Effective resistance can be used to bound the oversquashing between node features after an arbitrary number of layers of a GNN, unlike Balanced Forman Curvature which can only measure oversqushing after 2 consecutive layers.

In short, the reason for both of these generalizations is that effective resistance measures the *global* connectivity between a pair of vertices, while Balanced Forman curvature only measures the *local* connectivity between a pair of nodes. See Figure 2 for an illustration.

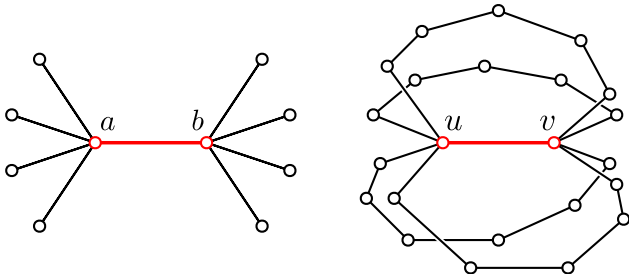


Figure 2. The edges $\{a, b\}$ and $\{u, v\}$ have the same Balanced Forman curvature of $\mathrm{Ric}(a, b) = \mathrm{Ric}(u, v) - 6/5$. However, their effective resistance are different ($R_{a,b} = 1$ and $R_{u,v} = 3/5$). This shows how the curvature only measure local connectivity and does not distinguish global connectivity as effective resistance does.

Comparison with Commute Time Bounds Concurrently to this work, Di Giovanni et al. (2023) showed that oversquashing between a pair of nodes u and v could be bounded by the commute time $\tau(u, v)$—the expected number of steps in a random walk from u to v and back to u. The commute time and effective resistance are proportional: $\tau(u, v) = 2mR_{u,v}$ (Chandra et al., 1996); thus, our Theorem 3.3 and their Theorem 5.5 are analogous. Indeed, both theorems agree that oversquashing occurs between nodes with large effective resistance/commute time. The two theorems also use similar techniques to connect effective resistance/commute time to the Jacobian of a GNN. The main differences between our theorems are the result of differences in the quantities we bound (both are related to the Jacobian of the GNN) and differences in assumptions about the GNN.

3.2. Effective Resistance and the Spectral Gap

Let $0 = \sigma_1 \leq \sigma_2 \leq \cdots \sigma_n$ denote the eigenvalues of the (un-normalized) Laplacian L. The second eigenvalue σ_2 is called the ***spectral gap***[1] of the graph G. The spectral gap is often used as a measure of the "bottleneck" of a graph. This is because the spectral gap is proportional to the size of the sparsest cut in the graph, a classic result known as ***Cheeger's Inequality*** (Chung, 1996).

Previous research has attempted to connect oversquashing to the spectral gap of the graph (Topping et al., 2021; Banerjee et al., 2022). This has motivated rewiring heuristics aimed at raising at the spectral gap (Arnaiz-Rodríguez et al., 2022; Banerjee et al., 2022; Deac et al., 2022; Karhadkar et al., 2022). However, unlike our theoretical analysis for effective resistance (Theorem 3.3 and Corollary 3.6), while the use of spectral gap for measuring oversquashing is intuitive, there was previously no theoretical evidence for how the spectral gap directly bounds information passing between nodes.

In this section, we first discuss the connections between spectral gap and effective resistance in order to derive a first-step theoretical justification for using spectral gap for bounding oversquashing. Then, we discuss potential limitations of only using the spectral gap.

The following existing result shows that the worst-case effective resistance between any pair of nodes is proportional to the spectral gap.

Theorem 3.7 (Theorem 4.2, (Chandra et al., 1996))**.** *Let $R_{\max}$ denote the maximum effective resistance between any pair of vertices in G. Then*

$$\frac{1}{n\sigma_2} \leq R_{\max} \leq \frac{2}{\sigma_2}.$$

Corollary 3.6 and Theorem 3.7 combine to reinforce the idea that low spectral gap is tied to oversquashing, as seen by the following corollary.

Corollary 3.8. *Under the same assumptions as in Corollary 3.6, one has that*

$$\sum_{u \neq v \in V} \left\| \frac{\partial h_u^{(r)}}{\partial x_v} \right\| \leq (2\alpha\beta)^r \frac{d_{\max}}{2} \left(\frac{n \cdot (n-1)}{d_{\min}} \left(r + 1 + \frac{\mu^{r+1}}{1-\mu} \right) - \frac{1}{n\sigma_2} \right).$$

[1] In this section, we focus on the spectral gap and eigenvalues of the *unnormalized* Laplacian, while previous papers studying oversquashing have focused on the spectral gap of the *normalized* Laplacian. There are variants of Cheeger's inequality for both the normalized and unnormalized spectral gap (Chung, 1997), so both spectral gaps provide a measure of the connectivity and bottleneck of a graph. The eigenvalues of L and $\hat{L}$ are also closely related as follows: $d_{\min}\lambda_k \leq \sigma_k \leq d_{\max}\lambda_k$.

Of course, the bound above is looser than the bound using R_{tot} in Corollary 3.6. Furthermore, the following result suggests that oversquashing behavior of the graph is tied not just to the spectral gap, but rather to the *entire* spectrum of the Laplacian. Therefore, raising the entire spectrum of the Laplacian, not just the spectral gap, could potentially further reduce oversquashing.

Theorem 3.9 (Section 2.5, (Ghosh et al., 2008))**.** *Let G be a connected graph with n vertices, Laplacian L, and total resistance R_{tot}. Then*

$$R_{tot} = n \cdot \operatorname{tr} L^{+} = n \sum_{i=2}^{n} \frac{1}{\sigma_i}$$

The higher eigenvalues of L also carry topological meaning about the graph. Just as the spectral gap λ_2 measures the obstruction to bipartitioning a graph (the "bottleneck"), the kth smallest eigenvalue λ_k of $\hat{L}$ is related to partitioning a graph into k parts (Lee et al., 2014). See Footnote 1 for the relationship between the eigenvalues λ_k and σ_k.

4. Minimizing Total Resistance by Rewiring

Motivated by Corollary 3.6, we propose to address oversquashing by "rewiring" a graph to minimize its total resistance. Adding any edge to the graph will decrease its total resistance (a result known as ***Rayleigh Monotonicity***), so in this section, we (1) derive a formula to determine how much adding a specific edge decreases the total resistance and (2) propose a rewiring method that greedily adds the edge to the graph that most decreases total resistance. Note that our "rewiring" just refers to adding edges, while some previous usage of the term "rewiring" might refer to replacing one edge with another (Topping et al., 2021; Banerjee et al., 2022).

Change to R_{tot} after adding one edge. We first need a new notion. The ***biharmonic distance*** between a pair of vertices u and v is

$$B_{u,v} = \sqrt{(1_u - 1_v)^T (L^{+})^2 (1_u - 1_v)}.$$

The biharmonic distance was first introduced in the context of geometry processing (Lipman et al., 2010). However, before it was properly named, it was discovered that the squared biharmonic distance between u and v is proportional to the partial derivative of the total resistance with respect to the weight of the edge $\{u, v\}$, i.e. $\partial R_{tot}/\partial w_{u,v} = -n \cdot B_{u,v}^2$ (Ghosh et al., 2008). This suggests that the biharmonic distance can be used as a measure for the effect an edge has on the global connectivity of the graph.

The following theorem may be seen as the unweighted and combinatorial analogue of the previous result (but is proved using completely different means.) This theorem allows us to calculate how much the total resistance decreases when an (unweighted) edge $\{u, v\}$ is added to the graph.

Theorem 4.1. *Let G be a connected graph with n vertices. Let $\{u, v\}$ be an edge not in G. The difference in total resistance after adding the edge $\{u, v\}$ to G is*

$$R_{tot}(G) - R_{tot}(G \cup \{u, v\}) = n \cdot \frac{B_{u,v}^2}{1 + R_{u,v}}$$

Sketch of proof of Theorem 4.1. Note that adding the edge $\{u, v\}$ to G changes the Laplacian from L to $L + (1_u - 1_v)(1_u - 1_v)^T$. Hence by Theorem 3.9 we need to compare the traces of the pseudoinverses of L and $L + (1_u - 1_v)(1_u - 1_v)^T$. This naturally leads us Woodbury's formula:

Lemma 4.2 (Woodbury's Formula)**.** *Let A be an invertible matrix. Let x be a vector. Then*

$$(A + xx^T)^{-1} = A^{-1} - A^{-1}x(1 + x^T A^{-1} x)^{-1} x^T A^{-1}.$$

As L is singular, we cannot apply Woodbury's Formula directly to $L + (1_u - 1_v)(1_u - 1_v)^T$. Hence, we consider the variant of the Laplacian $L + \frac{11^T}{n}$, where 1 is the all-ones vector. If G is connected, then $L + \frac{11^T}{n}$ is invertible. Moreover, it can be shown that

Lemma 4.3 ((Ghosh et al., 2008))**.** *Let G be a connected graph. Then*

- $R_{u,v} = (1_u - 1_v)^T (L + \frac{11^T}{n})^{-1} (1_u - 1_v)$;
- $B_{u,v}^2 = (1_u - 1_v)^T (L + \frac{11^T}{n})^{-2} (1_u - 1_v)$;
- $R_{tot} = n \cdot \operatorname{tr}(L + \frac{11^T}{n})^{-1} - n$.

We can therefore apply Lemma 4.2 to compute $(L + \frac{11^T}{n} + (1_u - 1_v)(1_u - 1_v)^T)^{-1}$, take the trace, and conclude the theorem. See Appendix A.4 for all the details. □

Figure 3 shows the value $n \cdot \frac{B_{u,v}^2}{1+R_{u,v}}$ for edges in various graphs.

Rewiring heuristic. Motivated by Theorem 4.1, we propose the following heuristic, ***Greedy Total Resistance (GTR) rewiring***, to minimize the total resistance: repeatedly add the edge $\{u, v\}$ that maximizes $B_{u,v}^2/(1 + R_{u,v})$. For disconnected graphs, the effective resistance and biharmonic distance between vertices in different components is not meaningful. Therefore, we only add edges between vertices that are already in the same connected component. While we could also use Theorem 4.1 to determine which edge to remove to most decrease the total resistance, we

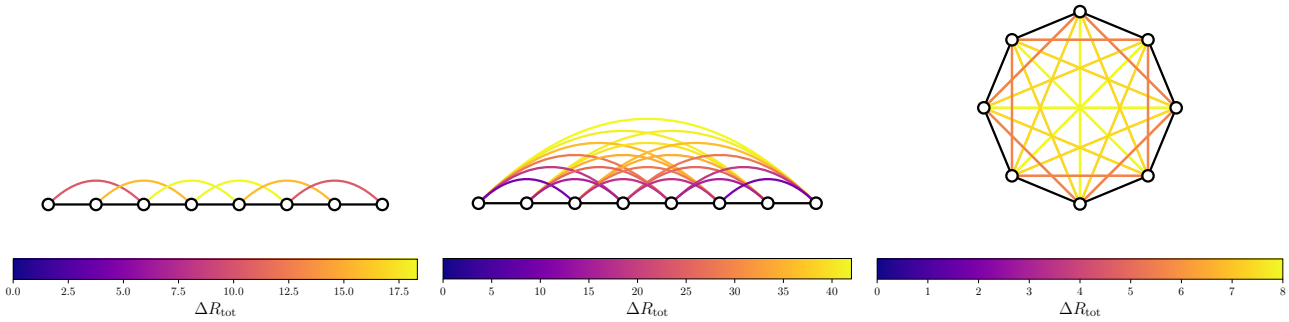


Figure 3. When an edge $\{u, v\}$ is added to a graph, it decreases the total resistance by $\Delta R_{\text{tot}} := n \cdot (B^2_{u,v}/(1 + R_{u,v}))$ (Theorem 4.1). This figure shows the value ΔR_{tot} for various pairs of vertices in graphs with $n = 8$ vertices. Edges originally in the graph are black, and edges not in the graph are colored according to ΔR_{tot}. Left: For pairs of vertices with equal effective resistance $R_{u,v} = 2$, edges towards the center of the graph have the highest biharmonic distance $B_{u,v}$. Center: The pairs of vertices that maximize ΔR_{tot} are those at opposite ends of the path. Right: The pairs of vertices that maximize ΔR_{tot} are on opposite sides of the cycle.

will only add edges in this paper. A PyTorch Geometric implementation of the GTR algorithm is available online[2]. See Appendix E for plots of how much GTR decreases total resistance for various datasets.

Time complexity. GTR can naively be implemented in $O(n^3)$ time, but there are more sophisticated algorithms that take time $O(m \operatorname{poly} \log n + n^2 \operatorname{poly} \log n)$. See Appendix B for an asymptotic and empirical analysis of its runtime.

Adding multiple edges. While Theorem 4.1 tells us which single edge most decreases the total resistance when added to the graph, unfortunately, we cannot use this formula to determine which set of $k \geq 2$ edges most decrease the total resistance of the graph. In Appendix C, we give an example of a graph where the two edges that most decrease the total resistance are not the two edges that maximize the formula in Theorem 4.1.

Another challenge for designing recursive algorithms to add multiple edges is that the amount an edge decreases the total resistance is ***non-monotonic*** with respect to subgraphs. By non-monotonic, we mean that for nested graphs $H \subset G$, the amount an edge decreases the total resistance when added to G can be *more* than the amount the same edge would decrease the total resistance when added to H. Appendix C gives an example where this is the case. Intuitively, this means that an edge can become more important to the global topology of a graph when more edges are added. This is in contrast to the effective resistance, which only decreases with the addition of more edges.

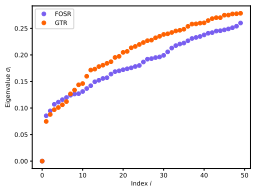
The best algorithm we know for computing the set of k edges that most decrease the total resistance is a brute-force search over all $O(\binom{\binom{n}{2}}{k})$ sets of k edges. It was recently shown that finding the k edges that most decrease the total resistance is NP-Hard (Kooij & Achterberg, 2023). Because of this, it is reasonable to use a heuristic rather than exactly compute the best edges to add to decrease total resistance.

[2] https://github.com/blackmit/gtr_rewiring

5. Experiments

We primarily compare our new GTR rewiring algorithm with the ***FoSR*** (for "first-order spectral rewiring") algorithm proposed by Karhadkar et al. (2022), as FoSR is the rewiring strategy with the best performance. FoSR aims at reducing oversquashing in graphs by increasing the spectral gap. FoSR is perhaps the rewiring heuristic most similar to GTR for two reasons. First, it only changes the topology of the graph by adding edges. Second, it is designed to increase the spectral gap of the graph, which will necessarily increase the total resistance of the graph.

5.1. Spectral Gap vs. Total Resistance



	FoSR	GTR
σ_2	0.085	0.075
R_{tot}	4250377	4114024

Figure 4. A comparision of the largest connected component of Cora after adding 50 edges with FoSR and GTR. Left: A plot of the smallest 50 eigenvalues of the Laplacian. Right: The spectral gap and total resistance.

To compare FoSR and GTR, we use both methods to add 50 edges to the largest connected component of the Cora citation network (McCallum et al., 2000). Figure 4 shows the 50 smallest eigenvalue after rewiring. FoSR increases the first few eigenvalues (including the spectral gap) more,

while GTR increases the larger eigenvalues more. In total, GTR does more to decrease the total resistance of the graph.

5.2. Graph Classification

We evaluate our rewiring heuristic, GTR, as a preprocessing step for training a graph neural network to perform graph classification. We compare GTR with the following rewiring method: making the last layer fully connected (Last FA) and making all layers fully connected (All FA) from (Alon & Yahav, 2021), DIGL from (Gasteiger et al., 2019), SDRF from (Topping et al., 2021), and FOSR (Karhadkar et al., 2022). We also report results for no rewiring (None). We conduct the same experiment as in (Karhadkar et al., 2022) for GTR; see Table 1 for results. All results except for GTR and those marked with an asterisk are taken from Table 1 of (Karhadkar et al., 2022).

Datasets. We test GTR on the same set of graph classification benchmarks as Karhadkar et al. (2022). All datasets are from the TUDataset (Morris et al., 2020).

Experiments. We compare four types of graph convolutions: GCN (Kipf & Welling, 2017), Relational-GCN (R-GCN) (Battaglia et al., 2018), GIN (Xu et al., 2019), and Relational-GIN (R-GIN). Relational graph neural networks perform different aggregation steps for edges of different types. In the case of GTR, we use two edge types: original graph edges and new edges added by the rewiring algorithms. We tune the number of edges added by GTR and fix all other hyperparameters. Full experimental details can be found in Appendix D.

Results. Test accuracies are presented in Table 1 and the number of edges added for each graph are reported in Appendix D. We observe the following: (1) In general, both our GTR and FoSR outperform the rewiring strategies DIGL, SDRF, or no rewiring at all. In particular, for the case of relational versions of GNNs (i.e., R-GCN and R-GIN), these two approaches often out-perform no-rewiring or SDRF by a large margin. Note that SDRF adds edges based on a local curvature criterion; while both FoSR and our GTR can add any edges, taking the global connectivity of graph into account. Table 1 shows that both global strategies outperform the local SDRF, especially for the relation-GNN cases. (2) The performance of our GTR and FoSR are similar for the GIN and R-GIN architectures. On R-GCN however, GTR not only outperforms FoSR, but often by a large margin.

5.3. Edge Ablation

In Appendix F, we repeat the experiment from Section 5.2 but vary the number of edges added. In particular, our experiments suggest that *there is no optimal number of edges to add* that works across datasets. Moreover, *performance does not necessarily increase as total resistance decreases*, which we can see by comparing FoSR and GTR to Every Layer FA in Table 1. Therefore, we recommend treating the number of edges added as a hyperparameter to be tuned during training.

5.4. Hidden Dimension Ablation

Another method for address oversquashing is to increase the hidden dimension of the GNN (Alon & Yahav, 2021; Di Giovanni et al., 2023). To compare this method with rewiring, in Appendix G, we repeat the experiment from Section 5.2 but vary both the number of edges added and the hidden dimension. We conclude that *rewiring and increasing the hidden dimension are complementary methods for addressing oversquashing*, as doing either or both increases the performance of GNNs.

6. Concluding Remarks

In this paper, we have provided theoretical evidence that effective resistance can be used as a bound on oversquashing between a pair of nodes in a graph, and that the total resistance can be used as a bound of total oversquashing in a graph. We have also empirically demonstrated that lowering total resistance improves the performance of graph neural networks. Indeed, rewiring techniques based on total effective resistance can significantly improve performance of GNN / R-GNNs for graph classification tasks, reinforcing the notion that improving the connectivity of a graph can improve the performance of graph neural networks.

Limitations and future work. We provide theoretical evidence (Theorem 3.3) showing that total effective resistance can be used to bound the amount of oversquashing in a graph. This is in contrast to previous work on oversquashing which relates oversquashing to the spectral gap through intuition alone. While we prove that the spectral gap can also be used to bound oversquashing (Corollary 3.8), the bound for total resistance is tighter than the bound for the spectral gap.

Despite the theoretical strength of using total resistance over spectral gap for measuring oversquashing, more research is needed to contrast the effects of the two on oversquashing. A challenge to this task is that the total resistance and spectral gap are intimately linked; for example, adding edges to the graph will necessarily both decrease the total resistance and increase the spectral gap. The oversquashing issue becomes more prominent for graphs with long range interactions (e.g., (Dwivedi et al., 2022)). Hence it will be interesting to explore a much broader family of graph benchmarks to study the pros and cons of different rewiring methods.

Finally, we also note that currently we employ a greedy approach to identify a collection of edges to be inserted into an input graph as shortcuts. As discussed in Section 4,

Table 1. Results of different combinations of rewiring and convolutions on different graph classification datasets. **First**, **second**, and **third** best results are colored. See Section 5.2 for discussion. All results except for GTR are from (Karhadkar et al., 2022), with the exception of R-GIN FoSR results marked with an asterisk ($*$); these are the best runs from the edge ablation study (Section 5.3).

GCN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
None	72.15 ± 2.44	70.98 ± 0.74	27.67 ± 1.16	68.26 ± 1.10	49.77 ± 0.82	33.78 ± 0.49
Last FA	70.05 ± 2.03	71.02 ± 0.96	26.47 ± 1.20	68.49 ± 0.95	48.98 ± 0.95	33.32 ± 0.44
Every FA	70.45 ± 1.96	60.04 ± 0.93	18.33 ± 1.04	48.49 ± 1.04	48.17 ± 0.80	51.80 ± 0.42
DIGL	79.70 ± 2.15	70.76 ± 0.77	35.72 ± 1.12	76.04 ± 0.78	64.39 ± 0.91	54.50 ± 0.41
SDRF	71.05 ± 1.87	70.92 ± 0.79	28.37 ± 1.17	68.62 ± 0.85	49.40 ± 0.90	33.45 ± 0.47
FoSR	80.00 ± 1.57	73.42 ± 0.81	25.07 ± 0.994	70.33 ± 0.72	49.66 ± 0.86	33.84 ± 0.58
GTR	79.10 ± 1.86	72.59 ± 2.48	27.52 ± 0.99	68.99 ± 0.61	49.92 ± 0.99	33.05 ± 0.40
R-GCN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
None	69.25 ± 2.09	69.52 ± 0.73	28.60 ± 1.19	49.85 ± 0.65	50.01 ± 0.92	33.60 ± 1.05
Last FA	70.55 ± 1.81	69.53 ± 0.82	28.23 ± 1.14	49.80 ± 0.63	50.65 ± 0.96	34.73 ± 1.19
Every FA	70.50 ± 1.84	71.67 ± 0.88	33.40 ± 1.14	49.95 ± 0.59	50.50 ± 0.89	33.62 ± 0.98
DIGL	73.40 ± 2.00	68.23 ± 0.85	28.28 ± 1.21	50.00 ± 0.62	49.67 ± 0.84	16.93 ± 1.44
SDRF	72.30 ± 2.22	69.11 ± 0.76	33.48 ± 1.25	58.62 ± 0.65	53.64 ± 1.04	67.99 ± 0.39
FoSR	84.45 ± 1.57	73.80 ± 0.69	35.66 ± 1.151	76.59 ± 0.53	64.05 ± 1.12	70.65 ± 0.48
GTR	$\mathbf{85.50 \pm 1.47}$	$\mathbf{75.78 \pm 0.76}$	41.33 ± 1.28	80.18 ± 0.60	65.09 ± 0.93	74.34 ± 0.41
GIN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
None	77.70 ± 3.60	70.80 ± 0.83	33.80 ± 1.12	86.79 ± 1.06	70.18 ± 0.99	72.99 ± 0.38
Last FA	83.45 ± 1.74	72.30 ± 0.67	47.40 ± 1.39	90.22 ± 0.48	70.91 ± 0.79	75.06 ± 0.41
Every FA	72.55 ± 3.02	70.38 ± 0.91	28.38 ± 1.05	50.36 ± 0.65	49.16 ± 0.87	32.89 ± 0.39
DIGL	79.70 ± 2.15	70.76 ± 0.77	35.72 ± 1.20	76.04 ± 0.77	64.39 ± 0.91	54.50 ± 0.41
SDRF	78.40 ± 2.80	69.81 ± 0.79	35.82 ± 1.09	86.44 ± 0.59	69.72 ± 1.15	72.96 ± 0.42
FoSR	78.00 ± 2.22	75.11 ± 0.82	29.20 ± 1.38	87.35 ± 0.60	71.21 ± 0.92	73.28 ± 0.42
GTR	77.60 ± 2.84	73.13 ± 0.69	30.57 ± 1.42	86.98 ± 0.66	71.28 ± 0.86	72.93 ± 0.42
R-GIN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
None	83.05 ± 1.44	70.50 ± 0.81	39.02 ± 1.17	87.97 ± 0.56	68.89 ± 0.87	75.54 ± 0.32
Last FA	80.60 ± 1.64	70.30 ± 0.84	$\mathbf{48.18 \pm 1.40}$	$\mathbf{90.00 \pm 0.65}$	69.71 ± 1.03	75.43 ± 0.49
Every FA	83.05 ± 1.52	71.05 ± 0.91	$\mathbf{54.95 \pm 1.33}$	56.86 ± 0.94	$\mathbf{71.48 \pm 0.88}$	75.43 ± 0.48
DIGL	81.45 ± 1.49	71.31 ± 0.76	37.60 ± 1.20	74.43 ± 0.72	63.93 ± 0.95	54.71 ± 0.42
SDRF	82.70 ± 1.78	70.70 ± 0.82	39.58 ± 1.33	86.83 ± 0.52	70.21 ± 0.81	$\mathbf{76.48 \pm 0.39}$
FoSR	$\mathbf{86.15 \pm 1.49}$	$\mathbf{75.25 \pm 0.86}^*$	45.55 ± 0.13	$\mathbf{90.94 \pm 0.47}^*$	$\mathbf{71.96 \pm 0.69}^*$	$\mathbf{77.20 \pm 0.38}^*$
GTR	$\mathbf{86.10 \pm 1.76}$	$\mathbf{75.64 \pm 0.74}$	$\mathbf{50.03 \pm 1.32}$	$\mathbf{90.41 \pm 0.41}$	$\mathbf{71.49 \pm 0.93}$	$\mathbf{77.45 \pm .039}$

finding the k best edges to add to decrease total resistance is NP-Hard (Kooij & Achterberg, 2023), and it is not clear whether such a greedy strategy even leads to an approximation algorithm of selecting the optimal set of k edges to minimizing total effective resistance. We leave the problem of identifying efficient approximation algorithms for the optimal edges or better heuristics for minimizing total effective resistance as a future direction to investigate.

Acknowledgements

Mitchell Black and Amir Nayyeri are supported in part by NSF grants CCF-1941086 and CCF-1816442. Zhengchao Wan and Yusu Wang are supported in part by NSF Grants CCF-2217033 and CCF-2112665, as well as a gift fund from Qualcomm. Mitchell Black would like to thank Yusu Wang for supporting a visit to UCSD where this project was initiated.

References

Ahmad, T., Jin, L., Lin, L., and Tang, G. Skeleton-based action recognition using sparse spatio-temporal gcn with edge effective resistance. *Neurocomputing*, 423:389–398, 2021.

Alev, V. L., Anari, N., Lau, L. C., and Oveis Gharan, S. Graph clustering using effective resistance. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

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

Arnaiz-Rodríguez, A., Begga, A., Escolano, F., and Oliver, N. M. Diffwire: Inductive graph rewiring via the lovász bound. In *The First Learning on Graphs Conference*, 2022. URL https://openreview.net/forum?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 *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8, 2022. doi: 10.1109/Allerton49937.2022.9929363.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Biggs, N. Algebraic potential theory on graphs. *Bulletin of the London Mathematical Society*, 29(6):641–682, 1997.

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, Dec 1996. ISSN 1420-8954. doi: 10.1007/BF01270385. URL https://doi.org/10.1007/BF01270385.

Chung, F. R. Laplacians of graphs and cheeger's inequalities. *Combinatorics, Paul Erdos is Eighty*, 2(157-172):13–2, 1996.

Chung, F. R. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Deac, A., Lackenby, M., and Veličković, P. Expander graph propagation. In *The First Learning on Graphs Conference*, 2022. URL https://openreview.net/forum?id=IKevTLt3rT.

Di Giovanni, F., Giusti, L., Barbero, F., Luise, G., Lio', P., and Bronstein, M. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2023.

Doyle, P. G. and Snell, J. L. *Random walks and electric networks*, volume 22. American Mathematical Soc., 1984.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=in7XC5RcjEn.

Gasteiger, J., Weißenberger, S., and Günnemann, S. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.

Ghosh, A., Boyd, S., and Saberi, A. Minimizing effective resistance of a graph. *SIAM review*, 50(1):37–66, 2008.

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.

Jambulapati, A. and Sidford, A. Ultrasparse ultrasparsifiers and faster laplacian system solvers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 540–559. SIAM, 2021.

Karhadkar, K., Banerjee, P. K., and Montúfar, G. Fosr: First-order spectral rewiring for addressing oversquashing in gnns, 2022. URL https://arxiv.org/abs/2210.11790.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Kirchhoff, G. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, 148:497–508, 1847.

Kooij, R. E. and Achterberg, M. A. Minimizing the effective graph resistance by adding links is np-hard, 2023.

Lee, J. R., Gharan, S. O., and Trevisan, L. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)*, 61(6):1–30, 2014.

Lipman, Y., Rustamov, R. M., and Funkhouser, T. A. Biharmonic distance. *ACM Trans. Graph.*, 29(3), Jul 2010. ISSN 0730-0301. doi: 10.1145/1805964.1805971. URL https://doi.org/10.1145/1805964.1805971.

Lovász, L. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

Lyons, R. and Peres, Y. *Probability on trees and networks*, volume 42. Cambridge University Press, 2017.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2): 127–163, Jul 2000. ISSN 1573-7659. doi: 10.1023/A:1009953814988. URL https://doi.org/10.1023/A:1009953814988.

Mémoli, F., Wan, Z., and Wang, Y. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

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

Spielman, D. Spectral and algebraic graph theory. Available at http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf (2021/12/01), 2019.

Spielman, D. A. and Srivastava, N. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40 (6):1913–1926, 2011. doi: 10.1137/080734029. URL https://doi.org/10.1137/080734029.

Spielman, D. A. and Teng, S.-H. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pp. 81–90, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138520. doi: 10.1145/1007352.1007372. URL https://doi.org/10.1145/1007352.1007372.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2021.

Velingker, A., Sinop, A. K., Ktena, I., Veličković, P., and Gollapudi, S. Affinity-aware graph networks, 2022.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/xu18c.html.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

A. Proofs

A.1. Proof of Lemma 3.1

Lemma 3.1. *Let G be a connected graph. Let u and v be two vertices. Then*

$$R_{u,v} = \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v\right)^T \hat{L}^+ \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v\right).$$

Proof. We will prove this using an alternative but well-known characterization of effective resistance in terms of uv-flows. First, we must define another matrix associated with a graph. Let ∂ be the $n \times m$ ***boundary matrix*** of the graph G, where $n := |V|$ and $m := |E|$. The matrix ∂ is defined such that for an edge $e = \{u, v\}$, the column $\partial 1_e = 1_u - 1_v$. (The order of u and v is arbitrary for what follows.)

Many of the definitions in this paper can alternatively be expressed in terms of the boundary matrix. The Laplacian can be expressed $L = \partial\partial^T$, the normalized Laplacian $\hat{L} = D^{-1/2}\partial(D^{-1/2}\partial)^T$, and the effective resistance $R_{u,v} = \min\{\|f\|^2 : \partial f = (1_u - 1_v)\}$. Phrased differently, the effective resistance between u and v is the minimum squared-2-norm of any uv-flow. This characterization of the effective resistance follows from the general fact that for any matrix AA^T and any vector $x \in \operatorname{im} A$ we have that $x^T(AA^T)^+x = (A^+x)^T(A^+x) = \min\{\|y\|^2 : Ay = x\}$. The proof of the current lemma just applies this fact twice.

$$\begin{aligned}
R_{u,v} =& (1_u - 1_v)^T L^+ (1_u - 1_v) \\
=& \min\{\|f\|^2 : \partial f = (1_u - 1_v)\} \\
=& \min\{\|f\|^2 : D^{-1/2}\partial f = D^{-1/2}(1_u - 1_v)\} && \text{(as } D^{-1/2} \text{ is bijective)} \\
=& \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v\right)^T \hat{L}^+ \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v\right).
\end{aligned}$$

□

A.2. Proof of Lemma 3.2

Lemma 3.2. *Let $u, v \in V$ and let $r \in \mathbb{N}$. Assume that $\|\nabla\phi_l\| \le \alpha$ and $\max\{\|\nabla\psi_l\|, 1\} \le \beta$ for all $l = 0, \ldots, r$, where ∇f denotes the Jacobian of a map f. Then*

$$\left\|\frac{\partial h_u^{(r)}}{\partial x_v}\right\| \le (2\alpha\beta)^r \sum_{l=0}^{r} (\hat{A}^l)_{uv}.$$

Proof. We prove this by induction on the layer r. For the base case of $r = 0$, either $u = v$ or $u \neq v$; in the first case

$$\frac{\partial h_u^{(0)}}{\partial x_v} = \frac{\partial x_v}{\partial x_v} = \mathrm{Id}_{d\times d},$$

and in the second case,

$$\frac{\partial h_u^{(0)}}{\partial x_v} = \frac{\partial x_u}{\partial x_v} = 0_{d\times d}.$$

Therefore,

$$\left\|\frac{\partial h_u^{(0)}}{\partial x_v}\right\| \le \max\{\|\mathrm{Id}_{d\times d}\|, \|0_{d\times d}\|\} = 1. \tag{1}$$

Assume that the statement holds for some $r \geq 0$. We now prove the inductive case of $r+1$.

$$
\begin{aligned}
&\left\| \frac{\partial h_u^{(r+1)}}{\partial x_v} \right\| = \left\| \nabla_1 \phi_r \cdot \frac{\partial h_u^{(r)}}{\partial x_v} + \nabla_2 \phi_r \cdot \sum_{w \in \mathcal{N}(u)} \hat{A}_{uw} \cdot \nabla \psi_r \cdot \frac{\partial h_w^{(r)}}{\partial x_v} \right\| \\
&\leq \|\nabla_1 \phi_r\| \cdot \left\| \frac{\partial h_u^{(r)}}{\partial x_v} \right\| + \|\nabla_2 \phi_r\| \cdot \sum_{w \in \mathcal{N}(u)} \hat{A}_{uw} \|\nabla \psi_r\| \cdot \left\| \frac{\partial h_w^{(r)}}{\partial x_v} \right\| && \text{(as } \hat{A}_{uw} \text{ positive } \forall\, u,\, w) \\
&\leq \alpha \cdot \left\| \frac{\partial h_u^{(r)}}{\partial x_v} \right\| + \alpha\beta \cdot \sum_{w \in \mathcal{N}(u)} \hat{A}_{uw} \cdot \left\| \frac{\partial h_w^{(r)}}{\partial x_v} \right\| \\
&\leq 2^r (\alpha\beta)^{r+1} \sum_{l=0}^{r} (\hat{A}^l)_{uv} + 2^r (\alpha\beta)^{r+1} \sum_{l=0}^{r} \sum_{w \in \mathcal{N}(u)} \hat{A}_{uw} (\hat{A}^l)_{wv} && \text{(induction)} \\
&= 2^r (\alpha\beta)^{r+1} \sum_{l=0}^{r} (\hat{A}^l)_{uv} + 2^r (\alpha\beta)^{r+1} \sum_{l=1}^{r+1} (\hat{A}^l)_{uv} && \text{(definition of matrix multiplication)} \\
&\leq (2\alpha\beta)^{r+1} \sum_{l=0}^{r+1} (\hat{A}^l)_{uv}.
\end{aligned}
$$

Here $\nabla \phi_r = [\nabla_1 \phi_r | \nabla_2 \phi_r]$ and $\nabla \psi_r$ denote the Jacobian matrices for ϕ_r and ψ_r, respectively. $\nabla_1 \phi_r$ corresponds to partial derivatives w.r.t. the first several arguments in ϕ_r corresponding to $h_v^{(r)}$ in the formula $\phi_r\left(h_v^{(r)}, \sum_{u \in \mathcal{N}(v)} \hat{A}_{uv} \psi_l \left(h_u^{(r)}\right)\right)$ and $\nabla_2 \phi_r$ is defined similarly. In the second inequality, we used the fact for 2-norm that $\|[A|B]\| \geq \max\{\|A\|, \|B\|\}$. In the third inequality, we used the fact that $\beta \geq 1$ and in this way we have that $\alpha \leq \alpha\beta$. □

A.3. Proof of Theorem 3.3

In this section, we provide proofs of Lemma 3.4, Lemma 3.5 and Theorem 3.3.

Lemma 3.4. *Let G be a connected, non-bipartite graph. Then $\hat{L}^+ = \sum_{j=0}^{\infty} \hat{A}_r^j$.*

Proof. First, recall that the eigenvalues of $\hat{A}_r$ are in the range $(-1, 1)$ if G is not bipartite. Also note that any number $\mu \in (-1, 1)$ satisfies $\sum_{j=0}^{\infty} \mu^j = \frac{1}{1-\mu}$. We prove the lemma by applying this fact to the spectral decomposition of $\hat{L}^+$.

$$
\begin{aligned}
\hat{L}^+ &= \sum_{i=2}^{n} \frac{1}{\lambda_i} z_i z_i^T = \sum_{i=2}^{n} \frac{1}{1-\mu_i} z_i z_i^T \\
&= \sum_{i=2}^{n} (\sum_{j=0}^{\infty} \mu_i^j) z_i z_i^T = \sum_{j=0}^{\infty} \hat{A}_r^j.
\end{aligned}
$$

□

Based on Lemma 3.4, we then prove Lemma 3.5 below.

Lemma 3.5. *Let G be a non-bipartite graph. Let u and v be two vertices in G. Then*

$$
R_{u,v} = \sum_{i=0}^{\infty} \left(\frac{1}{d_u} (\hat{A}^i)_{uu} + \frac{1}{d_v} (\hat{A}^i)_{vv} - \frac{2}{\sqrt{d_v d_u}} (\hat{A}^i)_{uv} \right).
$$

Proof. Observe that

$$
\begin{aligned}
&\left(\frac{1}{\sqrt{d_u}} 1_u - \frac{1}{\sqrt{d_v}} 1_v\right)^T \hat{A}_r^i \left(\frac{1}{\sqrt{d_u}} 1_u - \frac{1}{\sqrt{d_v}} 1_v\right) \\
= &\left(\frac{1}{\sqrt{d_u}} 1_u - \frac{1}{\sqrt{d_v}} 1_v\right)^T \hat{A}^i \left(\frac{1}{\sqrt{d_u}} 1_u - \frac{1}{\sqrt{d_v}} 1_v\right)
\end{aligned}
$$

for all $i \geq 0$ as $(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v)^T z_1 = 0$. We use this equation to alternatively express the effective resistance.

$$\begin{aligned}
R_{u,v} =& (\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v)^T \hat{L}^+ (\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v) \\
=& \sum_{i=0}^{\infty} (\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v)^T \hat{A}_r^i (\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v) && \text{(Lemma 3.4)} \\
=& \sum_{i=0}^{\infty} (\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v)^T \hat{A}^i (\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v) && \text{(Above observation)} \\
=& \sum_{i=0}^{\infty} \left(\frac{1}{d_u}(\hat{A}^i)_{uu} + \frac{1}{d_v}(\hat{A}^i)_{vv} - \frac{2}{\sqrt{d_u d_v}}(\hat{A}^i)_{uv} \right)
\end{aligned}$$

□

Now, we finish proving Theorem 3.3 as follows.

Theorem 3.3. *Let G be a non-bipartite graph. Let $u, v \in V$. Let $\|\nabla \phi_l\| \leq \alpha$ and $\max\{\|\nabla \psi_l\|, 1\} \leq \beta$. Let $d_{\min} = \min\{d_u, d_v\}$ and $d_{\max} = \max\{d_u, d_v\}$. Let $\max\{|\mu_2|, |\mu_n|\} \leq \mu$. Then*

$$\left\| \frac{\partial h_u^{(r)}}{\partial x_v} \right\| \leq (2\alpha\beta)^r \frac{d_{\max}}{2} \left(\frac{2}{d_{\min}} \left(r + 1 + \frac{\mu^{r+1}}{1-\mu} \right) - R_{u,v} \right)$$

Proof. Now, we will combine the equation for effective resistance of Lemma 3.5 with the bound on the Jacobian matrix of Lemma 3.2. This gives us the bound

$$\begin{aligned}
\left\| \frac{\partial h_u^{(r)}}{\partial x_v} \right\| &\leq (2\alpha\beta)^r \sum_{l=0}^{r} (\hat{A}^l)_{uv}. \\
&\leq (2\alpha\beta)^r \cdot \frac{\sqrt{d_u d_v}}{2} \cdot \left(\frac{1}{d_u} \sum_{l=0}^{\infty} (\hat{A}^l)_{uu} + \frac{1}{d_v} \sum_{l=0}^{\infty} (\hat{A}^l)_{vv} - \frac{2}{\sqrt{d_u d_v}} \sum_{l=r+1}^{\infty} (\hat{A}^l)_{uv} - R_{u,v} \right) \\
&\leq (2\alpha\beta)^r \cdot \frac{d_{\max}}{2} \cdot \left(\frac{1}{d_u} \sum_{l=0}^{\infty} (\hat{A}^l)_{uu} + \frac{1}{d_v} \sum_{l=0}^{\infty} (\hat{A}^l)_{vv} - \frac{2}{\sqrt{d_u d_v}} \sum_{l=r+1}^{\infty} (\hat{A}^l)_{uv} - R_{u,v} \right)
\end{aligned}$$

We now simplify some of the terms in this bound. First, we partition the sums in the right-hand side of this equation as

$$\begin{aligned}
& \frac{1}{d_u} \sum_{l=0}^{\infty} (\hat{A}^l)_{uu} + \frac{1}{d_v} \sum_{l=0}^{\infty} (\hat{A}^l)_{vv} - \frac{2}{\sqrt{d_u d_v}} \sum_{l=r+1}^{\infty} (\hat{A}^l)_{uv} \\
=& \left(\frac{1}{d_u} \sum_{l=0}^{r} (\hat{A}^l)_{uu} + \frac{1}{d_v} \sum_{l=0}^{r} (\hat{A}^l)_{vv} \right) \\
&+ \left(\frac{1}{d_u} \sum_{l=r+1}^{\infty} (\hat{A}^l)_{uu} + \frac{1}{d_v} \sum_{l=r+1}^{\infty} (\hat{A}^l)_{vv} - \frac{2}{\sqrt{d_u d_v}} \sum_{l=r+1}^{\infty} (\hat{A}^l)_{uv} \right) \\
=& \left(\frac{1}{d_u} \sum_{l=0}^{r} (\hat{A}^l)_{uu} + \frac{1}{d_v} \sum_{l=0}^{r} (\hat{A}^l)_{vv} \right) \\
&+ \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v \right)^T \sum_{l=r+1}^{\infty} \hat{A}^l \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v \right)
\end{aligned}$$

Let $\mu = \max\{|\mu_2|, |\mu_n|\}$. We can bound the second term in the above equation using the ***Courant-Fischer Theorem***, which says for a symmetric matrix B with maximum eigenvalue $\lambda_{\max}$ and any vector x, one has that $x^T B x \leq x^T x \cdot \lambda_{\max}$. Then, we have that

$$
\begin{aligned}
&\left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v\right)^T \sum_{l=r+1}^{\infty} \hat{A}^l \left(\frac{1}{\sqrt{d_u}}1_u - \frac{1}{\sqrt{d_v}}1_v\right) \\
&\leq \left(\frac{1}{d_u} + \frac{1}{d_v}\right) \sum_{l=r+1}^{\infty} \mu^l \;\leq\; \mu^{r+1}\left(\frac{1}{d_u} + \frac{1}{d_v}\right) \sum_{l=0}^{\infty} \mu^l \\
&\leq \mu^{r+1}\left(\frac{1}{d_u} + \frac{1}{d_v}\right)\frac{1}{1-\mu} \qquad \text{(as } \mu \in (-1,1)\text{)} \\
&\leq \mu^{r+1}\frac{2}{d_{\min}}\frac{1}{1-\mu}
\end{aligned}
$$

We now bound the first term. Again, we rely on the Courant-Fischer theorem, and note that $\hat{A}^l_{uu} = 1_u^T \hat{A}^l 1_u$; however, as $1_u^T z_1 \neq 0$, we only get a bound of $\hat{A}^l_{uu} \leq 1 \cdot 1_u^T 1_u = 1$. Thus,

$$
\frac{1}{d_u}\sum_{l=0}^{r}(\hat{A}^l)_{uu} + \frac{1}{d_v}\sum_{l=0}^{r}(\hat{A}^l)_{vv} \leq \frac{2}{d_{\min}}(r+1).
$$

□

A.4. Proof of Theorem 4.1

In this section, we prove Theorem 4.1, which gives a formula for how much the effective resistance changes when an edge is added. Recall that our strategy is to apply Woodbury's formula to compute $(L + \frac{11^T}{n} + (1_u - 1_v)(1_u - 1_v)^T)^{-1}$. Before doing this, we provide a proof for Lemma 4.3.

Lemma 4.3 ((Ghosh et al., 2008))**.** *Let G be a connected graph. Then*

- $R_{u,v} = (1_u - 1_v)^T (L + \frac{11^T}{n})^{-1}(1_u - 1_v)$;
- $B^2_{u,v} = (1_u - 1_v)^T (L + \frac{11^T}{n})^{-2}(1_u - 1_v)$;
- $R_{tot} = n \cdot \mathrm{tr}(L + \frac{11^T}{n})^{-1} - n$.

Proof. By Equation (7) of (Ghosh et al., 2008), one has that

$$
L^+ = \left(L + \frac{11^T}{n}\right)^{-1} - \frac{11^T}{n}.
$$

Then, we have that

$$
(L^+)^2 = \left(L + \frac{11^T}{n}\right)^{-2} - \frac{11^T}{n}\left(L + \frac{11^T}{n}\right)^{-1} - \left(L + \frac{11^T}{n}\right)^{-1}\frac{11^T}{n} + \frac{11^T}{n}.
$$

Note that vectors of the form $1_u - 1_v$ are orthogonal to the all-ones vector 1, i.e., $(1_u - 1_v)^T 1 = 1^T(1_u - 1_v) = 0$. Hence

$$
R_{u,v} = (1_u - 1_v)^T L^+ (1_u - 1_v) = (1_u - 1_v)^T (L + \frac{11^T}{n})^{-1}(1_u - 1_v),
$$

and

$$
B^2_{u,v} = (1_u - 1_v)^T (L^+)^2 (1_u - 1_v) = (1_u - 1_v)^T (L + \frac{11^T}{n})^{-2}(1_u - 1_v).
$$

Now, by Theorem 3.9, one has that

$$
R_{\text{tot}} = n \cdot \mathrm{tr}\, L^+ = n \cdot \mathrm{tr}\left(\left(L + \frac{11^T}{n}\right)^{-1} - \frac{11^T}{n}\right) = n \cdot \mathrm{tr}\left(L + \frac{11^T}{n}\right)^{-1} - n.
$$

□

Theorem 4.1. *Let G be a connected graph with n vertices. Let $\{u, v\}$ be an edge not in G. The difference in total resistance after adding the edge $\{u, v\}$ to G is*

$$R_{tot}(G) - R_{tot}(G \cup \{u, v\}) = n \cdot \frac{B_{u,v}^2}{1 + R_{u,v}}$$

Proof of Theorem 4.1. Adding the edge $\{u, v\}$ to G changes the Laplacian from L to $L + (1_u - 1_v)(1_u - 1_v)^T$. Then, by Lemma 4.3, we can find the difference in the total resistance by considering difference of $R_{\text{tot}}(G) = n \cdot \text{tr}(L + \frac{11^T}{n})^{-1} - n$ and $R_{\text{tot}}(G \cup \{u, v\}) = n \cdot \text{tr}(L + \frac{11^T}{n} + (1_u - 1_v)(1_u - 1_v)^T)^{-1} - n$. The difference of these is the trace of the third term in Woodbury's formula, which simplifies to the quantity in the statement as follows.

$$\begin{aligned}
&R_{\text{tot}}(G) - R_{\text{tot}}(G \cup \{u, v\}) \\
=&n \cdot \text{tr}\left(L + \frac{11^T}{n}\right)^{-1} - n \cdot \text{tr}\left(L + \frac{11^T}{n} + (1_u - 1_v)(1_u - 1_v)^T\right)^{-1} \\
=&n \cdot \text{tr}\left(\left(1 + (1_u - 1_v)^T \left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)^{-1} \cdot \left(\left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)\left(\left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)^T\right) \\
=&n \cdot \underbrace{\left(1 + (1_u - 1_v)^T \left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)^{-1}}_{c} \cdot \text{tr}\left(\left(\left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)\left(\left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)^T\right).
\end{aligned}$$

For the coefficient term c, one has that

$$\left(1 + (1_u - 1_v)^T \left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right) = (1 + R_{u,v}).$$

For the trace term, one has that

$$\begin{aligned}
&\text{tr}\left(\left(\left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)\left(\left(L + \frac{11^T}{n}\right)^{-1} (1_u - 1_v)\right)^T\right) \\
&= (1_u - 1_v)\left(L + \frac{11^T}{n}\right)^{-2} (1_u - 1_v)^T \\
&= B_{u,v}^2
\end{aligned}$$

by the fact that $\text{tr}(xx^T) = x^T x$ for any vector x. □

B. Runtime Analysis of GTR

B.1. Asymptotic Analysis

The time complexity for GTR rewiring depends on the time it takes to (step 1) compute the effective resistance and biharmonic distance for each pair of vertices, (step 2) find the pair of vertices maximizing $R_{u,v}/(1 + B_{u,v}^2)$, and (step 3) update the effective resistance and biharmonic distance. If we are adding k edges to the graph, the naive implementation for GTR takes $O(n^3 + kn^2)$ time. We can compute L^+ and L^{2+} in $O(n^3)$ time using the singular value decomposition, which we can use to compute all pairs effective resistance and biharmonic distance in time $O(n^2)$. In total, step (1) would take $O(n^3 + n^2)$ time. Step (2) would take $O(n^2)$ time to iterate over all pairs of vertices. Finally, for step (3), we can update L^+ and L^{2+} in $O(n^2)$ time. This is because adding the edge $\{u, v\}$ to G only causes a constant-rank change to the Laplacian; the Laplacian changes from L to $L + (1_u - 1_v)(1_u - 1_v)^T$ and the squared Laplacian changes from L^2 to $L^2 + L(1_u - 1_v)(1_u - 1_v)^T + (1_u - 1_v)(1_u - 1_v)^T L + (1_u - 1_v)(1_u - 1_v)^T$. The pseudoinverse of L^+ and L^{2+} can then be updated in $O(n^2)$ using Woodbury's Formula (see Lemma 4.2). [3]

[3] In general, Woodbury's Formula cannot be used to update the pseudoinverse of a matrix; however, in the special case of adding an edge to a connected graph, it can be used to update the pseudoinverse of L and L^2. In short, this is because the vector $1_u - 1_v$ is orthogonal to the kernels of L and L^2. See the discussion in Appendix A.4.

However, more efficient implementations for GTR are possible thanks to nearly-linear time Laplacian solvers: algorithms for solving linear systems of the form $Lx = b$ in $O(m \operatorname{poly} \log n)$ time (Spielman & Teng, 2004; Jambulapati & Sidford, 2021). Using these algorithms, the pseudoinverses L^+ and L^{2+} could be computed in $O(n \cdot m \operatorname{poly} \log n)$ by using Laplacian solvers to find the columns of the matrix. Alternatively, all-pairs effective resistance and biharmonic distance can be estimated in time $O(m \operatorname{poly} \log n + n^2 \operatorname{poly} \log n)$ using an algorithm that combines Laplacian solvers and Johnson-Lindenstrauss random projection (Spielman & Srivastava, 2011).

B.2. Experimental Analysis

We implemented the GTR algorithm in PyTorch Geometric; our analysis is available here: `https://github.com/blackmit/gtr_rewiring`. The fastest implementation of GTR we found was to use the naive algorithm; this is because we can calculate the pseudoinverse of the Laplacian using a GPU. The following table contains the amount of time needed to compute 50 edges using each algorithm.

	MUTAG	PROTEINS	ENZYMES	IMDB-BINARY	REDDIT-BINARY	COLLAB
FoSR	0.10	1.00	0.37	0.51	199.20	15.94
GTR	12.86	68.10	35.76	57.23	349.98	423.79

Table 2. Time in seconds to compute 50 additional edges to add to the graph using both FoSR and GTR

C. Counterexamples to the Optimality of GTR.

Theorem 4.1 proves that GTR adds the single edge that most decreases the total resistance; however, GTR will not necessarily add the k edges that most decrease total resistance for $k > 1$. Figure 5 gives an example where this is the case.

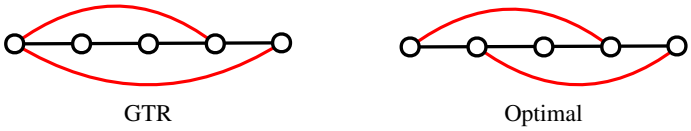


Figure 5. The path on 5 vertices is a counterexample showing that GTR does not add the k edges that most decrease R_{tot} when $k > 1$. Left: The two edges added by GTR. GTR first adds the edge connecting the first and last vertex in the path. The total resistance of this graph is $R_{\text{tot}} \approx 8.18$. Right: The two edges that most deceases the total resistance. The total resistance of this graph is $R_{\text{tot}} \approx 7.67$.

The amount an edge decreases the total resistance can increase as more edges are added to the graph. Figure 6 gives such an example. This can be interpreted as an edge becoming more important for the global topology of the graph as the graph changes.

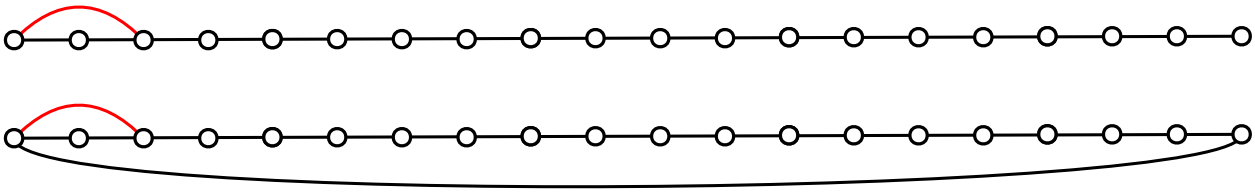


Figure 6. The path on 20 vertices is an example showing that the amount an edge decreases the total resistance is not monotonic. Top: Adding the red edge would decrease the total resistance by ≈ 30.33. Bottom: After adding the edge connecting the first and last vertex in the path, adding the red edge would decrease the total resistance by ≈ 40.17

D. Experimental Details

We use the same configuration of hyperparameters as in (Karhadkar et al., 2022). We use randomly generated $80\%/10\%/10\%$ train/validation/test splits of the data. We use the Adam optimizer and the `ReduceLROnPlateau` scheduler in Torch that reduces the learning rate after 10 epochs without an improvement in the validation accuracy. We use a stopping patience of 100 epochs of the validation loss. For the hyperparameter search, we consider average accuracies over 10 randomly

generated splits of the data. For the test results, we report the average test accuracy and 95% confidence intervals over 100 randomly generated splits.

Table 3. Number of edges added by GTR of FoSR for each dataset. Note that FoSR only contains the number of edges when our run in the edge ablation experiment (Section 5.3) outperformed the run in (Karhadkar et al., 2022). The number of edges added by FoSR for all other experiments can be found in the appendix to (Karhadkar et al., 2022).

GCN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
GTR	45	25	20	5	5	5
R-GCN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
GTR	50	10	40	20	40	25
GIN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
GTR	25	5	5	5	15	25
R-GIN						
Rewiring	Mutag	Proteins	Enzymes	Reddit-Binary	IMDB-Binary	Collab
FoSR	-	20	-	25	50	20
GTR	15	5	50	5	20	30

Table 4. Hyperparameters for Graph Classifcation. These are consistent across all GNN types. These are the same as used in the experiments in (Karhadkar et al., 2022)

Hyperparameters	
Number of Hidden Layers	4
Dimension of Hidden Layers	64
Dropout	0.5
Learning Rate	1.0×10^{-3}

E. Total Resistance vs. Number of Edges Added

Figure 7 shows the decrease in average total resistance across a dataset as edges are added to a graph by GTR or FoSR. GTR seems to outperform FoSR in decreasing total resistance.

F. Edge Ablation

Figure 8 shows the effect of adding between 0 and 50 edges on the classification accuracy across different graph classification datasets. We used the R-GIN architecture for the experiments and followed the same experimental procedure as described in Appendix D.

We see a variety of behaviors across the datasets. For some datasets like Proteins or IMDB-Binary, we see an initial large jump in accuracy after adding a few edges, but generally see little improvement by adding more edges. For Enzymes, the accuracy almost only increases as we add more edges, suggesting that the optimal number of edges was greater than the maximum of 50 we tested. The variety of behaviors suggest that there is no optimal number of edges to add that will maximize performance across datasets. Our experiments also suggest that, while adding some number of edges helps for all datasets, performance doesn't continue to increase as more edges are added.

For almost all datasets, we see the greatest rate of improvement in accuracy after adding a few edges. A possible explanation might be that the rate total resistance decreases is greatest for the first few edges added, as we see in Figure 7.

G. Hidden Dimension Ablation

Figure 9 shows the effect of adding between 0 and 30 edges and using a hidden dimension of 32, 64, or 128 on graph classification accuracy. We used the R-GIN architecture for these experiments and followed the same experimental

procedure as described in Appendix D. Generally, we see that both rewiring and increasing the hidden dimension improve the classification accuracy.

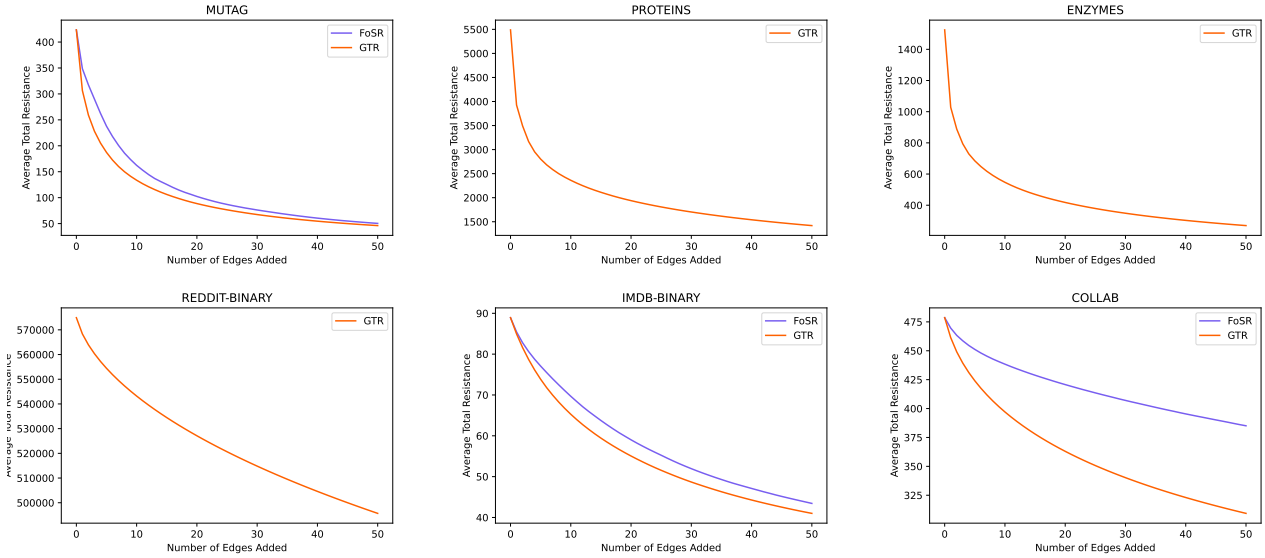


Figure 7. Plots of average total resistance vs. number of edges added. For disconnected graphs, plots show the sum of effective resistances for all pairs of vertices in the same connected component, as effective resistance between vertices in different connected components is ill-defined. As FoSR adds edges between different connected components and GTR does not, it would not be meaningful to compare total effective resistance for datasets with disconnected graphs (i.e., Proteins, Enzymes, and IMDB-Binary) as FoSR may connect these disconnected componets, which is why FoSR curves are not reported for these datasets.

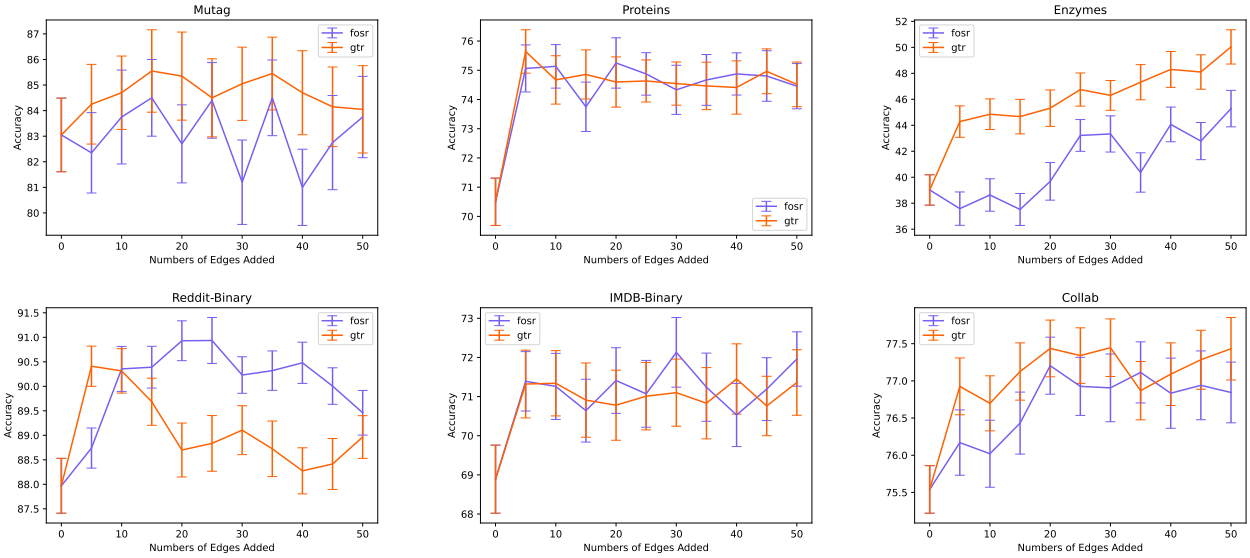


Figure 8. Plots of graph classification accuracy vs. number of edges added by GTR or FoSR.

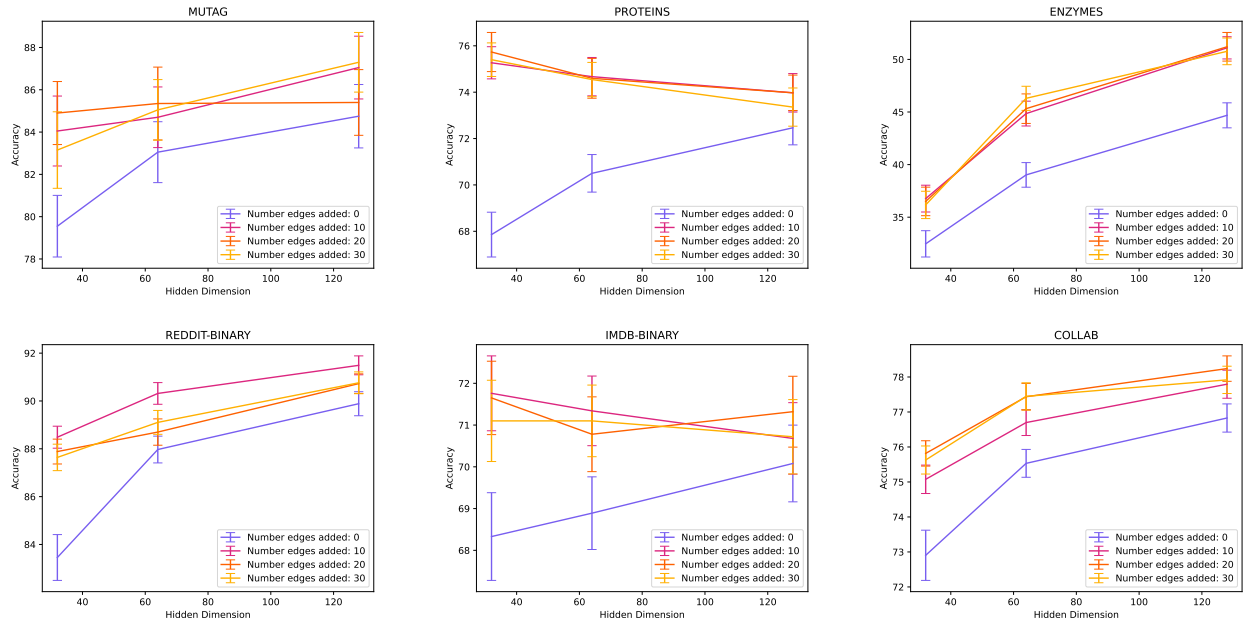


Figure 9. Plots of graph classification accuracy vs. hidden dimension for a variable number of edges added by GTR.