Towards Optimal Effective Resistance Estimation

Rajat Vadiraj Dwaraknath
Stanford University
rajatvd@stanford.edu

Ishani Karmarkar
Stanford University
ishanik@stanford.edu

Aaron Sidford
Stanford University
sidford@stanford.edu

Abstract

We provide new algorithms and conditional hardness for the problem of estimating effective resistances in n-node m-edge undirected, expander graphs. We provide an $\widetilde{O}(m\epsilon^{-1})$-time algorithm that produces with high probability, an $\widetilde{O}(n\epsilon^{-1})$-bit sketch from which the effective resistance between any pair of nodes can be estimated, to $(1 \pm \epsilon)$-multiplicative accuracy, in $\widetilde{O}(1)$-time. Consequently, we obtain an $\widetilde{O}(m\epsilon^{-1})$-time algorithm for estimating the effective resistance of all edges in such graphs, improving (for sparse graphs) on the previous fastest runtimes of $\widetilde{O}(m\epsilon^{-3/2})$ [Chu et. al. 2018] and $\widetilde{O}(n^2\epsilon^{-1})$ [Jambulapati, Sidford 2018] for general graphs and $\widetilde{O}(m + n\epsilon^{-2})$ for expanders [Li, Sachdeva 2022]. We complement this result by showing a conditional lower bound that a broad set of algorithms for computing such estimates of the effective resistances between all pairs of nodes require $\widetilde{\Omega}(n^2\epsilon^{-1/2})$-time, improving upon the previous best such lower bound of $\widetilde{\Omega}(n^2\epsilon^{-1/13})$ [Musco et. al. 2017]. Further, we leverage the tools underlying these results to obtain improved algorithms and conditional hardness for more general problems of sketching the pseudoinverse of positive semidefinite matrices and estimating functions of their eigenvalues.

1 Introduction

In a weighted, undirected graph G the *effective resistance* between a pair of vertices a and b, denoted $r_G(a, b)$, is defined as the energy of a unit of electric current sent from a to b in the natural resistor network induced by G. Effective resistances arise for a broad set of graph processing tasks and have multiple equivalent definitions. For example, $r_G(a, b)$ is proportional to the expected roundtrip commute time between a and b in the natural random walk induced on the graph and when $\{a, b\}$ is an edge in the graph, it is proportional to the probability that the edge is in a random spanning tree.

Effective resistances also form a particular class of metrics on the vertices [1,2] and are a key measure of proximity between vertex pairs. Correspondingly, effective resistances can arise in a variety of data analysis tasks. For example, effective resistances have been used in social network analysis for measuring edge centrality in social networks [3] as well as for measuring chemical distances [4].

Effective resistances have a broad range of algorithmic implications. Sampling edges of a graph using effective resistance is known to efficiently produce cut and spectral sparsifiers (sparse graphs which approximately preserve cuts, random walk properties, and more) [5–7]. Effective resistance-based graph sparsifiers have also been applied to develop fast graph attention neural networks [8], to design graph convolutional neural networks for action recognition [9], to sample from Gaussian graphical models [10], and beyond [11, 12]. Effective resistances have also been used in algorithms for maximum flow problems, [13–16, 16–18], sampling random spanning trees [19–21], and graph partitioning [22, 23]. More recently, effective resistances have also been used to analyze the problem of oversquashing in GNNs and in designing algorithms to alleviate oversquashing [24–26] and have been applied to increase expressivity when incorporated as edge features into certain GNNs [27].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Algorithms. Given the broad utility of effective resistances, there have been many methods for estimating and approximately compressing them [5, 28–31]. In this paper, our main focus is the following effective resistance estimation problem. (We use $x \approx_\epsilon y$ as shorthand for $(1-\epsilon)y \leqslant x \leqslant (1+\epsilon)y$ and assume all edge weights in graphs are positive. See Section 2 for notation more broadly).

Definition 1 (Effective Resistance Estimation Problem)**.** *In the* effective resistance estimation problem *we are given an undirected, weighted graph $G = (V, E, w)$, a set of vertex pairs $S \subseteq V \times V$, and $\epsilon \in (0, 1)$ and must output $\tilde{r} \in \mathbb{R}^S$ such that with high probability (whp.), $\tilde{r}_{(a,b)} \approx_\epsilon r_G(a, b)$ for all $(a, b) \in S$.*

The state-of-the-art runtimes for solving the effective resistance estimation problem on n-node, m-edge graphs are given in Table 2. To contextualize these results, consider the special case of estimating the effective resistance of a graph's edges, i.e., when $S = E$. This special case appears in many of the aforementioned applications, e.g., [13–15, 19, 20]. The state-of-the-art runtimes for this problem are $\widetilde{O}(n^2\epsilon^{-1})$ due to [28] and $\widetilde{O}(m\epsilon^{-1.5})$ due to [30]. A major open problem is whether improved runtimes, e.g., $\widetilde{O}(m\epsilon^{-1})$ (which would subsume prior work), are attainable.

One of the main results of this paper is resolving this open problem in the case of well-connected graphs, i.e., expanders. Expanders are a non-trivial, previously studied, special case that is often the first step or a key component for developing more general algorithms [32]. In particular we provide an $\widetilde{O}(m\epsilon^{-1}\overline{\kappa}(G))$ time algorithm where $\overline{\kappa}$ is a measure of the graph's connectivity. Previously, the only non-trivial improvement in this setting was an independently obtained runtime of $\widetilde{O}(m + n\epsilon^{-2}(\overline{\kappa}(G))^{3/2})$ for graphs with $\overline{\kappa}(G) = \widetilde{O}(1)$ [29]. We improve upon the result of [29] for sparse enough graphs and, as we explain in Section 1.1, we can almost match the result of [29] (up to an $m^{o(1)}$ factor) in dense graphs. [1]

Interestingly, we obtain our main result by providing new effective resistance sketch algorithms.

Definition 2 (Effective Resistance Sketch)**.** *We call a randomized algorithm an $(\mathcal{T}_{\mathrm{s}}, \mathcal{T}_{\mathrm{q}}, s)$-*effective resistance sketch algorithm *if given an input n-node, m-edge undirected, weighted graph $G = (V, E, w)$ and $\epsilon \in (0, 1)$ in time $O(\mathcal{T}_{\mathrm{s}}(G, \epsilon))$ it creates a binary string of length $O(s(G, \epsilon))$ from which when queried with any $a, b \in V$, it outputs $\tilde{r}_{a,b} \approx_\epsilon r_G(a, b)$ whp. in time $O(\mathcal{T}_{\mathrm{q}}(G, \epsilon))$.*

Effective resistance sketching algorithms immediately imply algorithms for the effective resistance estimation problem. We obtain our result by obtaining an $(\widetilde{O}(m\epsilon^{-1}), \widetilde{O}(1), \widetilde{O}(n\epsilon^{-1}))$-effective resistance sketch algorithm for expanders (see Section 1.1 for a comparison to prior work).

Lower Bounds. Given the central role of effective resistance estimation and the challenging open-problem of determining its complexity, previous work has sought complexity theoretic lower bounds for the problem. [33] showed a conditional lower bound of $\Omega(n^2\epsilon^{-1/13})$ for the problem by showing that an algorithm that computes effective resistances in $(n^2\epsilon^{-1/13+\delta})$ for some $\delta > 0$ time could be used to obtain a subcubic algorithm for the *triangle detection problem*, that we define below.

Definition 3 (Triangle Detection Problem)**.** *Given an n-node undirected unweighted graph $G = (V, E)$, determine whether there are distinct $a, b, c \in V$ with $\{a, b\} \in E$, $\{b, c\} \in E$ and $\{c, a\} \in E$.*

Currently, the only known subcubic algorithms for the triangle detection problem leverage fast matrix multiplication (FMM) and therefore their practical utility (in the worst case) is questionable.

Theorem 1 (Informal, [34])**.** *Given an algorithm which solves the triangle detection problem in subcubic time, we can produce a subcubic algorithm, which only performs combinatorial operations and uses the triangle detection algorithm, for Boolean matrix multiplication (BMM) and additional problems which currently are not known to be solvable subcubicly without FMM.*

This theorem implies that any subcubic algorithm for triangle detection that doesn't use FMM implies a subcubic BMM algorithm that doesn't use FMM. Consequently, subcubic triangle detection is a common hardness assumption used to illustrate barriers towards improving non-FMM based methods, e.g., the effective resistance estimation algorithms of this paper.

[1]While our algorithms for the effective resistance estimation problem (Definition 1) were obtained indpendently, our writing and discussion of effective resistance sketch algorithms (Definition 1) was informed by their paper. We provide a more complete comparison of our work with prior results in Table 1.

In this paper we take a key step towards closing the gap between the best known running times for effective resistance estimation and lower bounds by improving the conditional lower bound of $\Omega(n^2\epsilon^{-1/13})$ to $\widetilde{\Omega}(n^2\epsilon^{-1/2})$ for randomized algorithms. We show this lower bound holds *even for expander graphs*, and hence our effective resistance estimation algorithm (as well as [28]) are optimal up to an $\epsilon^{-1/2}$-factor among non-FMM based algorithms, barring a major breakthrough in BMM.

Broader Linear Algebraic Tools. The effective resistance between vertex a and vertex b in a graph G has a natural linear algebraic formulation. For all $a, b \in V$ it is known that $r_G(a,b) = \vec{\delta}_{a,b}\mathbf{L}_G^{\dagger}\vec{\delta}_{a,b}$, where $\mathbf{L}_G \in \mathbb{R}^{V\times V}$ is a natural matrix associated with G known as the *Laplacian matrix* and $\vec{\delta}_{a,b} = e_a - e_b$ (see Section 2 for notation). Thus, sketching effective resistances can be viewed as problems of preserving information about subsets of entries of the pseudoinverse of a Laplacian.

Both our algorithms and lower bounds develop more general tools for handling related problems for more general (not-necessarily Laplacian) matrices. On the algorithmic side, we show our techniques can also lead to algorithms and data structures for computing certain quadratic forms involving well-conditioned SDD and PSD matrices. On the hardness side, we show our techniques also improve triangle detection hardness bounds for estimating various properties of the singular values of a matrix.

Paper Organization. In the remainder of the introduction we provide a more precise statement and comparison of our results in Section 1.1. In the remainder of the paper we cover preliminaries in Section 2, present upper bounds in Section 3, and present lower bounds in Section 4. Omitted proofs and additional discussion of related proofs are deferred to the supplementary material.

1.1 Our Results

Algorithms. Here we outline our main algorithmic results pertaining to effective resistance sketching and estimation, and in Section 3 we describe a extensions of our work to broader linear algebraic problems involving SDD and PSD matrices. Our main algorithmic result is a new efficient effective resistance sketch for *expanders*, a term which is used to refer to graphs with $\widetilde{\Omega}(1)$-expansion.

Definition 4 (Expander)**.** *For $\alpha > 0$, we say that a graph $G = (V, E, w)$ has α-expansion if $\alpha \leqslant \phi(G)$, where $\phi(G)$ denotes the conductance of G and is defined as*

$$\phi(G) := \min_{S\subseteq V, S\notin\{0,V\}} \frac{\sum_{\{u,v\}:u\in S, v\in V\setminus S} w_{u,v}}{\min\left(\sum_{u\in S} d_u, \sum_{v\in V\setminus S} d_u\right)}, \text{ where } d_u := \sum_{\{u,v\}\in E} w_u.$$

Theorem 2. *There is an $(\widetilde{O}(m\epsilon^{-1}), \widetilde{O}(1), \widetilde{O}(n\epsilon^{-1}))$-effective resistance sketch algorithm for graphs with $\widetilde{\Omega}(1)$-expansion.*

Table 1 summarizes and compares our Theorem 2 to previous work on effective resistance sketches, including naive algorithms to explicitly compute the pseudoinverse of the Laplacian of G, which can be computed in $O(n^\omega)$ time using FMM or $\widetilde{O}(mn)$ time using a Laplacian system solver (labeled Solver). [2] A $(\mathcal{T}_\mathrm{s}, \mathcal{T}_\mathrm{q}, s)$ effective-resistance sketch algorithm implies an $O(\mathcal{T}_\mathrm{s} + |S|\mathcal{T}_\mathrm{q})$ algorithm for the effective resistance estimation problem. Hence, Theorem 2 implies the following.

Theorem 3 (Effective Resistance Estimation on Expanders)**.** *There is an $\widetilde{O}(m\epsilon^{-1} + |S|)$ time algorithm which solves the effective resistance estimation problem for graphs with $\widetilde{\Omega}(1)$-expansion.*

Effective resistance sketches are a common approach to solving the effective resistance estimation problem; but there are also approaches to the problem that do not explicitly construct effective resistance sketches. Table 2 summarizes prior work on effective resistance estimation more broadly.

There has been a long line of research on the problem of computing sketches and sparsifiers of graph Laplacians [5,28,30,36] (i.e., computing a sparse graph G' such that quadratic forms in the Laplacian of G' approximately preserves quadratic forms in the Laplacian of G). Building on this work, [30] showed there is an algorithm which processes a graph G on n nodes and m edges in $O(m^{1+o(1)})$ time and produces a sparse sketch graph H with only $\widetilde{O}(n\epsilon^{-1})$ edges such that $r_G(a,b) \approx r_H(a,b)$

[2] $\omega \leqslant n^{2.37188}$ [35] denotes the fast matrix multiplication constant.

for all $a, b \in V$. Consequently, any algorithm which runs in $\widetilde{O}(m\epsilon^{-c})$ on expanders can be improved to run in $\widetilde{O}(m^{1+o(1)} + n\epsilon^{-(c+1)})$ on expanders simply by running the algorithm on H instead of G.

Table 1: Effective Resistance Sketch

Citation	$\mathcal{T}_p$	$\mathcal{T}_q$	s
FMM	n^{ω}	1	n^2
Solver	nm	1	n^2
[5]	$m\epsilon^{-2}$	ϵ^{-2}	$n\epsilon^{-2}$
[28]	$n^2\epsilon^{-1}$	1	$n\epsilon^{-1}$
[29]	$m + n\epsilon^{-2}$	1	$n\epsilon^{-1}$
This Paper	$m\epsilon^{-1}$	1	$n\epsilon^{-1}$

Table 2: Effective Resistance Estimation

Citation	Runtime	S Restrictions
FMM	$n^{\omega} + \lvert S\rvert$	None
Solver	$nm + \lvert S\rvert$	None
[5]	$n^2\epsilon^{-2}$	$S = V \times V$
[28]	$n^2\epsilon^{-1}$	$S = V \times V$
[19]	$m + (n + \lvert S\rvert)\epsilon^{-2}$	None
[37]	$m + \lvert S\rvert\, \epsilon^{-2}$	None
[30]	$m\epsilon^{-1.5}$	$S = E$
[29]	$m + n\epsilon^{-2} + \lvert S\rvert$	None
This Paper	$m\epsilon^{-1} + \lvert S\rvert$	None

Summary of prior work on Effective Resistance Sketch (Table 1) and Effective Resistance Estimation (Table 2) algorithms on n-node, m-edge expanders. All time and space complexities are reported up to to $\widetilde{O}(\cdot)$. The methods of This Paper and [29] apply only to expanders; however, the remaining works apply to general graphs. As discussed, when $m^{1+o(1)} + n\epsilon^{-(c+1)} = o(m\epsilon^{-c})$, any runtime dependence on $m\epsilon^{-c}$ in the table can be improved to a dependence on $m^{o(1)+1} + n\epsilon^{-(c+1)}$.

Lower Bounds For the effective resistance estimation problem, [33] showed that any combinatorial algorithm (i.e., an algorithm that uses only combinatorial operations in the sense of Theorem 1) which solves the effective resistance estimation problem for $S = V \times V$ in $\widetilde{O}(n^2\epsilon^{-1/13+\delta})$ for some $\delta > 0$, would imply a combinatorial subcubic deterministic algorithm which detects a triangle in an n-node undirected unweighted graph. We improve on their result, as follows.

Theorem 4. *Given a combinatorial algorithm which solves the effective resistance estimation problem for $S = V \times V$ on graphs with $\widetilde{\Omega}(1)$-expansion in $\widetilde{O}(n^2\epsilon^{-1/2+\delta})$ time for $\delta > 0$, we can produce a randomized combinatorial algorithm which solves the triangle detection problem on an n-node graph in $\widetilde{O}(n^{3-2\delta})$ time whp.*

Theorem 4 implies an $\widetilde{\Omega}(n^2\epsilon^{-1/2})$ randomized conditional lower bound for the problem of estimating effective resistances of all pairs of nodes in an undirected unweighted expander graph, while [33] shows only an $\Omega(n^2\epsilon^{-1/13})$ lower bound. As we show in Section 4, by a simple reduction we can extend any $\widetilde{\Omega}(n^2\epsilon^{-c})$ lower bound for the all-pairs effective resistance problem to a $\widetilde{\Omega}(m\epsilon^{-c})$ lower bound for the all-edges effective resistance problem. Consequently, our result also yields a $\widetilde{\Omega}(m\epsilon^{-1/2})$ randomized lower bound for the problem of estimating effective resistances of all edges in an undirected expander graph.

In addition to conditional lower bounds for effective resistance estimation, we also improve on existing conditional lower bounds for the problem of estimating spectral sums that we define below.

Definition 5 (Spectral Sum)**.** *For $f : \mathbb{R}^+ \to \mathbb{R}^+$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ with singular values $\sigma_1(\mathbf{A}) \leqslant \sigma_2(\mathbf{A}) \leqslant \cdots \leqslant \sigma_n(\mathbf{A})$, we define the spectral sum $\mathcal{S}_f : \mathbb{R}^{n\times n} \to \mathbb{R}^+$ as $\mathcal{S}_f(\mathbf{A}) := \sum_{i=1}^{n} f(\sigma_i(\mathbf{A}))$.*

[33] showed that for several spectral sums $\mathcal{S}_f$, any combinatorial algorithm that outputs $Y \approx_\epsilon \mathcal{S}_f(\mathbf{A})$ in $(n^{\gamma}\epsilon^{-c})$ time on an $n \times n$ PSD matrix would imply an $O(n^{\gamma+\alpha c})$ time combinatorial algorithm which solves the triangle detection problem, where the scaling α varies depending on the specific $\mathcal{S}_f$ (see Table 3). We build on their results to show improved randomized conditional lower bounds for several spectral sum estimation problems, as presented in Theorem 5 below.

Theorem 5. *Given a combinatorial algorithm which on input $\mathbf{B} \in \mathbb{R}^{n\times n}$ outputs a spectral sum estimate $Y \approx_\epsilon \mathcal{S}_f(\mathbf{B})$ in $O(n^{\gamma}\epsilon^{-c})$ time with $\gamma \geqslant 2$ for the spectral sums in Table 3, we can produce a randomized combinatorial algorithm that can detect a triangle in an n-node graph whp. in $\widetilde{O}(n^{\gamma+\alpha c})$ time, where α is a scaling that depends on properties of the function f (see Table 3 for values of α for several spectral sums.)*

Spectral Sum	[33]		This Paper	
	TD Runtime	Lower Bound	TD Runtime	Lower Bound
Schatten 3-norm	$n^{\gamma+4c}$	$n^2\epsilon^{-1/4}$	$n^{\gamma+5c/2}$	$n^2\epsilon^{-2/5}$
Schatten p-norm, $p \neq 1, 2,$	$n^{\gamma+13c}$	$n^2\epsilon^{-1/13}$	$n^{\gamma+10c}$	$n^2\epsilon^{-1/10}$
SVD Entropy	$n^{\gamma+6c}$	$n^2\epsilon^{-1/6}$	$n^{\gamma+5c}$	$n^2\epsilon^{-1/5}$
Log Determinant	$n^{\gamma+6c}$	$n^2\epsilon^{-1/6}$	$n^{\gamma+5c}$	$n^2\epsilon^{-1/5}$
Trace of Exponential	$n^{\gamma+13c}$	$n^2\epsilon^{-1/13}$	$n^{\gamma+10c}$	$n^2\epsilon^{-1/10}$

Table 3: Runtimes for the triangle detection (TD) problem in an n-node graph using algorithms that produce $(1 \pm \epsilon)$ multiplicative approximations to various spectral sums in $O(n^\gamma \epsilon^{-c})$ time. The second columns contain the best achievable runtimes for $\gamma = 2$ that do not use FMM, barring a breakthrough in subcubic triangle detection. All runtimes are reported up to $\widetilde{O}(\cdot)$.

2 Preliminaries

General notation. We use $\mathbf{A}_{i,j}$ to denote the (i, j)-th entry of $\mathbf{A}$. For $\mathbf{A} \in \mathbb{R}^{n\times n}$ we use $\lambda(\mathbf{A})$ for its spectrum, $\lambda_i(\mathbf{A})$ and $\sigma_i(\mathbf{A})$ for its i-th smallest eigenvalue and singular value respectively, and $\rho(\mathbf{A}) := |\lambda_n(\mathbf{A})|$ for its spectral radius. $\|\cdot\|_p$ denotes the ℓ_p-norm. When $\mathbf{A}$ is PSD, $\lambda_{\min}(\mathbf{A})$ denotes its smallest nonzero eigenvalue and $\kappa(\mathbf{A}) := \lambda_n(\mathbf{A})/\lambda_{\min}(\mathbf{A})$ denotes its pseudo-condition number. We use $\langle\cdot,\cdot\rangle$ for the Euclidean inner product, $\mathbb{1}$ for the all ones vector, and e_i for the i-th standard basis vector. We define $\vec{\delta}_{i,j} := e_i - e_j$ and $[k] := \{1, ..., k\}$. We use $x \approx_\epsilon y$ as shorthand for $(1-\epsilon)y \leqslant x \leqslant (1+\epsilon)y$. For $v \in \mathbb{R}^n$, we use $v[i : j]$ for the sub-vector from index i to j. We use $v \perp w$ to indicate that $v, w \in \mathbb{R}^n$ are orthogonal (i.e., $v^\top w = 0$).

Graphs. We use $G = (V, E, w)$ to denote a weighted undirected graph on V with edges E and edge weights $w \in \mathbb{R}^E_{>0}$ (or $G = (V, E)$ if unweighted). We use $\mathbf{A}_G$ to denote its (weighted) adjacency matrix $(\mathbf{A}_G)_{u,v} = w_{u,v}$ for $u, v \in V \times V$ and $\mathbf{D}_G$ to denote its diagonal (weighted) degree matrix $(\mathbf{D}_G)_u = \sum_{\{u,v\}\in E} w_{u,v}$ for $u \in V$ (treated as $w_{u,v} = 1$ if G is unweighted). We define $\mathbf{L}_G := \mathbf{D}_G - \mathbf{A}_G$ as its graph Laplacian. $d_{\max}(G)$ and $d_{\min}(G)$ refer to the max and min diagonal entry in $\mathbf{D}_G$. We may drop the argument or subscript G if it is clear from context. The effective resistance between nodes $i, j \in V$ is denoted $r_G(i, j) = \vec{\delta}^\top_{i,j}\mathbf{L}^\dagger_G\vec{\delta}_{i,j}$. We assume all input graphs are connected, as effective resistances can be computed separately on connected components. We use $\mathbf{B}_G$ to denote the $E \times V$ edge-incidence matrix of G, where $(\mathbf{B}_G)_{e,u} = 1$ and $(\mathbf{B}_G)_{e,v} = -1$ for all $e \in E$, and $e = \{u, v\}$, with $\mathbf{B}_{Ge,l} = 0$ for all $l \neq u$ and $l \neq v$.

Symmetric Diagonally Dominant (SDD) Matrices A matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is SDD if it can be decomposed as $\mathbf{M} = \mathbf{D_M} - \mathbf{A_M}$, where the $\mathbf{D_M}$ is a diagonal matrix with non-negative entries and $\mathbf{A}_M$ is a matrix with zeros on the diagonal such that $d_{i,i} > \sum_{j=1}^n |a_{i,j}|$. We define the normalization of $\mathbf{M}$ as $\mathbf{N}_M := \mathbf{D}_M^{-1/2}\mathbf{M}\mathbf{D}_M^{-1/2}$. Throughout this paper, we assume, without loss of generality that $\mathbf{D}_M$ has strictly positive entries on the diagonal (otherwise, we can simply remove an entire row and column of zeros). We use $d_{\max}(\mathbf{M})$ and $d_{\min}(\mathbf{M})$ to denote the max and min entry in the diagonal of $\mathbf{D}_M$ respectively. We may drop the argument or subscript $\mathbf{M}$ if it is clear from context.

Spectral Graph Theory. Our results leverage well-known spectral graph theory results. In particular, our algorithm complexities are parameterized by the normalized pseudo condition number of a graph (or SDD matrix).

Definition 6 (Normalized (pseudo-)condition number)**.** *We define the* normalized (pseudo-)condition number *of an SDD matrix* $\mathbf{M} \in \mathbb{R}^n$ *as* $\bar{\kappa}(\mathbf{M}) := \lambda_n(\mathbf{N}_M)/\lambda_{\min}(\mathbf{N}_M)$.

To connect the normalized condition number to expander graphs, we can apply Cheeger's inequality, which guarantees that if G has α-expansion for some $\alpha = \widetilde{\Omega}(1)$, then $\bar{\kappa}(\mathbf{L}_G) = \lambda_n(\mathbf{N}_{L_G})/\lambda_2(\mathbf{N}_{L_G}) \leqslant 4/\alpha^2 = \widetilde{O}(1)$.

Theorem 6 (Cheeger's Inequality [38])**.** *Let* $G = (V, E, w)$ *be an undirected graph. Then,* $\frac{1}{2}\lambda_2(\mathbf{N}_L) \leqslant \phi(G) \leqslant \sqrt{2\lambda_2(\mathbf{N}_L)}$.

In addition, we leverage the fact that effective resistances can be expressed in several equivalent expressions. In particular, it is known that:

$$r_G(i,j) = \vec{\delta}_{i,j}^\top \mathbf{L}_G^\dagger \vec{\delta}_{i,j} = \vec{\delta}_{i,j} D^{-1/2} \mathbf{N}_G{}^\dagger D^{-1/2} \vec{\delta}_{i,j} = (\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j})^\top (\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j}),$$

where $\mathbf{W}_G \in \mathbb{R}^{E \times E}$ is the diagonal matrix of weights in G [5].

Runtimes and Space Complexities. In our algorithmic results and analysis, when clear from context, we use $\widetilde{O}(\cdot)$ (resp., $\widetilde{\Omega}(\cdot)$) notation to hide polylogarithmic factors (resp., inverse polylogarithmic factors) in the number of vertices, the number of edges, the size of the matrix, the number of nonzero entries in a matrix, the maximum and minimum diagonal element of a matrix, the maximum and minimum weighted degree of a graph, ϵ, the condition number, and the normalized psuedo-condition number of a matrix. We say event E occurs with high probability in t if $\mathbb{P}[E] \geqslant 1 - t^{-c}$, where $c > 0$ can be controlled by appropriately configuring the algorithm parameters. We may simply say that an event occurs "with high probability" or "whp." if it occurs with high probability in the size of a matrix or number of nodes in a graph.

3 Algorithmic Results

In this section, we present our main algorithmic results. Section 3.1 outlines our approach to effective resistance sketches and estimation; we defer discussion of our approach to SDD and PSD extensions to the supplementary material. Section 3.2, presents our original results on effective resistance sketching and estimation and generalizations to SDD matrices. Section 3.3 extends our techniques to yield an interesting data structure for estimating quadratic forms of PSD matrices.

3.1 Our Approach

Approach in prior work: Johnson Lindenstrauss sketches. Our starting inspiration is a classic result of [5], which obtains an $(\widetilde{O}(m\epsilon^{-2}), \widetilde{O}(\epsilon^{-2}), \widetilde{O}(n\epsilon^{-2}))$-effective resistance sketch by using the Johnson Lindenstrauss Lemma (JL) [39] and its algorithmic instantiations [40].

Lemma 1 (Johnson-Lindenstrauss Lemma [40]). *Given fixed vectors $v_1, ..., v_n \in \mathbb{R}^d$ and $\epsilon \in (0,1)$, let $\mathbf{J}$ be an independently sampled random matrix in $\{\pm 1/\sqrt{k}\}^{k \times d}$. For $k = \widetilde{O}(\log(n)\epsilon^{-2})$, whp. in n, $\|\mathbf{J}v_i\|_2 \approx_\epsilon \|v_i\|_2$ for all $i \in [n]$.*

[5] observe that $r_G(i,j) = (\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j})^\top (\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j})$. Consequently, w.h.p in n, $\|\mathbf{J}\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j}\|_2 \approx_\epsilon r_G(i,j)$. With SDD linear system solvers, $\mathbf{J}\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger$ can be approximated in $\widetilde{O}(m\epsilon^{-1})$ time, from which $\|\mathbf{J}\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j}\|_2$ can be queried in $\widetilde{O}(\epsilon^{-2})$ time.

Our approach: asymmetric CountSketch in ℓ_1. Towards improving upon JL sketches for effective resistance estimation, our key tool is to use other sketching algorithms. In particular we use that there are algoirthms that achieve better than $\widetilde{O}(\epsilon^{-2})$-sketch dimension for vectors with small ℓ_1 norm with comparable guarantees, e.g., CountSketch. CountSketch is a classic memory-efficient algorithm for estimating the number of occurences of various datapoints in a data stream [41] and efficiently computing inner products [42]. Given $v \in \mathbb{R}^n$ and integer parameters $s, t > 0$, CountSketch transforms v to a vector $\mathbf{S}v \in \mathbb{R}^{3ts \times n}$, where $\mathbf{S} \in \mathbb{R}^{3ts \times n}$ is a $3t$-column-sparse 0/1 matrix. Lemma 2 is a special case of Theorem 4 from [42], which provides accuracy guarantees for inner product estimation using CountSketch.

Lemma 2 (Special Case of [42], Theorem 4). *Let vectors $v, w \in \mathbb{R}^n$. Let $\mathbf{S}$ be a random CountSketch matrix. Let $x_i = \langle (\mathbf{S}v)[(i-1)s+1 : i \cdot s], (\mathbf{S}w)[(i-1)s+1 : i \cdot s] \rangle$ for $i \in [3t]$, and let X denote the median of $\{x_i\}$. For $s = O\left(\min\left(\frac{\|v\|_1\|w\|_1}{\epsilon|\langle v,w\rangle|}, \frac{\|v\|_2^2\|w\|_2^2}{\epsilon^2|\langle v,w\rangle|^2}\right)\right)$, and $t = \log(n^c)$, $|X - \langle v,w\rangle| \leqslant \epsilon\,|\langle v,w\rangle|$ with probability at least $\Omega(1-n^{-c})$.*

To improve the guarantee in Lemma 2 to hold whp. *for all* $v, w \in S$ rather than for each fixed pair, one can simply choose $t = \log(n^c|S|)$ and apply a union bound. Consequently, if we knew that $\|\mathbf{W}_G^{1/2} \mathbf{B}_G \mathbf{L}_G^\dagger \vec{\delta}_{i,j}\|_1^2 / r_G(i,j) = \widetilde{O}(1)$, then building a CountSketch with $s = \widetilde{O}(\epsilon^{-1})$ would yield a $\widetilde{O}(n\epsilon^{-1})$-size sketch, improving over the $\widetilde{O}(n\epsilon^{-2})$ sketch obtained using the ℓ_2 JL sketch in [5]. Unfortunately, it is unclear if and when such a bound holds, and so, it is unclear how the ℓ_1 CountSketches could be useful in this setting. This leads to the main insight that fuels our algorithms. Rather than seeking a symmetric factorization of $r_G(i,j)$ as a quadratic form $v^\top v$ and applying a sketching procedure to v, we instead work with an *asymmetric factorization*. In particular, we observe

$$r_G(i,j) = \langle \mathbf{D}_{\mathbf{L}_G}^{-1}\vec{\delta}_{i,j}, \mathbf{D}_{\mathbf{L}}^{1/2}(\mathbf{N}_{\mathbf{L}_G})^\dagger \mathbf{D}_{\mathbf{L}_G}^{-1/2}\vec{\delta}_{i,j}\rangle. \tag{1}$$

At first glance, it may seem unclear why (1) is helpful. However, we show that indeed, for expanders

$$\left\|\mathbf{D}_{\mathbf{L}}^{-1}\vec{\delta}_{i,j}\right\|_1 \left\|\mathbf{D}_{\mathbf{L}}^{1/2}(\mathbf{N}_{\mathbf{L}})^\dagger \mathbf{D}_{\mathbf{L}}^{-1/2}\vec{\delta}_{i,j}\right\|_1 / r_G(i,j) = \widetilde{O}(1). \tag{2}$$

Our main result essentially follows from (2). Using SDD linear system solvers, we can efficiently approximate $\mathbf{S}\mathbf{D}_{\mathbf{L}}^{1/2}(\mathbf{N}_{\mathbf{L}})^\dagger \mathbf{D}_{\mathbf{L}}^{-1/2} \in \mathbb{R}^{\widetilde{O}(\epsilon^{-1})\times n}$ in $\widetilde{O}(m\epsilon^{-1})$ time, yielding our $\mathcal{T}_{\mathrm{q}}$ of $\widetilde{O}(m\epsilon^{-1})$ and s of $\widetilde{O}(n\epsilon^{-1})$ (see discussion in supplementary material). Moreover, $\mathbf{S}\mathbf{D}_{\mathbf{L}}^{-1}$ is $\widetilde{O}(1)$-sparse, due to the structure of Count-Sketch. So, using our (approximate) access to $\mathbf{S}\mathbf{D}_{\mathbf{L}}^{1/2}(\mathbf{N}_{\mathbf{L}})^\dagger \mathbf{D}_{\mathbf{L}}^{-1/2}$, for any query $i,j \in V$, we can efficiently approximate (1) using the recovery procedure of Lemma 2 in $\widetilde{O}(1)$ time.

3.2 Our Results

We use the approach in Section 3.1 to develop algorithms to compute spectral sketch data structures for SDD matrices $\mathbf{M}$ with $\widetilde{O}(1)$ normalized condition number, as defined in Definitions 6 and 7. So, as discussed in Section 2, this implies that our spectral sketch algorithms will automatically apply to normalized Laplacians of expanders and enable us to compute effective resistances.

Definition 7 (Spectral Sketch Data Structure)**.** *We say an algorithm produces an $(\mathcal{T}_{\mathrm{s}}, \mathcal{T}_{\mathrm{q}}, s)$-spectral sketch data structure for a PSD matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ if given $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\epsilon \in (0,1)$, the algorithm creates a binary string of length $O(s(A,\epsilon))$ in time $O(\mathcal{T}_{\mathrm{s}}(A,\epsilon))$, from which, for any supported query $b \in \mathbb{R}^n$, w.h.p it outputs $q_A(b) \approx_\epsilon b^\top \mathbf{A} b$ in time $O(\mathcal{T}_{\mathrm{q}}(A,\epsilon)(\mathrm{nnz}(b))^2)$.*

Our spectral sketches of SDD matrices $\mathbf{M}$ will only support $\mathbf{D}_M$-numerically sparse query vectors.

Definition 8 ($\mathbf{D}$-numerically sparse)**.** *For a diagonal matrix $\mathbf{D}$, the $\mathbf{D}$-numerical sparsity of $x \in \mathbb{R}^n$ is $\mathrm{ns}_{\mathbf{D}}(x) := \left\|\mathbf{D}^{-1}x\right\|_1 \|x\|_1 / \left\|\mathbf{D}^{-1/2}x\right\|_2^2$. We say x is $(c,\mathbf{D})$-numerically sparse if $\mathrm{ns}_{\mathbf{D}}(x) \leqslant c$.*

Definition 8 is restrictive; however, several natural classes of vectors satisfy the requirements. For example, $\mathbb{1}_i$ are $(1,\mathbf{D})$-numerically sparse and $\vec{\delta}_{i,j}$ is $(2,\mathbf{D}_M)$-numerically sparse for any $i,j \in [n]$ and invertible diagonal matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ (see supplementary material for additional examples.)

The following asymmetric rearrangement of quadratic forms is crucial to our analysis.

Lemma 3. *Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be SDD and $x \in \mathbb{R}^n$ be orthogonal to* $\ker(\mathbf{M})$*. Then,*

$$x^\top \mathbf{M}^\dagger x = \frac{1}{2}\left\langle \mathbf{D}_M^{-1}x, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2}x\right\rangle = \left\langle \mathbf{D}_M^{-1/2}x, \mathbf{N}_M^\dagger \mathbf{D}_M^{-1/2}x\right\rangle \geqslant \frac{1}{2}\left\|\mathbf{D}_M^{-1/2}x\right\|_2^2.$$

Proof. For notational convenience, let $\mathbf{N}_{\ell M} = \mathbf{N}_M/2$. Let $v = \mathbf{M}^\dagger x = (\mathbf{D}_M - \mathbf{A}_M)^\dagger x$. Note that $\mathbf{D}_M^{-1/2}x \perp \ker(\mathbf{N}_M)$, and $2\mathbf{D}_M^{1/2}\mathbf{N}_{\ell M}\mathbf{D}_M^{1/2} = \mathbf{M}$. Consequently, $v = \frac{1}{2}\mathbf{D}_M^{-1/2}\mathbf{N}_{\ell M}^\dagger \mathbf{D}_M^{-1/2}x$, and hence $x^\top \mathbf{M}^\dagger x = \frac{1}{2}\langle \mathbf{D}_M{}^{-1}x, \mathbf{D}_M{}^{1/2}\mathbf{N}_{\ell M}^\dagger \mathbf{D}_M{}^{-1/2}x\rangle$. The second equality now follows immediately by rearranging terms. To obtain the inequality, note that, because $\mathbf{M}$ is SDD and $\mathbf{D}$ is invertible, $\mathbf{N}_M$ is PSD. Furthermore, $2\mathbf{I} - \mathbf{N}_M = \mathbf{I} - \mathbf{D}_M^{-1/2}\mathbf{A}_M\mathbf{D}_M^{-1/2}$, which is also PSD, as $\lambda(\mathbf{A}_M) \subset [-d_{\max}, d_{\max}]$. So, $0 \leqslant \lambda(\mathbf{N}_M) \leqslant 2$. So, $\lambda_{\min}(\mathbf{N}_M) \geqslant 1/2$ and the lemma follows. ☐

Lemma 4 bounds $\left\|\mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2}x\right\|_1$. The proof uses the power series expansion of $(\mathbf{N}_M/2)^\dagger$.

Lemma 4. *Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be an SDD matrix and $x \in \mathbb{R}^n$ be a unit vector orthogonal to* $\ker(\mathbf{M})$*. Then*

$$\left\|\mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M{}^{-1/2}x\right\|_1 \leqslant \max\left(1, 2\bar{\kappa}(\mathbf{M})\log\left(\sqrt{nd_{\max}}2\bar{\kappa}(\mathbf{M})/\sqrt{d_{\min}}\right)\right)\|x\|_1 + 1.$$

Combining our Lemmas 3 and 4 with the guarantees of Lemma 2 and prior work on SDD linear system solvers (see supplementary material), we obtain the following theorem.

Theorem 7. *For any SDD matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$, there is an algorithm to construct an $(\widetilde{O}(\overline{\kappa}(\mathbf{M})\,\mathrm{nnz}(\mathbf{M})\epsilon^{-1}), \widetilde{O}(1), \widetilde{O}(\overline{\kappa}(\mathbf{M})\,n\epsilon^{-1}))$-spectral sketch data structure of $\mathbf{M}^{\dagger}$ supported over queries S, where S is any set of $(\widetilde{O}(1), \mathbf{D}_M)$-numerically sparse vectors orthogonal to* $\ker(\mathbf{M})$.

Because the $\vec{\delta}_{i,j}$ queries appearing in effective resistance computations are 2-sparse and $(2, \mathbf{D}_M)$-numerically sparse for all SDD matrices $\mathbf{M}$, taking $\mathbf{M} = \mathbf{L}_G$ in Theorem 7 immediately implies Theorem 2 and Theorem 3 (see supplementary material for detailed discussion and pseudocode.)

3.3 Extensions to PSD Matrices

Our approach of approximating quadratic forms via asymmetric inner products also yields a query-efficient sketching procedure for approximating quadratic forms of well-conditioned PSD matrices.

Theorem 8. *There is an algorithm to construct an $(\widetilde{O}(\kappa(\mathbf{A})\mathrm{nnz}(\mathbf{A})\epsilon^{-2}), \widetilde{O}(1), \widetilde{O}(\kappa(\mathbf{A})n\epsilon^{-2}))$-spectral sketch data structure of $\mathbf{A}$ supported over S, where S is any set of vectors orthogonal to* $\ker(\mathbf{A})$ *and $\mathbf{A}$ is PSD.*

In comparison, JL [39] gives an $(\widetilde{O}(n^{\omega}), \widetilde{O}(\epsilon^{-2}), \widetilde{O}(n\epsilon^{-2}))$-spectral sketch data structure using efficient square root algorithms [43]. JL achieves better compression than Theorem 8, while Theorem 8 achieves faster query time. When the matrix is well conditioned and error tolerance is sufficiently high, Theorem 8 may achieve better construction time and query time than JL while maintaining comparable compression.

4 Lower Bounds

In this section, we present our main hardness results. Section 4.1 outlines our approach. In Section 4.2, we present our lower bounds for the problem of estimating effective resistances for all pairs of nodes (case where $S = V \times V$), which we call the "all pairs effective resistance estimation problem." Section 4.3 shows our techniques also yield lower bounds for other spectral sum estimation problems.

4.1 Our Approach

Approaches of Previous Work The approach of [33] begins with the fact that G has a triangle if and only if $\mathrm{tr}\left(\mathbf{A}_G^3\right)/6 \geqslant 1$. They use the fact that various spectral sums $\mathcal{S}_f$ of the of the SDD matrix $\mathbf{I} - \delta\mathbf{A}_G$ (for δ sufficiently small) can be expressed as a power series $\mathcal{S}_f(\mathbf{I} - \delta\mathbf{A}_G) = \sum_{k=0}^{\infty} c_k \delta^k\, \mathrm{tr}(\mathbf{A}_G^k)$. The first two terms of this series can be computed directly. So given $Y \approx_{\epsilon} \mathcal{S}_f(\mathbf{I} - \delta\mathbf{A}_G)$, one can estimate $\mathrm{tr}\left(\mathbf{A}_G^3\right)$, where the estimation error is controlled by the magnitude of the first two terms of the series and the tail error due to truncating at the third term. By bounding this estimation error, [33] show that, for appropriate choices of δ, $Y \approx_{\epsilon} \mathcal{S}_f(\mathbf{I} - \delta\mathbf{A}_G)$ yields an additive 1/2 approximation to $\mathrm{tr}\left(\mathbf{A}_G^3\right)$, which is sufficient for triangle detection. They also reduce the problem of estimating the spectral sum $\mathrm{tr}\left(\mathbf{B}^{-1}\right)$ for an SDD matrix $\mathbf{B}$ to the all pairs effective resistance estimation problem.

Our Approach We use three key techniques to better bound the estimation errors incurred in the power-series-inspired approach of [33]. This yields faster reductions and better lower bounds for effective resistance estimation. Rather than obtaining effective resistance lower bounds by reducing the problem of computing $\mathrm{tr}\left(\mathbf{A}_G^3\right)/6$ to computing the *trace* of an SDD matrix as in [33], we use a reduction that closely resembles the structure of effective resistances. For $\alpha > 0$ sufficiently small,

$$\left(\mathbf{I} - \frac{\alpha}{n}\mathbf{A}_G\right)^{-1} = \sum_{k=0}^{\infty} \frac{\alpha^k}{n^k}\mathbf{A}_G^k. \tag{3}$$

Since $\mathbf{A}_G$ is known, given access to $\vec{\delta}_{i,j}^{\top}(\mathbf{I} - \frac{\alpha}{n}\mathbf{A}_G)^{-1}\vec{\delta}_{i,j}$, we can estimate the entries of $\mathbf{A}_G^2$, where the estimation error is controlled by α and the tail error of truncating (3) at the third term. By bounding this estimation error, for appropriate choice of α, we can obtain additive 1/2 approximations to all entries of $\mathbf{A}_G^2$, which is sufficient to identify all paths of length 2. We can then detect a triangle by simply scanning for an edge $\{u, v\}$ such that u and v are connected by a path of length 2. Estimating the entries of $\mathbf{A}_G^2$ leads to lower estimation error than estimating $\operatorname{tr}\left(\mathbf{A}_G^3\right)$ as in [33]). Second, we use a standard randomized reduction that reduces the triangle detection problem to the triangle detection problem restricted to tripartite graphs. The reduction relies on the fact that a randomly sampled tripartition of the original graph preserves triangles with constant probability. To detect a triangle in a tripartite graph $G = (V_1 \sqcup V_2 \sqcup V_3, E)$, we construct a graph H by removing all edges $E_{1,2} := \{\{u, v\} \in E : u \in V_1, v \in V_2\}$ between V_1 and V_2. G has a triangle if and only if there is an edge $\{u, v\} \in E_{1,2}$ and a path of length 2 between u and v in H. Crucially, we can show that the third term $\bar{\mathbf{A}}_H^3$ does not contribute to the tail error when estimating the $\{u, v\}$-th entry of $\left(\mathbf{A}_H^2\right)$ using (3). Third, to lower the spectral norm of $\mathbf{A}_H$ (and consequently better bound the convergence of the power series (3)), we introduce the symmetric random signing of $\mathbf{A}_H$ below.

Definition 9 (Symmetric Random Signing)**.** *Given a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, its* symmetric random signing *$\bar{\mathbf{A}}$ is the random matrix with $\bar{\mathbf{A}}_{i,j} := \xi_{i,j}\mathbf{A}_{i,j}$, where $\xi_{i,j}$ are independent Rademacher random variables that satisfy $\xi_{i,j} = \xi_{j,i}$.*

We show that this random signing preserves the non-zeroness of entries of $\mathbf{A}_H^2$ with constant probability, allowing us to detect whether G has a triangle even if we replace $\mathbf{A}_G$ in (3) with $\bar{\mathbf{A}}_H$ instead. This is beneficial, as matrix Chernoff guarantees $\|\bar{\mathbf{A}}_H\|_2 = \widetilde{O}(\sqrt{n})$ whp. whereas $\|\mathbf{A}_H\|_2$ may be as large as n. So the tail error of truncating the power series is smaller. To compute entries of $\bar{\mathbf{A}}_H^2$ efficiently using effective resistance estimates on expanders, we first show that we can use all pairs effective resistance estimates on expanders to estimate $\vec{\delta}_{i,j}^T\mathbf{M}^{-1}\vec{\delta}_{i,j}$ for all $i, j \in [n]$, where $\mathbf{M} = (\mathbf{I} - \mathbf{Q})$ is an SDD matrix with $\rho(\mathbf{Q}) \leqslant 1/3$. Then, by choosing $\mathbf{M} = \mathbf{I} - \frac{\alpha}{n}\bar{\mathbf{A}}_H$ as in (3) for an appropriate constant α, we can estimate $\bar{\mathbf{A}}_H^2$ from estimates of $\vec{\delta}_{i,j}^T\mathbf{M}^{-1}\vec{\delta}_{i,j}$. This yields our lower bound on the all pairs effective resistance estimation problem.

Additionally, we show that random signing also preserves the non-zeroness of $\operatorname{tr}\left(\mathbf{A}_G^3\right)$ with constant probability, and leverage this to obtain improved randomized conditional lower bounds for various spectral sum estimation problems. We closely follow the trace estimation approach of [33], and again use the smaller spectral radius of $\bar{\mathbf{A}}_G$ to improve bounds on the power series truncation error.

4.2 Improved Lower Bounds for Effective Resistance Estimation

In this section we provide a series of reductions which yield our main result on lower bounds for the all pairs effective resistance estimation problem for all pairs of nodes (case where $S = V \times V$).

Definition 10. *In the* SDD effective resistance estimation problem*, given an SDD matrix $\mathbf{M}$ such that $\mathbf{D}_\mathbf{M} = \mathbf{I}$, $\mathbf{A}_\mathbf{M} = \mathbf{Q}$, with $\rho(\mathbf{Q}) \leqslant 1/3$ and $\epsilon \in (0, 1)$, we must output $\widetilde{r} \in \mathbb{R}^{n^2}$ such that $\widetilde{r}_{a,b} \approx_\epsilon \vec{\delta}_{a,b}^\top\mathbf{M}^{-1}\vec{\delta}_{a,b}$ $\forall a, b \in [n]$. We call $\vec{\delta}_{a,b}^\top\mathbf{M}^{-1}\vec{\delta}_{a,b}$ the SDD effective resistance of (a, b) in $\mathbf{M}$.*

For brevity, we use $\widetilde{r}(\mathbf{M})$ to refer to the solution of the SDD effective resistance problem on input $\mathbf{M}$. Our first step is to show that an algorithm for the all pairs effective resistance estimation problem on expanders implies an algorithm for the SDD effective resistance estimation problem.

Lemma 5. *Given an algorithm to solve the all pairs effective resistance estimation problem on graphs with $\widetilde{\Omega}(1)$-expansion in $\widetilde{O}(n^2\epsilon^{-c})$ time for some $c > 0$, we can produce an algorithm to solve the SDD effective resistance estimation problem in $\widetilde{O}(n^2\epsilon^{-c})$ time.*

To prove Lemma 5, we first prove the lemma for the case where $\mathbf{Q}$ is entrywise non-negative by constructing an expander G with $n + 1$ vertices such that $r_G(a, b) = \vec{\delta}_{a,b}^\top\mathbf{M}^{-1}\vec{\delta}_{a,b}$ for all $a, b \in [n]$. We extend the reduction to arbitrary $\mathbf{Q}$ by constructing $\mathbf{Q}'$ of size $2n$ so that $\mathbf{Q}'$ is entrywise non-negative and $\widetilde{r}(\mathbf{I} - \mathbf{Q})$ is a simple linear transform of $\widetilde{r}(\mathbf{I} - \mathbf{Q}')$.

We turn our attention to reducing the triangle detection problem to the SDD effective resistance problem. As discussed, a key aspect of our approach is to work with the random signing $\bar{\mathbf{A}}_G$ of $\mathbf{A}_G$. Lemma 6 shows that to determine whether $(\mathbf{A}_G^2)_{i,j} > 0$ with constant probability, it suffices to determine whether $(\bar{\mathbf{A}}_G^2)_{i,j} > 0$. Matrix Chernoff ensures whp. $\rho(\bar{\mathbf{A}}_G) = \widetilde{O}(\sqrt{n})$, while $\rho(\mathbf{A}_G)$ could be as large as n [44]. So, estimating entries of $\bar{\mathbf{A}}_G$ leads to lower power series tail error.

Lemma 6. *For $i \neq j$, if $(\mathbf{A}_G^2)_{i,j} = 0$, then $(\bar{\mathbf{A}}_G^2)_{i,j} = 0$; if $(\mathbf{A}_G^2)_{i,j} > 0$, $\mathbb{P}\left[|(\bar{\mathbf{A}}_G^2)_{i,j}| > 1\right] \geqslant 1/2$.*

The idea of the proof is that if $\{a,b\},\{b,c\}$ exist in G, either $\xi_{a,c}=1$ or $\xi_{a,c}=-1$ results in $(\bar{\mathbf{A}}_G^2)_{a,c}>0$. Finally, we use the power series approach in Section 4.1 to obtain Theorem 9.

Theorem 9. *Given an algorithm which solves the SDD effective resistance estimation problem in $\widetilde{O}(n^2\epsilon^{-c})$ time, we can produce a randomized algorithm that solves the triangle detection problem in $\widetilde{O}(n^{2(1+c)})$ time whp.*

Theorem 9 and Lemma 5 with $c=1/2-\delta$ immediately imply our main result Theorem 4.

Additionally, we extend the lower bound of Theorem 4 to the all edges effective resistance estimation by the following reduction.

Lemma 7. *Given an algorithm to solve the all edges effective resistance estimation problem (i.e., Definition 1 where $S=E$) in $\widetilde{O}(m\epsilon^{-c})$ time, we can produce an algorithm to solve the all pairs effective resistance estimation problem in $\widetilde{O}(n^2\epsilon^{-c})$ time for some $c>0$.*

The rough idea behind the reduction is to add a complete graph of edges of sufficiently small weight that would not change the effective resistances much. Lemma 7 combined with Theorem 4 then imply a $\widetilde{\Omega}(m\epsilon^{-1/2})$ randomized lower bound for the all edges effective resistance estimation problem on graphs with graphs with $\widetilde{\Omega}(1)$-expansion.

4.3 Improved Lower Bounds for Spectral Sum Estimation

Finally, we discuss our improved lower bounds for various spectral sum estimation problems. Analogous to Lemma 6, in the following lemma we show that to determine whether a graph has a triangle (i.e., $\mathrm{tr}\left(\mathbf{A}_G^3\right)>0$) with constant probability, it suffices to determine whether $\mathrm{tr}\left(\bar{\mathbf{A}}_G^3\right)>0$.

Lemma 8. *If $\mathrm{tr}\left(\mathbf{A}_G^3\right)=0$, then $\mathrm{tr}\left(\bar{\mathbf{A}}_G^3\right)=0$, and if $\mathrm{tr}\left(\mathbf{A}_G^3\right)>0$ then $\mathbb{P}\left[\left|\mathrm{tr}\left(\bar{\mathbf{A}}_G^3\right)\right|>0\right]\geqslant 1/4$.*

The central idea of the proof is that if a triangle $\{a,b\},\{b,c\},\{c,a\}$ exists in G, then amongst the 4 possible configurations of the Rademacher random signing variables $\xi_{a,b}$ and $\xi_{b,c}$, at least one configuration must result in $\left|\mathrm{tr}\left(\bar{\mathbf{A}}_G^3\right)\right|>0$. By following the proof of Theorem 15 from [33], and replacing their use of $\mathbf{A}_G$ with a symmetric random signing $\bar{\mathbf{A}}_G$, we obtain an improved randomized version of their result by leveraging the smaller spectral radius of $\bar{\mathbf{A}}_G$. Theorem 5 follows by applying this result to the functions f that define the corresponding spectral sums (see supplementary material).

5 Conclusion

In this paper we provided improved upper and lower bounds on the problem of estimating and sketching effective resistances on expanders. On the algorithmic side we show how sketches tailored to ℓ_1 when carefully applied to asymmetric formulations of the quadratic form of the Laplacian pseudoinverse gave our results. On the lower bound side, we provided an alternative to the trace estimation approach of [33] for showing lower bounds and coupled it with techniques of randomly signing edges of the graph to obtain our results. Further, we showed that these techniques had broader implications for addressing algorithmic challenges in numerical linear algebra.

Beyond the natural open problem of improving both our upper and lower bounds towards bringing them together, there are interesting open problems in broadening the applicability of both our upper and lower bounds. For example, obtaining an $\widetilde{O}(m\epsilon^{-1})$ time algorithm for estimating the effective resistance of all edges in a general (non-expander) graph and extending our $\widetilde{\Omega}(n^2\epsilon^{-1/2})$ lower bounds to deterministic algorithms remain interesting open problems. We hope that the results of this paper provide useful tools for addressing each.

Acknowledgments and Disclosure of Funding

We thank Hongyue Li for helpful discussions and work on this project at various stages. Aaron Sidford was supported in part by a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, a PayPal research award, and a Sloan Research Fellowship. Ishani Karmarkar was supported in part by an NSF CAREER Award CCF-1844855, NSF Grant

CCF-1955039, a PayPal research grant, and a Stanford Institute for Computational and Mathematical Engineering (ICME) Fellowship. Rajat Vadiraj Dwaraknath was supported by a Stanford ICME Fellowship.

References

[1] Aleksander Madry, Damian Straszak, and Jakub Tarnawski. Fast generation of random spanning trees and the effective resistance metric. In *Symposium on Discrete Algorithms*, 2014.

[2] Karel Devriendt. Effective resistance is more than distance: Laplacians, simplices and the schur complement. In *Linear Algebra and its Applications*, 2022.

[3] Huan Li and Zhongzhi Zhang. Kirchhoff index as a measure of edge centrality in weighted networks: Nearly linear time algorithms. In *Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2018.

[4] Douglas Klein and Milan Randic. Resistance distance. In *Journal of Mathematical Chemistry*, 1993.

[5] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *SIAM Journal of Computing*, 2009.

[6] Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. In *SIAM Journal of Computing*, 2015.

[7] David Durfee, Yu Gao, Gramoz Goranci, and Richard Peng. Fully dynamic spectral vertex sparsifiers and applications. In *Symposium on Theory of Computing*, 2019.

[8] Rakshith S Srinivasa, Cao Xiao, Lucas Glass, and Jimeng Sun Justin Romberg. Fast graph attention networks using effective resistance based graph sparsification. In *arXiv preprint arXiv:2006.08796*, 2020.

[9] Tasweer Ahmad, Lianwen Jin, Luojun Lin, and GuoZhi Tang. Skeleton-based action recognition using sparse spatio-temporal gcn with edge effective resistance. In *Neurocomputing*, 2021.

[10] Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Efficient sampling for gaussian graphical models via spectral sparsification. In *Proceedings of the Conference on Learning Theory*, 2015.

[11] Daniele Calandriello, Alessandro Lazaric, Ioannis Koutis, and Michal Valko. Improved large-scale graph learning through ridge spectral sparsification. In *International Conference on Machine Learning*, 2018.

[12] Alexander Mercier, Samuel Scarpino, and Cristopher Moore. Effective resistance against pandemics: Mobility network sparsification for high-fidelity epidemic simulations. In *PLOS Computational Biology*, 2022.

[13] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Symposium on Theory of Computing*, 2010.

[14] Aleksander Madry. Navigating central path with electrical flows: from flows to matchings, and back. In *Symposium on Foundations of Computer Science*, 2013.

[15] Aleksander Madry. Computing maximum flow with augmenting electrical flows. In *Computing Maximum Flow with Augmenting Electrical Flows*, 2016.

[16] Jan van den Brand, Yu Gao, Arun Jambulapati, Yin Tat Lee, Yang P Liu, Richard Peng, and Aaron Sidford. Faster maxflow via improved dynamic spectral vertex sparsifiers. In *Symposium on Theory of Computing*, 2022.

[17] Jan Van Den Brand, Yang P Liu Yin Tat Lee, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *Symposium on Theory of Computing*, 2021.

[18] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *Symposium on Foundations of Computer Science*, 2014.

[19] David Durfee, Rasmus Kyng, John Peebles, Anup B. Rao, and Sushant Sachdeva. Sampling random spanning trees faster than matrix multiplication. In *Symposium on Theory of Computing*, 2017.

[20] Aaron Schild. An almost-linear time algorithm for uniform random spanning tree generation. In *Symposium on Theory of Computing*, 2018.

[21] Aleksander Madry, Damian Straszak, and Jakub Tarnawski. Fast generation of random spanning trees and the effective resistance metric. In *Symposium on Discrete Algorithms*, 2014.

[22] Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph clustering using effective resistance. In *arXiv preprint arXiv:1711.06530*, 2017.

[23] Zhiqiang Zhao and Zhuo Feng. Effective-resistance preserving spectral reduction of graphs. In *Proceedings of the Annual Design Automation Conference*, 2019.

[24] Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio', and Michael Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International Conference on Machine Learning*, 2023.

[25] Pradeep Banerjee, Kedar Karhadkar, and Guido Montúfar Yu Guang Wang, Uri Alon. Oversquashing in gnns through the lens of information contraction and graph expansion. In *Allerton Conference on Communication, Control, and Computing*, 2022.

[26] Mitchell Black, Amir Nayyeri, Zhengchao Wan, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning*, 2023.

[27] Ameya Velingker, Ali Kemal Sinop, Ira Ktena, Petar Velivcković, and Sreenivas Gollapudi. Affinity-aware graph networks. In *arXiv preprint arXiv:2206.11941*, 2022.

[28] Arun Jambulapati and Aaron Sidford. Efficient $\widetilde{O}(n/\epsilon)$ spectral sketches for the laplacian and its pseudoinverse. In *Symposium on Discrete Algorithms*, 2018.

[29] Lawrence Li and Sushant Sachdeva. A new approach to estimating effective resistances and counting spanning trees in expander graphs. In *Symposium on Discrete Algorithms*, 2022.

[30] Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junxing Wang. Graph sparsification, spectral sketches, and faster resistance computation, via short cycle decompositions. In *Symposium on Foundations of Computer Science*, 2018.

[31] Gramoz Goranci, Monika Henzinger, and Pan Peng. Dynamic effective resistances and approximate schur complement on separable graphs. In *Embedded Systems and Applications*, 2018.

[32] Michael Dinitz, Robert Krauthgamer, and Tal Wagner. Towards resistance sparsifiers, 2015.

[33] Cameron Musco, Praneeth Netrapalli, Aaron Sidford, Shashanka Ubara, and David P Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. In *arXiv preprint arXiv:1704.04163*, 2017.

[34] Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. In *Symposium on Foundations of Computer Science*, 2018.

[35] Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *https://arxiv.org/abs/2210.10173*, 2023.

[36] Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. On sketching quadratic forms. In *Proceedings of the Conference on Innovations in Theoretical Computer Science*, 2015.

[37] Huan Li and Aaron Schild. Spectral subspace sparsification. In *Symposium on Foundations of Computer Science*, 2018.

[38] Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Problems in Analysis*, 1971.

[39] William B. Johnson. Extensions of lipschitz mappings into hilbert space. In *Contemporary mathematics*, 1984.

[40] Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the Symposium on Principles of Database Systems*, 2001.

[41] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Theoretical Computer Science*, 2004.

[42] Kasper Green Larsen, Rasmus Pagh, and Jakub Tetek. Countsketches, feature hashing and the median of three. In *International Conference on Machine Learning*, 2021.

[43] Prateek Jain, Chi Jin, Sham M. Kakade, and Praneeth Netrapalli. Global convergence of non-convex gradient descent for computing matrix squareroot. In *International Conference on Artificial Intelligence and Statistics*, 2017.

[44] Joel A Tropp. User-friendly tail bounds for sums of random matrices. In *Foundations of computational mathematics*, 2012.

[45] Yu Gao, Yang Liu, and Richard Peng. Fully dynamic electrical flows: Sparse maxflow faster than goldberg–rao. In *SIAM Journal on Computing*, 2023.

[46] Jan van den Brand, Yu Gao, Arun Jambulapati, Yin Tat Lee, Yang P. Liu, Richard Peng, and Aaron Sidford. Faster maxflow via improved dynamic spectral vertex sparsifiers. In *Symposium on Theory of Computing,*, 2022.

[47] Ignacio Garcia-Marco, Pascal Koiran, Timothée Pecatte, and Stéphan Thomassé. On the complexity of partial derivatives. In *Theoretical Computer Science*, 2017.

[48] Jacques Morgenstern. Note on a lower bound on the linear complexity of the fast fourier transform. In *Journal of the Association for Computing Machinery*, 1973.

[49] Amir Shpilka. Lower bounds for matrix product. In *SIAM Journal on Computing*, 2003.

[50] Virginia Vassilevska Williams, Eyob Woldeghebriel, and Yinzhan Xu. Algorithms and lower bounds for replacement paths under multiple edge failure. In *Symposium on Foundations of Computer Science*, 2022.

[51] Mahdi Boroujeni, Sina Dehghani, Soheil Ehsani, MohammadTaghi HajiAghayi, and Saeed Seddighin. Subcubic equivalences between graph centrality measures and complementary problems. In *arXiv preprint arXiv:1905.08127*, 2019.

[52] Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians: Fast, sparse, and simple. In *Symposium on Foundations of Computer Science*, 2016.

[53] Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving sdd systems in nearly-linear time. In *Symposium on Theory of Computing*, 2014.

[54] Daniel A Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. In *SIAM Journal on Computing*, 2013.

[55] Arun Jambulapati and Aaron Sidford. Ultrasparse ultrasparsifiers and faster laplacian system solvers. In *Symposium on Discrete Algorithms*, 2021.

[56] Neha Gupta and Aaron Sidford. Exploiting numerical sparsity for efficient learning: Faster eigenvector computation and regression. In *Neural Information Processing Systems*, 2018.

[57] Suk-Geun Hwang. Cauchy's interlace theorem for eigenvalues of hermitian matrices. In *The American mathematical monthly*, 2004.

6 Supplementary Material

In this supplementary material section, we include additional discussion and proofs of the results from the main body. In Section 6.1, we discuss additional related work on effective resistance estimation and fine-grained complexity analysis more broadly. In Section 6.2, we discuss additional details and provide omitted proofs pertaining to our algorithmic results in Section 3. In Section 6.3, we discuss additional details and provide omitted proofs pertaining to our hardness results in Section 4.

6.1 Additional Related Work

Here we supplement the discussion in Section 1 by briefly discussing additional work related to the effective resistance estimation problem and providing a more detailed comparison of our results to [29].

Dynamic effective resistance estimation. Effective resistance estimation and sketching are part of a broader family of previously studied problems involving graph compression and effective resistance estimation. For example, there is a related line of work on dynamically maintaining effective ressistance estimates in dynamic graphs, e.g., [19, 31], which in turn is related to problems of dynamically maintaining electric flows in graphs, e.g., [45, 46]. Whether our techniques have ramifications for these related problems is an interesting question for future work.

Fine-grained complexity analysis. Our effective resistance estimation lower bounds fall under a broader topic of fine grained complexity analysis, i.e., the problem of characterizing the optimal complexity of problems which are known to have polynomial time solutions. Here, we provide references to a few examples. [34] showed subcubic equivalences between the problem of triangle detection, Boolean matrix multiplication, and several other graphical problems. As discussed, [33] utilize the results of [34] to obtain conditional lower bounds on several spectral sum approximation problems– many of which we also study in this paper. Similarly, [47–49] provided several conditional complexity lower bounds for linear algebraic problems, conditional on the use of particular computational models. [50] and [51] also provide fine grained lower bounds for the fault replacement paths problem, problems on graph centrality measures, and complementary problems. Making connections between our techniques for our effective resistance estimation lower bounds and these prior works in fine-grained complexity analysis is an interesting open problem.

Comparison with the approach of [29]. Effective reisistance sketching and estimation for expanders was previously studied in [29]. [29] produces an $(\widetilde{O}(m + n\epsilon^{-2}), \widetilde{O}(1), \widetilde{O}(n\epsilon^{-1}))$ effective resistance sketch for graphs with $\widetilde{\Omega}(1)$-expansion. Our work provides a different, independently obtained runtime for effective resistance estimation by producing an $(\widetilde{O}(m\epsilon^{-1}), \widetilde{O}(1), \widetilde{O}(n\epsilon^{-1}))$ effective resistance sketch for graphs with $\widetilde{\Omega}(1)$ expansion. Additionally, our work can be applied to produce an $(\widetilde{O}(m^{1+o(1)} + n\epsilon^{-2}), \widetilde{O}(1), \widetilde{O}(n\epsilon^{-1}))$ effective resistance sketch, by running on a sparse graphical sketch, such as that guaranteed by [30] (see Section 1.1). Consequently, our results match those of [29] up to an $m^{o(1)}$ factor in all regimes, and improve for sufficiently high accuracy on sparse graphs.

Given a graph G with $\widetilde{\Omega}(1)$-expansion, [29] proposes an algorithm which is motivated by the idea of storing $\widetilde{O}(1)$-sparse approximations to the columns of $\mathbf{L}_G^\dagger$, which would clearly be sufficient for querying effective resistances of G in $\widetilde{O}(1)$ time. However, it is unclear whether the columns of $\mathbf{L}_G^\dagger$ have small ℓ_1 norm, and consequently, it is unclear how to obtain these sparse approximations. Consequently, their algorithm instead estimates the following vector σ_u for each $u \in V$,

$$\sigma_u = \frac{1}{2}\sum_{t=0}^{\infty}\left(\left(\frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{A}_G\mathbf{D}_G^{-1}\right)^t e_u - \pi\right),$$

where $\pi = \frac{\mathbf{D}_{\mathbf{L}_G}\mathbb{1}}{\mathbb{1}^\top \mathbf{D}_{\mathbf{L}_G}\mathbb{1}}$. They show that σ_u is closely related to $\mathbf{D}_G\mathbf{L}_G^\dagger e_u - \pi$, and consequently access to σ_u is sufficient for estimating effective resistances. Additionally, they use structural properties of expanders to show that each σ_u must have small ℓ_1 norm and that it can be computed efficiently by running lazy random walks on G (i.e., the random walk which, at each step follows the natural random walk on G with probability 1/2 and stays idle with probability 1/2). These key properties of σ_u enable their result.

Our approach is similar to that of [29] in that we also reformulate the effective resistance between two vertices as an inner product between two vectors whose ℓ_1 norm we can bound; however, the vectors we consider are not the σ_u vectors considered in [29]. Instead, we rewrite $r_G(i,j)$ as the inner product of $\mathbf{D}_{\mathbf{L}_G}^{-1}\vec{\delta}_{i,j}$ with $\mathbf{D}_{\mathbf{L}_G}^{1/2}(\mathbf{N}_{\mathbf{L}_G}/2)^{\dagger}\mathbf{D}_{\mathbf{L}_G}^{-1/2}\vec{\delta}_{i,j}$. Similar to [29], we then use similar underlying structural properties of expanders to argue that the ℓ_1 norms of these vectors is not too large. However, instead of using random walks to estimate these vectors, we use sketching techniques (specifically, CountSketch) and Laplacian system solvers to estimate them, an idea which is inspired by the work of [5]. The differences in the specific effective resistance vectors we estimate and the different technique of estimating them is what leads to the difference in runtime between [29] and our own work. [3] [29] also provide an extension of their effective resistance sketch techniques to well-conditioned SDD matrices, which we also do in our generalized Theorem 7; our lower bounds on the SDD effective resistance estimation problem (see Section 4.2) therefore also apply to the work of [29].

6.2 Additional Discussion of Algorithmic Results

In Section 6.2.1, we discuss related background on CountSketch and SDD linear system solvers, which are crucial to our proofs of our upper bound results. In Section 6.2.2 we expand on our discussion of $\mathbf{D}_{\mathbf{M}}$-numerical sparsity from Definition 8 and provide our motivation for studying such queries. In Section 6.2.3 we present omitted proofs from Section 3.2 and Section 3.3.

6.2.1 Relevant Technical Background

CountSketch. Below we prove that Lemma 2 follows from Theorem 4 in [42].

Proof of Lemma 2. Applying Theorem 4 from [42] with $t = 2$,

$$\mathbb{V}\left[|X - \langle v, w\rangle|\right] \leqslant \min\left(3\frac{\|v\|_1^2\|w\|_1^2}{s^2}, 2\frac{\|v\|_2^2\|w\|_2^2}{s}\right).$$

Applying Markov's inequality for the second moment,

$$\mathbb{P}\left[|X - \langle v, w\rangle| > 2\sqrt{3}\min\left(\frac{\|v\|_1\|w\|_1}{s}, \frac{\|v\|_2\|w\|_2}{\sqrt{s}}\right)\right] \leqslant \frac{1}{4}.$$

Consequetly, setting $s = O\left(\min\left(\frac{\|v\|_1\|w\|_1}{\epsilon\langle v,w\rangle}, \frac{\|v\|_2^2\|w\|_2^2}{\epsilon^2\langle v,w\rangle^2}\right)\right)$ suffices for $\mathbb{P}\left[|X - \langle v, w\rangle| \leqslant \epsilon\langle v, w\rangle\right] \geqslant 3/4$. To improve the failure probability to $O(n^{-c})$, it suffices to repeat the sketch $O\left(\log(n^c)\right)$ times and output the median. ☐

SDD Linear System Solvers. In order to compute our effective resistance sketches efficiently, we apply a CountSketch matrix $\mathbf{S}$ to $\mathbf{M}^{\dagger}$, where $\mathbf{M}$ is an SDD matrix. To do this efficiently, we leverage the following theorem.

Theorem 10 (SDD Linear System Solver). *Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be SDD and consider any $\beta > 0$. There exists a randomized algorithm which, with high probability in n, processes a graph in time $\widetilde{O}(\mathrm{nnz}(\mathbf{M}))$ to create access to a linear operator $\mathbf{Q}_\beta \in \mathbb{R}^{n\times n}$ such that $\mathbf{Q}_\beta$ can be applied to any $b \in \mathbb{R}^n$ with $b \perp \ker(\mathbf{M})$ in time $\widetilde{O}(m)$ and $\left\|\mathbf{Q}_\beta b - \mathbf{M}^{\dagger}b\right\|_{\mathbf{M}} \leqslant \beta\left\|\mathbf{M}^{\dagger}b\right\|_{\mathbf{M}}$.*

Many SDD linear system solvers can be viewed as the type of an operator $\mathbf{Q}_\beta$ required in Theorem 10. For a particular example in which this is apparent, consider the operator corresponding to the iterative solver proposed in [52] or the solver from [53]. There is a long line of research on nearly linear time SDD and Laplacian system solvers, beginning with the work of [54] and leading to current state-of-the-art randomized algorithm of [55].

[3] More precisely, when writing the dependence on $\bar{\kappa}(G) := \bar{\kappa}(\mathbf{L}_G)$, [29] has an $\widetilde{O}(m + n\epsilon^{-2}\bar{\kappa}(G)^3 + |S|)$ runtime for the effective resistance estimation problem. This paper instead presents a runtime of $\widetilde{O}(m\epsilon^{-1}\bar{\kappa}(G) + |S|)$ for the effective resistance estimation problem.

6.2.2 Examples of D_M-numerical Sparsity

As we discussed, our spectral sketch data structures in Section 3.2 allow for a restricted set of queries to the pseudoinverse, in particular those that are $\mathbf{D}_M$-numerically sparse as defined in Definition 8. Here, we elaborate on types of queries that are $\mathbf{D}_M$-numerically sparse.

- $\vec{\delta}_{i,j}$ is $(2, \mathbf{D})$-numerically sparse for any invertible $\mathbf{D} \in \mathbb{R}^{n\times n}$. As we are interested primarily in effective resistance estimation in this paper, this provides the primary motivation for studying this class of vectors.
- Standard basis vectors are $(1, \mathbf{D})$-numerically sparse for any invertible $\mathbf{D} \in \mathbb{R}^{n\times n}$.
- When $\mathbf{D}$ is the identity matrix, Definition 8 reduces to the standard definition of numerical sparsity [56].
- More generally, if $x \in \mathbb{R}^n$ is γ-numerically sparse, then it is $\left(\gamma \frac{\max_i d_{i,i}}{\min_i d_{i,i}}, \mathbf{D}\right)$-numerically sparse for any diagonal $\mathbf{D} \in \mathbb{R}^{n\times n}$. Note that if $\mathbf{D}$ is approximately a multiple of the identity, then the $\mathbf{D_M}$-numerical sparsity is approximately equal to the numerical sparsity, up to constants.

The approaches discussed in Section 3.2 apply to all of these examples.

6.2.3 Omitted Proofs form Section 3

In this section, we provide proofs of omitted results from Section 6.2.3. For notational convenience, in this section we define $\mathbf{A}_{\ell M} := \frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{D}_M^{-1/2}\mathbf{A}_M\mathbf{D}_M^{-1/2}$. Note that $\mathbf{N}_M = 2\left(\mathbf{I} - \mathbf{A}_{\ell M}\right)$.

First, let us prove Lemma 4. We first prove the following lemma regarding the power series expansion of $\mathbf{N}_{\ell M} := (\mathbf{N}_M/2)^\dagger$.

Lemma 9. *Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be SDD. For any $x \perp \ker(M)$,*

$$(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} x = \sum_{k=0}^{\infty} \left(\mathbf{A}_{\ell M}\right)^k \mathbf{D}_M^{-1/2} x,$$

and for $m \geqslant 1$,

$$\left\| (\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} x - \sum_{k=0}^{m} \left(\mathbf{A}_{\ell M}\right)^k \mathbf{D}_M^{-1/2} x \right\|_1 \leqslant \frac{\sqrt{n} 2\overline{\kappa}(M)}{\sqrt{d_{\min}}} \exp\left(-\frac{m+1}{2\overline{\kappa}(M)}\right) \|x\|_2.$$

Proof. Let r denote the rank of M. Let $q_1, ..., q_n$ denote orthonormal eigenvectors of $\mathbf{A}_{\ell M}$ associated with $\lambda_1(\mathbf{A}_{\ell M}), ..., \lambda_n(\mathbf{A}_{\ell M})$ respectively, $\mathbf{Q}$ denote the orthogonal matrix whose i-th column is q_i, and $\widetilde{\Lambda}$ denote the diagonal matrix of the $\lambda_i(\mathbf{A}_{\ell M})$'s.

An orthogonal eigendecomposition of $(\mathbf{N}_M/2)^\dagger$ is given by $(\mathbf{N}_M/2)^\dagger = \mathbf{Q}\Lambda\mathbf{Q}^\top$, where Λ is the diagonal matrix whose entries are given by

$$\Lambda_{i,i} = \begin{cases} 0, & i = r+1, ..., n \\ \frac{1}{1-\lambda_i(\mathbf{A}_{\ell M})}, & i = 1, ..., r \end{cases}.$$

So, for any $t \in [r]$, $(\mathbf{N}_M/2)^\dagger q_t = \frac{1}{1-\lambda_t(\mathbf{A}_{\ell M})} q_t$, where $\lambda_t(\mathbf{A}_{\ell M}) \in (0,1)$; and consequently, $0 \leqslant \lambda_t(\mathbf{A}_{\ell M}) < 1$ is in the radius of convergence for the power series of $\frac{1}{1-x}$, and hence

$$\sum_{k=0}^{\infty} \left(\mathbf{A}_{\ell M}\right)^k = \sum_{k=0}^{\infty} \mathbf{Q}\widetilde{\Lambda}^k\mathbf{Q}^\top q_t = q_t \sum_{k=0}^{\infty} \lambda_t^k = \frac{1}{1-\lambda_t(\mathbf{A}_{\ell M})} q_t.$$

Now, $x \perp \ker(M)$ implies $\mathbf{D}_M^{-1/2} x \perp \ker(\mathbf{N}_M)$; and consequently, $\mathbf{D}_M^{-1/2} x$ can be expressed as a linear combination of $q_1, ..., q_r$. The first statement in the lemma now follows by linearity.

For the second statement, note that $\lambda_{\min}(\mathbf{N}_{\ell M}) = \lambda_n(\mathbf{N}_{\ell M})/\overline{\kappa}(M) \geqslant 1/(2\overline{\kappa}(M))$, so $\lambda_r(\mathbf{A}_{\ell M}) \leqslant 1 - 1/(2\overline{\kappa}(M))$. Since $\mathbf{D}_M^{-1/2} x \perp q_{r+1}, ..., q_n$,

$$\left\| (\mathbf{A}_{\ell M})^k \mathbf{D}_M^{-1/2} x \right\|_2 \leqslant \left(1 - \frac{1}{2\overline{\kappa}(M)}\right)^k \frac{\|x\|_2}{\sqrt{d_{\min}}}.$$

Using the fact that $\|x\|_1 \leqslant \sqrt{n}\|x\|_2$ for all x, for $m \geqslant 1$, we have

$$\begin{aligned}
\left\|(1/2\mathbf{N}_M)^\dagger \mathbf{D}_M^{-1/2}x - \sum_{k=0}^{m} (\mathbf{A}_{\ell M})^k \mathbf{D}_M^{-1/2}x\right\|_1 &\leqslant \sqrt{n}\left\|(1/2\mathbf{N}_M)^\dagger \mathbf{D}_M^{-1/2}x - \sum_{k=0}^{m} (\mathbf{A}_{\ell M})^k \mathbf{D}_M^{-1/2}x\right\|_2 \\
&\leqslant \sqrt{n}\sum_{k=m+1}^{\infty} \left\|(\mathbf{A}_{\ell M})^k \mathbf{D}_M^{-1/2}x\right\|_2 \\
&\leqslant \frac{\sqrt{n}2\bar{\kappa}(M)}{\sqrt{d_{\min}}}\left(1-\frac{1}{2\bar{\kappa}(M)}\right)^{m+1}\|x\|_2 \\
&\leqslant \frac{\sqrt{n}2\bar{\kappa}(M)}{\sqrt{d_{\min}}}\exp\left(-\frac{m+1}{2\bar{\kappa}(M)}\right)\|x\|_2.
\end{aligned}$$

□

Using Lemma 9, the proof of Lemma 4 is now straightforward.

Proof of Lemma 4. Let $m \geqslant \max\left(1, 2\bar{\kappa}(M)\log\left(\frac{\sqrt{nd_{\max}}2\bar{\kappa}(M)}{\sqrt{d_{\min}}}\right)\right)$. By Lemma 9,

$$\begin{aligned}
\left\|\mathbf{D}_M^{1/2}\mathbf{N}_{\ell M}^{\dagger}\mathbf{D}_M^{-1/2}x - \sum_{k=0}^{m} D^{1/2}\mathbf{A}_{\ell M}{}^k\mathbf{D}_M^{-1/2}x\right\|_1 &= \left\|\mathbf{D}_M^{1/2}\left(\mathbf{N}_{\ell M}^{\dagger}\mathbf{D}_M^{-1/2}x - \sum_{k=0}^{m} \mathbf{A}_{\ell M}{}^k\mathbf{D}_M^{-1/2}x\right)\right\|_1 \\
&\leqslant \frac{\sqrt{nd_{\max}}2\bar{\kappa}(M)}{\sqrt{d_{\min}}}\exp\left(-\frac{m+1}{2\bar{\kappa}(M)}\right)\|x\|_2 \leqslant 1.
\end{aligned}$$

Using triangle inequality plus the observation that $\mathbf{D}_M^{1/2}\mathbf{A}_{\ell M}\mathbf{D}_M^{-1/2} = 1/2\mathbf{I} + 1/2\mathbf{A}_M\mathbf{D}_M^{-1}$ has each column normalized to have absolute column sum at most 1,

$$\left\|\mathbf{D}_M^{1/2}\mathbf{N}_{\ell M}^{\dagger}\mathbf{D}_M^{-1/2}x\right\|_1 \leqslant \left\|\sum_{k=0}^{m}\mathbf{D}_M^{1/2}\mathbf{A}_{\ell M}{}^k\mathbf{D}_M^{-1/2}x\right\|_1 + 1 \leqslant m\|x\|_1 + 1.$$

□

Next, we will prove our result in Theorem 7. The corresponding algorithm pseudocode for constructing the spectral sketch data structure is given in Algorithm 1. The algorithm pseudocode for querying the spectral sketch data structure is given in Algorithm 2. The proof of Theorem 7 guarantees that it suffices to set $t = \widetilde{O}(1)$ and $s = \widetilde{O}(\bar{\kappa}(\mathbf{M})\,\epsilon^{-1})$ in Algorithm 1 and Algorithm 2. In the following, we use $\|x\|_{\mathbf{A}} := x^\top\mathbf{A}x$ to be the $\mathbf{A}$-seminorm for any PSD matrix $\mathbf{A}$.

Proof of Theorem 7. It suffices to assume, without loss of generality, that S is a set of unit vectors, as at query time, for any vector b we can compute $\|b\|_2$ in $O(\mathrm{nnz}(b))$ time and rescale. Set $\beta = \frac{\min(1,d_{\min}^3)\lambda_{\min}(\mathbf{M})\epsilon}{2\max(1,d_{\max}^2)\sqrt{n\max(1,d_{\max})}}$. By Theorem 10, in $\widetilde{O}(\mathrm{nnz}(\mathbf{M}))$, whp. we can obtain access to a linear operator $\mathbf{Q}_\beta$ such that $\mathbf{Q}_\beta$ can be applied to any $b \in S$ in time $\widetilde{O}(\mathrm{nnz}(\mathbf{M}))$ and $\|\mathbf{Q}_\beta b - \mathbf{M}^\dagger b\|_M \leqslant \beta\|\mathbf{M}^\dagger b\|_M = \beta\|b\|_{\mathbf{M}^\dagger}$. Then,

$$\lambda_{\min}(\mathbf{M})\|(\mathbf{Q}_\beta b - \mathbf{M}^\dagger)b\|_2^2 \leqslant \|(\mathbf{Q}_\beta b - \mathbf{M}^\dagger)b\|_M^2 \leqslant \beta^2\|b\|_{M^\dagger}^2 \leqslant \frac{\beta^2}{\lambda_{\min}(\mathbf{M})}\|b\|_2^2.$$

So, $\|(\mathbf{Q}_\beta b - \mathbf{M}^\dagger)b\|_2 \leqslant \frac{\beta}{\lambda_{\min}(\mathbf{M})} \leqslant \epsilon$. Consequently, by triangle inequality, we have that

$$\|2\mathbf{D}_M\mathbf{Q}_\beta b\|_1 \leqslant \|2\mathbf{D}_M\mathbf{M}^\dagger b\|_1 + \|2\mathbf{D}_M\mathbf{Q}_\beta b - 2\mathbf{D}_M\mathbf{M}^\dagger b\|_1$$

where $2\mathbf{D}_M\mathbf{M}^\dagger b = \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger\mathbf{D}_M^{-1/2}b$ and $\|2\mathbf{D}_M\mathbf{Q}_\beta b - 2\mathbf{D}_M\mathbf{M}^\dagger b\|_1 \leqslant 2\sqrt{nd_{\max}}\|\mathbf{Q}_\beta b - \mathbf{M}^\dagger b\|_2 \leqslant \epsilon$. Consequently,

$$\|2\mathbf{D}_M\mathbf{Q}_\beta b\|_1 \leqslant \|\mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger\mathbf{D}_M^{-1/2}b\|_1 + \epsilon.$$

Now, Lemma 4 guarantees that $\|2\mathbf{D}_M \mathbf{Q}_\beta b\|_1 \leqslant \widetilde{O}(\overline{\kappa}(\mathbf{M}) + \epsilon)\|b\|_1 = \widetilde{O}(\overline{\kappa}(\mathbf{M}))\|b\|_1$. Similarly,

$$\begin{aligned}\left|\langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle - \langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle\right| &= \left|\langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{M}^\dagger b\rangle - \langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle\right| \\ &\leqslant 2 d_{\max} \left\|\mathbf{D}_M^{-1} b\right\|_2 \|(\mathbf{Q}_\beta b - \mathbf{M}^\dagger) b\|_2 \\ &\leqslant 2\frac{d_{\max}}{d_{\min}} \frac{\beta}{\lambda_{\min}(\mathbf{M})} \leqslant \epsilon\left(\frac{1}{2d_{\min}^2}\right).\end{aligned}$$

Lemma 3 guarantees that $\langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle \geqslant O(1)\|\mathbf{D}_{\mathbf{M}}^{-1/2} b\|_2^2$. Consequently,

$$\frac{\|\mathbf{D}_M^{-1} b\|_1 \|2\mathbf{D}_M \mathbf{Q}_\beta b\|_1}{\langle \mathbf{D}_M^{-1/2} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle} = \widetilde{O}(\overline{\kappa}(\mathbf{M})) \frac{\|\mathbf{D}_M^{-1} b\|_1 \|b\|_1}{\|\mathbf{D}_{\mathbf{M}}^{-1/2} b\|_2^2} = \widetilde{O}(\overline{\kappa}(\mathbf{M})) \mathrm{ns}_{\mathbf{D}_M}(b) = \widetilde{O}(\overline{\kappa}(\mathbf{M})).$$

So, using $\widetilde{O}(1)$ copies of a CountSketch matrix $\mathbf{S} \in \mathbb{R}^{\widetilde{O}(\overline{\kappa}(\mathbf{M})\epsilon^{-1}) \times n}$, Lemma 2 guarantees that we can compute an X such that whp.

$$\left|X - \langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle\right| \leqslant \epsilon \langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle.$$

Moreover, we showed above that

$$\begin{aligned}\left|\langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle - \langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle\right| &\leqslant \epsilon\left(\frac{1}{2d_{\min}^2}\right) \\ &\leqslant \epsilon \langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle,\end{aligned}$$

where the last line uses the observation from Lemma 3, that $\langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle \geqslant \frac{1}{2}\|\mathbf{D}_{\mathbf{M}}^{-1} b\|_2^2$. It now follows that

$$\begin{aligned}\left|X - \langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle\right| &\leqslant \epsilon \langle \mathbf{D}_M^{-1} b, 2\mathbf{D}_M \mathbf{Q}_\beta b\rangle + \epsilon \langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle \\ &\leqslant 2\epsilon(1+\epsilon)\langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle \\ &\leqslant 4\epsilon \langle \mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle.\end{aligned}$$

Thus, by Lemma 3, $\frac{1}{2}X \approx_{4\epsilon} \langle \frac{1}{2}\mathbf{D}_M^{-1} b, \mathbf{D}_M^{1/2}(\mathbf{N}_M/2)^\dagger \mathbf{D}_M^{-1/2} b\rangle = \langle b, M^\dagger b\rangle$; hence, rescaling ϵ by a constant factor of 4 yields the approximation guarantee (without changing the size of $\mathbf{S}$ by more than constant factors).

Consequently, we see that by storing $\mathbf{S}\mathbf{D}_M \mathbf{Q}_\beta$, we can support queries $b \in S$. To justify the runtime guarantee, note that Theorem 10 shows we can compute $\mathbf{S}\mathbf{D}_M \mathbf{Q}_\beta$ in $\widetilde{O}(\mathrm{nnz}(\mathbf{M})\overline{\kappa}(\mathbf{M})\epsilon^{-1})$ time, by applying an approximate SDD system solver to each row in $\mathbf{S}$. To support queries, we need only store $\mathbf{S}\mathbf{D}_M \mathbf{Q}_\beta$ and $\mathbf{S}$, which requires only $\widetilde{O}(n\overline{\kappa}(\mathbf{M})\epsilon^{-1})$ bits.

Finally, we justify the query complexity. The key observation is that $\mathbf{S}$ is $\widetilde{O}(1)$-column sparse. Computing X requires taking the median of $\widetilde{O}(1)$ quantities, each of which requires computing an inner product involving an $\widetilde{O}(\mathrm{nnz}(b))$-sparse vector $\mathbf{S}\mathbf{D}_M^{-1} b$. Using this fact, X can be computed in $\widetilde{O}(\mathrm{nnz}(b)^2)$. □

We can now provide a proof of Theorem 2, which follows almost immediately from Theorem 7.

Proof of Theorem 2. Let $G = (V, E, w)$ be a graph with $\widetilde{\Omega}(1)$-expansion. As argued previously (see Section 3.2), $\mathbf{L}_G$ is SDD with $\widetilde{O}(1)$ normalized condition number, $\vec{\delta}_{i,j} \perp \ker \mathbf{L}_G$, and $\vec{\delta}_{i,j}$ is $(2, \mathbf{D}_{\mathbf{L}_G})$- numerically sparse. To see why Theorem 2 holds, simply observe that we can boost the whp. guarantee in Theorem 7, which holds for each fixed query, to hold whp. for all $\vec{\delta}_{i,j}$ for $i, j \in V$ by maintaining $O(\log(n))$ copies of the spectral sketch data structure guaranteed in Theorem 7 as our sketch, and then, at query time, taking the median of the results of the outputs from querying each of the $O(\log(n))$ copies of the sketch. □

Algorithm 1: SpectralSketchSDDInverse

Input: A matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, error tolerance $\epsilon \in (0,1)$, and integers $s, t > 0$.

Output: A CountSketch matrix $\mathbf{S} \in \mathbb{R}^{(3t)s \times n}$ and $\widetilde{\mathbf{S}} \in \mathbb{R}^{(3t)s \times n}$, an approximation of $2\mathbf{S}\mathbf{D}_{\mathbf{M}}\mathbf{M}^{\dagger}$

Set β as in the proof of Theorem 7;
Generate a random CountSketch matrix $\mathbf{S} \in \mathbb{R}^{(3t)s \times n}$ (see Theorem 4 of [42]) ;
Generate access to a linear operator $\mathbf{Q}_{\beta}$ such that for all $b \perp \ker(\mathbf{M})$, $\|\mathbf{Q}_{\beta}\mathbf{b} - \mathbf{M}^{\dagger}\mathbf{b}\|_{M} \leqslant \beta\|M^{\dagger}b\|_{\mathbf{M}}$ (See Theorem 10);
Compute $\widetilde{\mathbf{S}} = 2\mathbf{S}\mathbf{D}_{\mathbf{M}}\mathbf{Q}_{\beta}$;
return $(\mathbf{S}, \widetilde{\mathbf{S}})$

Algorithm 2: QuerySketchSDDInverse

Input: $\mathbf{D}_{\mathbf{M}}$, output of Algorithm 1, query vector b, and integers $s, t > 0$ as inputted to Algorithm 1.

Output: An approximation to $\langle b, \mathbf{M}^{\dagger}b\rangle$.

Set $\widetilde{b} = b/\|b\|_2$;
for $i \in [3t]$ **do**
  $x_i = \frac{1}{2}\langle(\mathbf{S}\mathbf{D}_{M}^{-1}\widetilde{b})[(i-1)s+1:(i)s]i, (\widetilde{\mathbf{S}}\widetilde{b})[(i-1)s+1:(i)s]\rangle$;
end
return $\|b\|_2^2\mathrm{median}\{x_i\}$.

Theorem 3 is now a direct corollary of Theorem 2. Finally, we provide the proof of Theorem 8.

Proof of Theorem 8. Let $x \perp \ker(\mathbf{A})$. Note that for any $\|x\|_2 \|\mathbf{A}x\|_2 = \kappa(\mathbf{A}^{1/2})x^{\top}\mathbf{A}x$. So, by the ℓ_2 norm guarantees from Lemma 2, it suffices to build a CountSketch matrix S with $s = \widetilde{O}\left(\kappa(\mathbf{A})\epsilon^{-2}\right)$ to guarantee that for any $x \perp \ker(\mathbf{A})$, $\langle \mathbf{S}x, \mathbf{S}\mathbf{A}x\rangle \approx_{\epsilon} \langle x, \mathbf{A}x\rangle$. The time to compute the sketch $\mathbf{SA}$ is at most $\widetilde{O}(\kappa(\mathbf{A})\mathrm{nnz}(\mathbf{A})\epsilon^{-2})$. Due to the column-sparsity of $\mathbf{S}$, $\mathbf{S}x$ is $\widetilde{O}(\mathrm{nnz}(x))$ sparse, and consequently, $\langle \mathbf{S}x, \mathbf{S}\mathbf{A}x\rangle$ can be computed in $\widetilde{O}(\mathrm{nnz}(x)^2)$ time. ☐

6.3 Lower Bounds

Here we formalize the approach for lower bounds that was discussed in Section 4.

6.3.1 Proofs for Lower Bounds on Effective Resistance Estimation

In this section, we present proofs that were omitted from Section 4.2. First, we formalize a standard randomized reduction from triangle detection in general graphs to triangle detection in tripartite graphs in Lemma 10 below.

Lemma 10. *Given an algorithm which can solve the triangle detection problem on an n-node tripartite undirected graph in $O(n^{\gamma})$ time, we can produce a randomized algorithm which can solve the triangle detection problem on an arbitrary n-node undirected graph G in $\widetilde{O}(n^{\gamma})$ time whp.*

Proof. We first sample a tripartite subgraph H of G by assigning each vertex in G to a random tripartition with equal probability 1/3 and deleting edges within each resulting tripartition. First, note that if G has no triangles then H also has no triangles since H is a subgraph of G. Second, observe that if G has a triangle $\{a, b, c\}$, it is also a triangle in H if each vertex ends up in a different tripartition. This occurs with probability at least $(1/3)^3 = 1/27$ which is a constant. Therefore, solving the triangle detection problem on H and returning the same output also successfully solves the triangle detection problem on G with probability at least $1/27$. We can repeat this randomized procedure $\log(n^c)$ times to boost the success probability to at least $1 - n^{-c}$, which is whp. in n. This randomized algorithm runs in $\widetilde{O}(n^{\gamma})$ time, completing the proof. ☐

Next, we establish some crucial properties of symmetric random signing that enables our results. First, we show in Lemma 11 below that the symmetric random signing of the adjacency matrix of a graph leads to a smaller spectral radius.

Lemma 11. *Let $G = (V, E)$ be an undirected unweighted graph on n nodes. Let $\bar{\mathbf{A}}_G$ denote a symmetric random signing of $\mathbf{A}_G$. With high probability, $\rho(\bar{\mathbf{A}}) \leqslant \widetilde{O}(\sqrt{n})$.*

Proof. Let $\sigma^2 := \left\| \sum_{\{u,v\}\in E} \left(\mathbf{E}_{u,v}\right)^2 \right\|_2$ where $\mathbf{E}_{u,v}$ is the adjacency matrix of a graph with only a single edge between u and v. Note that entries of $\mathbf{E}_{u,v}^2$ indicates paths of length two in this graph. Therefore, this is a diagonal matrix that satisfies $\left(\mathbf{E}_{u,v}^2\right)_{i,i} = 1$ if and only if $i \in \{u, v\}$. Consequently,

$$\sigma^2 = \left\| \sum_{\{u,v\}\in E} \left(\mathbf{E}_{u,v}\right)^2 \right\|_2 = \|\mathbf{D}\|_2 \leqslant d_{\max} \leqslant n.$$

We can now write $\bar{\mathbf{A}} = \sum_{\{u,v\}\in E} \xi_{u,v}\mathbf{E}_{u,v}$ and apply the Matrix Rademacher concentration result (Theorem 1.2) from [44], to get that for any constant $c > 1$,

$$\mathbb{P}\left[\lambda_n\left(\bar{\mathbf{A}}\right) \geqslant \sqrt{d_{\max}}\log(cn)\right] \leqslant n\exp\left(\frac{-cd_{\max}\log(n)}{2d_{\max}}\right) = n\frac{1}{n^c} = n^{-c+1}.$$

□

Next, we prove Lemma 6 presented in Section 4.2 which states that the symmetric random signing preserves the non-zeroness of the entries of $\mathbf{A}_G^2$ with constant probability.

Lemma 6. *For $i \neq j$, if $(\mathbf{A}_G^2)_{i,j} = 0$, then $(\bar{\mathbf{A}}_G^2)_{i,j} = 0$; if $(\mathbf{A}_G^2)_{i,j} > 0$, $\mathbb{P}\left[|(\bar{\mathbf{A}}_G^2)_{i,j}| > 1\right] \geqslant 1/2$.*

Proof. Note that

$$\left(\bar{\mathbf{A}}^2\right)_{i,j} = \sum_{k=1}^{n} \bar{\mathbf{A}}_{i,k}\bar{\mathbf{A}}_{k,j}$$

If $\mathbf{A}_{i,j}^2 = 0$, then G has no path of length exactly two between i and j, so each term $\mathbf{A}_{i,k}\mathbf{A}_{k,j}$ in the summation above must be zero, and hence $\left(\bar{\mathbf{A}}^2\right)_{i,j} = 0$, completing the first part of the lemma.

Since $\left(\bar{\mathbf{A}}^2\right)_{i,j}$ is only supported on the integers, to prove the second statement, it suffices to show that when $\left(\mathbf{A}^2\right)_{i,j} > 0$, $\mathbb{P}\left[\left|\left(\bar{\mathbf{A}}^2\right)_{i,j}\right| = 0\right]$ is no larger than 1/2. To see this, note that if $\mathbf{A}_{i,j}^2 \neq 0$, then then G has at least one path of length exactly 2 between i and j. That is, there exists a $k' \in [n]$ such that $\mathbf{A}_{i,k'}\mathbf{A}_{k',j} = 1$. We can write

$$\left(\bar{\mathbf{A}}^2\right)_{i,j} = \sum_{k=1}^{n} \xi_{i,k}\xi_{k,j}\mathbf{A}_{i,k}\mathbf{A}_{k,j}$$

Because $i \neq j$, for every $k, \ell \in [n]$ each $\xi_{i,k}$ is always independent from any other $\xi_{\ell,j}$ term appearing in the sum. Moreover, if $\xi_{i,k}$ appears in the sum, then $\xi_{k,i}$ never appears in the sum. Therefore, for each $\{\xi_{i,k}\xi_{k,j}\}_{k\in[n]}$ are themselves independently chosen Rademacher random variables, and all terms in the summation are independent. Separating out the k'-th entry,

$$\left(\bar{\mathbf{A}}^2\right)_{i,j} = \xi_{i,k'}\xi_{k',j} + \sum_{k\neq k'}^{n} \xi_{i,k}\xi_{k,j}\mathbf{A}_{i,k}\mathbf{A}_{k,j} \stackrel{D}{=} \xi' + Z,$$

where ξ' is a Rademacher random variable and $Z := \sum_{k\neq k'}^{n} \xi_{i,k}\xi_{k,j}\mathbf{A}_{i,k}\mathbf{A}_{k,j}$ is independent from ξ'. For any value of Z, $\mathbb{P}[\xi' = -Z] \leqslant \frac{1}{2}$. Therefore,

$$\mathbb{P}\left[\left(\bar{\mathbf{A}}^2\right)_{i,j} = 0\right] \leqslant \frac{1}{2},$$

completing the second part of the result. □

Now, we prove Lemma 5 in two steps. First, we show in Lemma 12 that given an algorithm to solve the all pairs effective resistance estimation problem on expanders, we can produce an algorithm to solve the SDD effective resistance estimation problem specifically on SDD matrices with non-positive offidiagonals. Then, we show in Lemma 14 that this algorithm can be used to produce an algorithm to solve the general SDD effective resistance estimation problem.

Lemma 12. *Given an algorithm that solves the all pairs effective resistance estimation problem on graphs with $\widetilde{\Omega}(1)$-expansion in $\widetilde{O}(n^2\epsilon^{-c})$ time for some $c > 0$, we can produce an algorithm which takes as input an SDD matrix $\mathbf{M} = \mathbf{I} - \mathbf{Q}$ such that $\mathbf{Q}$ is entrywise non-negative and $\rho(\mathbf{Q}) \leqslant \frac{1}{3}$, and solves the SDD effective resistance estimation problem for $\mathbf{M}$ in $\widetilde{O}(n^2\epsilon^{-c})$ time.*

Proof. Let $v := (\mathbf{I} - \mathbf{Q})\mathbb{1}$. Note that v is entrywise non-negative. Consider the matrix

$$\mathbf{L} := \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \|v\|_1 \end{pmatrix} - \begin{pmatrix} \mathbf{Q} & v \\ v^\top & 0 \end{pmatrix}.$$

and note that it is the Laplacian matrix. Since $\mathbf{M}$ is a principal submatrix of $\mathbf{L}$, we have by the eigenvalue interlacing theorem [57] that $\lambda_1(\mathbf{M}) \leqslant \lambda_2(\mathbf{L})$. But since $\rho(\mathbf{Q}) \leqslant 1/3$, we have that $\lambda_1(\mathbf{M}) \leqslant 2/3$ and consequently $\lambda_2(\mathbf{L}) \geqslant 2/3$. Therefore, $\mathbf{L}$ is the Laplacian of an $\widetilde{\Omega}(1)$-expander (see Cheeger's Inequality Theorem 6). We claim that for $i, j \in n$,

$$\vec{\delta}_{i,j}^\top \mathbf{L}^\dagger \vec{\delta}_{i,j} = \vec{\delta}_{i,j}^\top \mathbf{M}^{-1} \vec{\delta}_{i,j},$$

which is sufficient to prove the lemma. Note that for any $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, and $\alpha \in \mathbb{R}$,

$$\mathbf{L}\begin{pmatrix} x + \alpha\mathbb{1} \\ y + \alpha \end{pmatrix} = \begin{pmatrix} (\mathbf{I} - \mathbf{Q})x + v^\top x \\ v^\top x - \|v\|_1 y \end{pmatrix}.$$

Let

$$\mathbf{L}^\dagger \vec{\delta}_{i,j} = \begin{pmatrix} z_x \\ z_y \end{pmatrix}.$$

Then, note that

$$\mathbf{L}\begin{pmatrix} z_x - z_y\mathbb{1} \\ 0 \end{pmatrix} = \begin{pmatrix} \vec{\delta}_{i,j} \\ 0 \end{pmatrix} \implies (\mathbf{I} - \mathbf{Q})(z_x - z_y\mathbb{1}) = \vec{\delta}_{i,j}.$$

Consequently,

$$\vec{\delta}_{i,j}^\top (\mathbf{I} - \mathbf{Q})^{-1} \vec{\delta}_{i,j} = \vec{\delta}_{i,j}^\top (z_x - z_y\mathbb{1}) = \vec{\delta}_{i,j}^\top z_x - z_y \vec{\delta}_{i,j}^\top \mathbb{1} = \vec{\delta}_{i,j}^\top z_x = \vec{\delta}_{i,j}^\top \mathbf{L}^\dagger \vec{\delta}_{i,j}.$$

□

Now, to prove Lemma 14, we first show a useful property of block symmetric matrices in the helper lemma below.

Lemma 13. *Suppose $\mathbf{A} = \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y} & \mathbf{X} \end{pmatrix}$ where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n}$. Then,*

$$\vec{\delta}_{i,j}^\top (\mathbf{X} - \mathbf{Y}) \vec{\delta}_{i,j} = \frac{1}{2}\left[\vec{\delta}_{i,n+i}^\top \mathbf{A} \vec{\delta}_{i,n+i} + \vec{\delta}_{j,n+j}^\top \mathbf{A} \vec{\delta}_{j,n+j}\right] - \vec{\delta}_{i,n+j}^\top \mathbf{A} \vec{\delta}_{i,n+j} + \vec{\delta}_{i,j}^\top \mathbf{A} \vec{\delta}_{i,j}.$$

Proof. Note that

$$\vec{\delta}_{i,j}^\top (\mathbf{X} - \mathbf{Y}) \vec{\delta}_{i,j} = (\mathbf{X}_{i,i} + \mathbf{X}_{j,j}) - 2\mathbf{X}_{i,j} + 2\mathbf{Y}_{i,j} - (\mathbf{Y}_{i,i} + \mathbf{Y}_{j,j}).$$

Meanwhile,

$$\begin{aligned} \frac{1}{2}\vec{\delta}_{i,n+i}^\top \mathbf{A} \vec{\delta}_{i,n+i} &= (\mathbf{X}_{i,i} - \mathbf{Y}_{i,i}), \\ \frac{1}{2}\vec{\delta}_{j,n+j}^\top \mathbf{A} \vec{\delta}_{j,n+j} &= (\mathbf{X}_{j,j} - \mathbf{Y}_{j,j}), \\ -\vec{\delta}_{i,n+j}^\top \mathbf{A} \vec{\delta}_{i,n+j} &= -\mathbf{X}_{i,i} + 2\mathbf{Y}_{i,j} - \mathbf{X}_{j,j}, \\ \vec{\delta}_{i,j}^\top \mathbf{A} \vec{\delta}_{i,j} &= \mathbf{X}_{i,i} - 2\mathbf{X}_{i,j} + \mathbf{X}_{j,j}, \end{aligned}$$

and adding these four terms together concludes the proof. □

Lemma 14. *Suppose we are given an algorithm which takes as input an SDD matrix $\mathbf{M} = \mathbf{I} - \mathbf{Q}$ such that $\mathbf{Q}$ is entrywise non-negative and $\rho(\mathbf{Q}) \leqslant \frac{1}{3}$, and solves the SDD effective resistance estimation problem for $\mathbf{M}$ in $\widetilde{O}(n^2\epsilon^{-c})$ time. Then, we can produce an algorithm which takes as input an SDD matrix $\mathbf{M}' = \mathbf{I} - \mathbf{Q}'$ and $\rho(\mathbf{Q}') \leqslant \frac{1}{3}$, and solves the SDD effective resistance estimation problem for $\mathbf{M}'$ in $\widetilde{O}(n^2\epsilon^{-c})$ time.*

Proof. We can decompose $\mathbf{Q}'$ as $\mathbf{Q}' = \mathbf{P} - \mathbf{N}$ where $\mathbf{P}$ is a matrix which contains only the positive offdiagonal entries of $\mathbf{Q}'$ and $-\mathbf{N}$ is a matrix which contains all the negative offdiagonal entries. Therefore both $\mathbf{P}$ and $\mathbf{N}$ themselves are entrywise non-negative. We define

$$\mathbf{Q} := \begin{pmatrix} \mathbf{P} & \mathbf{N} \\ \mathbf{N} & \mathbf{P} \end{pmatrix}.$$

Note that $\mathbf{Q}$ is also entrywise non-negative. We also have $\rho(\mathbf{Q}) \leqslant 1/3$. To see this, assume for the sake of contradiction that $\mathbf{Q}$ has an eigenvalue $\lambda > 1/3$. This means that there must exist some $x \in \mathbb{R}^{2n}$ such that $\mathbf{Q}x = \lambda x$. Let $x = [x_1; x_2]$ where $x_1, x_2 \in \mathbb{R}^n$. The eigenvalue equation then implies that $\mathbf{P}x_1 + \mathbf{N}x_2 = \lambda x_1$ and $\mathbf{N}x_1 + \mathbf{P}x_2 = \lambda x_2$. Subtracting these equations yields $(\mathbf{P} - \mathbf{N})(x_1 - x_2) = \lambda(x_1 - x_2)$ which means that λ is also an eigenvalue of $\mathbf{Q}'$. This is a contradiction since $\rho(\mathbf{Q}') \leqslant 1/3$. Therefore, $\rho(\mathbf{Q}) \leqslant 1/3$.

Now, consider the following block decomposition of $\mathbf{Q}^k$

$$\mathbf{Q}^k = \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y} & \mathbf{X} \end{pmatrix}$$

where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n}$. We will show by induction that $\mathbf{Q}'^k = (\mathbf{P} - \mathbf{N})^k = \mathbf{X} - \mathbf{Y}$. In the base case, when $k = 1$, this is trivially true. Now, assume that the claim holds for all $k \leqslant m$ for some m. Consider the following block decomposition of $\mathbf{Q}^m$

$$\mathbf{Q}^m = \begin{pmatrix} \mathbf{W} & \mathbf{Z} \\ \mathbf{Z} & \mathbf{W} \end{pmatrix}.$$

By the inductive hypothesis, we know that $\mathbf{Q}'^m = (\mathbf{P} - \mathbf{N})^m = \mathbf{W} - \mathbf{Z}$. Now, we have that

$$\mathbf{Q}^{m+1} = \begin{pmatrix} \mathbf{W} & \mathbf{Z} \\ \mathbf{Z} & \mathbf{W} \end{pmatrix} \begin{pmatrix} \mathbf{P} & \mathbf{N} \\ \mathbf{N} & \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{WP} + \mathbf{ZN} & \mathbf{WN} + \mathbf{ZP} \\ \mathbf{WN} + \mathbf{ZP} & \mathbf{WP} + \mathbf{ZN} \end{pmatrix},$$

and also that

$$\mathbf{Q}'^{m+1} = (\mathbf{W} - \mathbf{Z})(\mathbf{P} - \mathbf{N}) = \mathbf{WP} - \mathbf{ZP} - \mathbf{WN} + \mathbf{ZN} = (\mathbf{WP} + \mathbf{ZN}) - (\mathbf{WN} + \mathbf{ZP}).$$

Hence, the claim follows by induction. By Lemma 13, it follows that

$$\vec{\delta}_{i,j}^{\top} \mathbf{Q}'^k \vec{\delta}_{i,j} = \frac{1}{2}\left[\vec{\delta}_{i,n+i}^{\top} \mathbf{Q}^k \vec{\delta}_{i,n+i} + \vec{\delta}_{j,n+j}^{\top} \mathbf{Q}^k \vec{\delta}_{j,n+j}\right] - \vec{\delta}_{i,n+j}^{\top} \mathbf{Q}^k \vec{\delta}_{i,n+j} + \vec{\delta}_{i,j}^{\top} \mathbf{Q}^k \vec{\delta}_{i,j}. \tag{4}$$

Now we can use the power series expansion of $(\mathbf{I} - \mathbf{Q})^{-1}$ to say, for any $u, v \in [2n]$,

$$\vec{\delta}_{u,v}^{\top} (\mathbf{I} - \mathbf{Q})^{-1} \vec{\delta}_{u,v} = \sum_{k=0}^{\infty} \vec{\delta}_{u,v}^{\top} \mathbf{Q}^k \vec{\delta}_{u,v}.$$

Similarly, for any $i, j \in [n]$,

$$\vec{\delta}_{i,j}^{\top} (\mathbf{I} - \mathbf{Q}')^{-1} \vec{\delta}_{i,j} = \sum_{k=0}^{\infty} \vec{\delta}_{i,j}^{\top} \mathbf{Q}'^k \vec{\delta}_{i,j}.$$

So, by linearity and (4), it follows that

$$\begin{aligned} \vec{\delta}_{i,j}^{\top} (\mathbf{I} - \mathbf{Q}')^{-1} \vec{\delta}_{i,j} =& \frac{1}{2}\left[\vec{\delta}_{i,n+i}^{\top} (\mathbf{I} - \mathbf{Q})^{-1} \vec{\delta}_{i,n+i} + \vec{\delta}_{j,n+j}^{\top} (\mathbf{I} - \mathbf{Q})^{-1} \vec{\delta}_{j,n+j}\right] \\ &- \vec{\delta}_{i,n+j}^{\top} (\mathbf{I} - \mathbf{Q})^{-1} \vec{\delta}_{i,n+j} + \vec{\delta}_{i,j}^{\top} (\mathbf{I} - \mathbf{Q})^{-1} \vec{\delta}_{i,j}, \end{aligned}$$

and this completes the proof. $\square$

Lemma 5 follows directly from Lemma 12 and Lemma 14. Finally, we prove Theorem 9 below.

Theorem 9. *Given an algorithm which solves the SDD effective resistance estimation problem in $\widetilde{O}(n^2\epsilon^{-c})$ time, we can produce a randomized algorithm that solves the triangle detection problem in $\widetilde{O}(n^{2(1+c)})$ time whp.*

Proof. As G is tripartite, let $V = V_1 \sqcup V_2 \sqcup V_3$ be the partition of G such that no edge has both endpoints in V_i for some $i \in [3]$. Let $E_{1,2} := \{\{u,v\} \in E : u \in V_1, v \in V_2\}$, and let $H := (V, E \backslash E_{1,2})$. Let $\mathbf{A} = \mathbf{A}_H$ denote the adjacency matrix of H and let $\bar{\mathbf{A}}$ be a random signing of $\mathbf{A}$.

Suppose that G has a triangle. Then, there exists a pair of vertices $u \in V_1, v \in V_2$ such that $\{u,v\} \in E_{12}$ and H contains a path of length two between u and v. Furthermore, observe that because there are no edges between V_1 and V_2 in H, H has no paths of length three between V_1 and V_2. Consequently, in order to find a triangle in G, it suffices to check, for each $i \in V_1, j \in V_2$ with $\{i,j\} \in E_{1,2}$, whether there exists a path of length two between node i and node j in H. In other words, we need to check if there exists some $\{i,j\} \in E_{1,2}$ such that $\mathbf{A}^2_{i,j} > 0$. By Lemma 6, we can instead check if $\left|\bar{\mathbf{A}}^2_{i,j}\right| > 0$ and we would still correctly detect a triangle with probability at least 1/2. Note that this check requires requires only $O(\mathrm{nnz}(\mathbf{A}))$ additional time, since $|E_{1,2}| < \mathrm{nnz}(\mathbf{A})$.

So, our goal now is to compute an accurate enough estimate of $\left|\bar{\mathbf{A}}_{i,j}\right|^2$ for all $\{i,j\} \in E_{1,2}$ given the effective resistance estimate $\tilde{r}_{i,j}$. To this end, let $\mathbf{N} = \left(\mathbf{I} - \frac{\alpha}{n}\bar{\mathbf{A}}\right)^{-1}$ for some $\alpha < \frac{1}{3}$. Note that the max row-sum of $\bar{\mathbf{A}}$ is 1/3, so the inverse exists. Lemma 11 guarantees that with high probability, $\rho\left(\bar{\mathbf{A}}\right) \leqslant \widetilde{O}(\sqrt{d_{\max}}) = \widetilde{O}(\sqrt{n})$. We condition on this event in the remainder of the proof. Consequently, we can express $\mathbf{N}$ as a power series,

$$\mathbf{N} = \sum_{k=0}^{\infty} \left(\frac{\alpha}{n}\right)^k \bar{\mathbf{A}}^k.$$

Now, let $\widetilde{\mathbf{N}}$ denote the truncation of $\mathbf{N}$ at the third term in this power series. That is, $\widetilde{\mathbf{N}} = \mathbf{I} + \frac{\alpha}{n}\bar{\mathbf{A}} + \frac{\alpha^2}{n^2}\bar{\mathbf{A}}^2$. Therefore, we have for all $\{i,j\} \in E_{1,2}$,

$$\bar{\mathbf{A}}^2_{i,j} = \frac{n^2}{\alpha^2}\widetilde{\mathbf{N}}_{i,j}.$$

Noticing that $\mathbf{N}_{i,j} = \frac{\mathbf{N}_{i,i}+\mathbf{N}_{j,j}-r_{i,j}}{2}$ motivates us to define our estimate of $\bar{\mathbf{A}}^2_{i,j}$ that we denote $P_{i,j}$ as follows

$$P_{i,j} := \frac{n^2}{\alpha^2}\left[\frac{\widetilde{\mathbf{N}}_{i,i} + \widetilde{\mathbf{N}}_{j,j} - \tilde{r}_{i,j}}{2}\right].$$

Now, observe that $\widetilde{\mathbf{N}}_{i,i} = 1 + \frac{\alpha^2}{n^2}\bar{\mathbf{A}}^2_{i,i}$, and $\bar{\mathbf{A}}^2_{i,i}$ is simply the degree of vertex i in H. Note that the random signing does not affect the fact that the diagonal entries of the square of the adjacency matrix are the degrees. Therefore, $\widetilde{\mathbf{N}}_{i,i}$ can also be computed for all i in only $O(\mathrm{nnz}(\mathbf{A}))$ time. The additive error between our estimate $P_{i,j}$ and $\left|\bar{\mathbf{A}}\right|^2_{i,j}$ takes the form

$$\left|P_{i,j} - \bar{\mathbf{A}}^2_{i,j}\right| = \left|\frac{n^2}{\alpha^2}\left[\widetilde{\mathbf{N}}_{i,j} - \frac{\widetilde{\mathbf{N}}_{i,i} + \widetilde{\mathbf{N}}_{j,j} - \tilde{r}_{i,j}}{2}\right]\right|.$$

By triangle inequality and plugging in the definition of $r_{i,j}$, we break up the error into four pieces

$$\left|P_{i,j} - \bar{\mathbf{A}}^2_{i,j}\right| \leqslant \frac{n^2}{\alpha^2}\left[\left|\widetilde{\mathbf{N}}_{i,j} - \mathbf{N}_{i,j}\right| + \left|\widetilde{\mathbf{N}}_{i,i} - \mathbf{N}_{i,i}\right|/2 + \left|\widetilde{\mathbf{N}}_{j,j} - \mathbf{N}_{j,j}\right|/2 + \left|\tilde{r}_{i,j} - r_{i,j}\right|/2\right]. \quad (5)$$

We now bound each term separately. Consider any $i \in V_1$. Since, H contains no triangle containing i, $\left(\bar{\mathbf{A}}^3\right)_{i,i} = 0$. So, we have that

$$\left|\mathbf{N}_{i,i} - \widetilde{\mathbf{N}}_{i,i}\right| = \left|\sum_{k=4}^{\infty} \frac{\alpha^k}{n^k}\left(\bar{\mathbf{A}}^k\right)_{i,i}\right| \leqslant \left\|\sum_{k=4}^{\infty} \frac{\alpha^k}{n^k}\bar{\mathbf{A}}^k\right\|_2$$

$$\leqslant \sum_{k=4}^{\infty} \widetilde{O}\left(\frac{\alpha}{\sqrt{n}}\right)^k = \frac{\widetilde{O}(\frac{\alpha}{\sqrt{n}})^4}{1-\widetilde{O}(\frac{\alpha}{\sqrt{n}})}$$

Similarly, for any $i \in V_1$ and $j \in V_2$, note that $\bar{\mathbf{A}}^3_{i,j} = 0$ because H has no edges between V_1 and V_2, and there are no edges within each tripartition V_i. Consequently, by a similar argument as above,

$$\begin{aligned}
\left|\mathbf{N}_{i,j} - \widetilde{\mathbf{N}}_{i,j}\right| &= \sum_{k=4}^{\infty} (\bar{\mathbf{A}}^4)_{i,j} = \frac{1}{2}\sum_{k=4}^{\infty} e_i^\top(\bar{\mathbf{A}}^k)e_i + e_j^\top(\bar{\mathbf{A}}^k)e_j - \vec{\delta}_{i,j}^\top(\bar{\mathbf{A}}^k)\vec{\delta}_{i,j} \\
&= \frac{1}{2}\left(e_i^\top e_i + e_j^\top e_j + \vec{\delta}_{i,j}^\top\vec{\delta}_{i,j}\right)\left\|\sum_{k=4}^{\infty}\frac{\alpha^k}{n^k}\bar{\mathbf{A}}^k\right\|_2 \\
&\leqslant 2\sum_{k=4}^{\infty}\widetilde{O}\left(\frac{\alpha}{\sqrt{n}}\right)^k = \frac{2\widetilde{O}(\frac{\alpha}{\sqrt{n}})^4}{1-\widetilde{O}(\frac{\alpha}{\sqrt{n}})}.
\end{aligned}$$

Finally, consider the magnitude of the approximation error between $\widetilde{r}_{i,j}$ and $r_{i,j}$. We have

$$|r_{i,j} - \widetilde{r}_{i,j}| \leqslant \epsilon\,|r_{i,j}|\,.$$

Note that $|r_{i,j}| = \left|\vec{\delta}_{i,j}^\top\mathbf{N}\vec{\delta}_{i,j}\right| \leqslant \|\vec{\delta}_{i,j}\|^2\|\mathbf{N}\| \leqslant 2\|\left(\mathbf{I} - \frac{\alpha}{n}\bar{\mathbf{A}}\right)^{-1}\| \leqslant \frac{2}{1-\frac{\alpha}{n}\|\bar{\mathbf{A}}\|_2} \leqslant \frac{2}{1-\widetilde{O}(\frac{\alpha}{\sqrt{n}})}$. Plugging these estimates of the errors into (5), we get

$$\left|P_{i,j} - \bar{\mathbf{A}}^2_{i,j}\right| \leqslant \frac{n^2}{\alpha^2}\left[\frac{\widetilde{O}\left(\frac{\alpha}{\sqrt{n}}\right)^4 + \epsilon}{1-\widetilde{O}\left(\frac{\alpha}{\sqrt{n}}\right)}\right].$$

Since $\left|\bar{\mathbf{A}}^2_{i,j}\right| \in \{0,1\}$, to compute $\bar{\mathbf{A}}^2_{i,j}$, we need approximate it to additive 1/2 error. That is, we require

$$\frac{1}{1-\widetilde{O}(\frac{\alpha}{\sqrt{n}})}\left[\widetilde{O}(\alpha^2) + \epsilon\frac{n^2}{\alpha^2}\right] < \frac{1}{2},$$

or equivalently,

$$\epsilon < \frac{\alpha^2 - \widetilde{O}(\frac{\alpha^3}{\sqrt{n}})}{2n^2} - \frac{\widetilde{O}(\alpha^4)}{n^2} = \widetilde{O}\left(\frac{1}{n^2}\right),$$

where the last step follows from the fact that we can take α to be a sufficiently small constant. Therefore, estimates $\widetilde{r}_{i,j}$ with $\epsilon = \widetilde{O}(n^{-2})$ for all $\{i,j\} \in E_{1,2}$ are sufficient to determine if $\left|\bar{\mathbf{A}}^2_{i,j}\right|$ is 0 or 1. As noted earlier, checking this for all edges in $E_{1,2}$ takes $O(\mathrm{nnz}(\mathbf{A}))$ additional time. Therefore, by plugging in $\epsilon = \widetilde{O}(n^{-2})$, we can use an algorithm that solves the SDD effective resistance estimation problem in $\widetilde{O}(n^2\epsilon^{-c})$ time to solve the triangle detection problem in $\widetilde{O}(n^2 n^{2c}) = \widetilde{O}(n^{2(1+c)})$ time whp. and this completes the proof. □

Lemma 7. *Given an algorithm to solve the all edges effective resistance estimation problem (i.e., Definition 1 where $S = E$) in $\widetilde{O}(m\epsilon^{-c})$ time, we can produce an algorithm to solve the all pairs effective resistance estimation problem in $\widetilde{O}(n^2\epsilon^{-c})$ time for some $c > 0$.*

Proof. The idea is that, given a graph G on n nodes and m edges, we can always add to it a complete graph of edges of sufficiently small weight that would not change the effective resistances much. Let $\mathbf{L}_G$ be the graph Laplacian and H be the graph obtained by adding a complete graph with uniform edge weight $\alpha > 0$. Then, $\mathbf{L}_H = \mathbf{L}_G + \alpha(n\mathbf{I} - \mathbb{1}\mathbb{1}^T)$. It suffices to find α small enough such that $x^T\mathbf{L}_H^\dagger x \approx_\epsilon x^T\mathbf{L}_G^\dagger x$ for all $x \perp \mathbb{1}$. First, we use the fact that if $x^T\mathbf{L}_H x \approx_\epsilon x^T\mathbf{L}_G x$, then we also have $x^T\mathbf{L}_H^\dagger x \approx_\epsilon x^T\mathbf{L}_G^\dagger x$ [28]. To show the former, we need

$$x^T(\mathbf{L}_G + \alpha(n\mathbf{I} - \mathbb{1}\mathbb{1}^T) - \mathbf{L}_G)x \leqslant \epsilon x^T\mathbf{L}_G x$$

$$\implies \alpha \leqslant \epsilon \frac{x^T \mathbf{L}_G x}{x^T (n\mathbf{I} - \mathbb{1}\mathbb{1}^T) x}$$

for all $x \perp \mathbb{1}$. Therefore, taking $\alpha \leqslant \epsilon \frac{\lambda_{\min}(\mathbf{L}_G)}{n}$ is sufficient to have $x^T \mathbf{L}_H^\dagger x \approx_\epsilon x^T \mathbf{L}_G^\dagger x$ for all $x \perp \mathbb{1}$. As a consequence, we can estimate all pairs effective resistances on G by estimating all edges effective resistances on H. So, any combinatorial $\tilde{O}(m\epsilon^{-c})$ time algorithm for the all edges effective resistance estimation problem would immediately imply a combinatorial $\tilde{O}(n^2\epsilon^{-c})$ time algorithm for all pairs effective resistance estimation. □

6.3.2 Proofs for Lower Bounds on Spectral Sum Estimation

In this section, we present proofs for the results presented in Section 4.3. First, we prove Lemma 8 below.

Lemma 8. *If* $\mathrm{tr}\left(\mathbf{A}_G^3\right) = 0$*, then* $\mathrm{tr}\left(\bar{\mathbf{A}}_G^3\right) = 0$*, and if* $\mathrm{tr}\left(\mathbf{A}_G^3\right) > 0$ *then* $\mathbb{P}\left[\left|\mathrm{tr}\left(\bar{\mathbf{A}}_G^3\right)\right| > 0\right] \geqslant 1/4$.

Proof. Again for convenience, let $\bar{\mathbf{A}} = \bar{\mathbf{A}}_G$. Denote by $\xi_{ab} = \xi_{ba}$ the Rademacher random variable used to decide the sign of edge (a, b). We can write

$$\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)/6 = \sum_{\text{triangles } \{i,j,k\} \text{ in } G} \xi_{ij}\xi_{jk}\xi_{ki} =: T \tag{6}$$

If $\mathrm{tr}\left(\mathbf{A}^3\right) > 0$, then G must have at least one triangle. Consider the following cases:

1. G has an odd number of triangles. In this case, since each term in the sum in (6) is either $+1$ or -1, and there is an odd number of terms in the sum, so $\mathrm{tr}\left(\bar{\mathbf{A}}^3\right) > 0$ wp 1.
2. G has an even number of triangles. First, define $T(a,b) := \sum_{\text{triangles } \{a,b,k\} \text{ that contain } \{a,b\}} \xi_{ab}\xi_{bk}\xi_{ak}$ We now subdivide this into two cases:
   (a) There exists an edge $\{a, b\}$ that is part of an odd number of triangles. We can decompose the sum in (6) as follows:
   $$\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)/6 = \underbrace{T(a,b)}_{S_1} + \underbrace{T - T(a,b)}_{S_2}$$
   Suppose there exists some realization of the random variables ξ_{ij} such that $\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)/6 = 0$. Since S_1 has odd terms, it must be non-zero. By flipping the sign of ξ_{ab}, we can flip the sign of S_1, and so $-S_1 + S_2 \neq 0$. Therefore, for every configuration the variables ξ_{ij} that result in a 0 trace, there exists an equally likely configuration that results in $\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)/6 \neq 0$. Therefore, $\mathbb{P}\left[\left|\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)\right| > 0\right] \geqslant 1/2$.
   (b) Every edge in G is a part of an even number of triangles. Let $\{a, b, c\}$ be a triangle in G. In this case, we decompose (6) as follows:
   $$\begin{aligned}\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)/6 = &\underbrace{\xi_{ab}\xi_{bc}\xi_{ac}}_{S_1} + \underbrace{T(a,b) - \xi_{ab}\xi_{bc}\xi_{ac}}_{S_2} + \underbrace{T(b,c) - \xi_{ab}\xi_{bc}\xi_{ac}}_{S_3} \\ &+ \underbrace{T - T(a,b) - T(b,c) + \xi_{ab}\xi_{bc}\xi_{ac}}_{S_4}\end{aligned}$$
   Consider all 4 possible values of the pair of random variables $\{\xi_{ab}, \xi_{bc}\}$. Since each S_i has an odd number of terms, $S_i \neq 0$ for all i. We observe that it is not possible for all 4 equally likely configurations of $\{\xi_{ab}, \xi_{bc}\}$ to result in $S_1 + S_2 + S_3 + S_4 = 0$, so at least one configuration must result in $T \neq 0$. Therefore, $\mathbb{P}\left[\left|\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)\right| > 0\right] \geqslant 1/4$.

□

To prove our main hardness result Theorem 5, we first present a hardness result for a more general class of spectral sums in the theorem below. It closely resembles Theorem 15 from [33], but with improved bounds as a result of the random signing.

Theorem 11 (Improved randomized version of Theorem 15 from [33]). *Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a function such that it can be expressed as $f(x) = \sum_{k=0}^{\infty} c_k (x-1)^k$ where $|c_k/c_3| \leqslant h^{k-3}$ for $k > 3$ and $x \in (0,2)$. Given an algorithm which takes as input a graph $G = (V,E)$ on n nodes and, in $O(n^\gamma \epsilon^{-c})$ time, outputs an estimate $X \approx_{\epsilon_1/9} \mathcal{S}_f(\mathbf{I} - \delta \bar{\mathbf{A}}_G)$ with δ and ϵ_1 satisfying $\delta = \min\left\{(\sqrt{n}\log(\alpha n))^{-1}, (10n^3 h \log(\alpha n))^{-1}\right\}$ and $\epsilon_1 = \min\left\{1, \left|c_3\delta^3/(c_0 n)\right|, \left|c_3\delta/(c_2 n^2)\right|\right\}$ for some constant $\alpha > 1$, we can produce an algorithm that solves the triangle detection problem in $O\left(n^2 + n^\gamma \epsilon_1^{-c}\right)$ time whp.*

Proof. For convenience, we write $\mathbf{A} = \mathbf{A}_G$. The proof closely follows the proof of Theorem 15 from [33], but we replace $\mathbf{A}$ with its symmetrically random signed version that we denote $\bar{\mathbf{A}}$. We present a full proof here for completeness.

First, we note that by lemma 11, we have that $\|\bar{\mathbf{A}}\|_2 \leqslant \sqrt{n}\log \alpha n$ whp. for some constant $\alpha > 1$. We define $\bar{\mathbf{B}} = \mathbf{I} - \delta\bar{\mathbf{A}}$ and consequently $\bar{\mathbf{B}}$ is PSD whp. Now, using the definition of f, we have

$$\sum_{i=1}^{n} \sigma_i(\bar{\mathbf{B}}) = \sum_{i=1}^{n} f(1 - \delta\lambda_i(\bar{\mathbf{A}})) = \sum_{i=1}^{n}\sum_{k=0}^{\infty} c_k(\delta\lambda_i(\bar{\mathbf{A}}))^k = \sum_{k=0}^{\infty} c_k\delta^k \mathrm{tr}\left(\bar{\mathbf{A}}^k\right).$$

We analyze the tail of this power series. Specifically, we have

$$\left|\sum_{k=4}^{\infty} c_k\delta^k \mathrm{tr}\left(\bar{\mathbf{A}}^k\right)\right| \leqslant |c_3|\,\delta^3 \sum_{k=4}^{\infty} \left|\mathrm{tr}\left(\bar{\mathbf{A}}^k\right)\right| \delta^{k-3}\left|\frac{c_k}{c_3}\right|. \tag{7}$$

Now, we have $\left|\mathrm{tr}\left(\bar{\mathbf{A}}^k\right)\right| \leqslant \|\bar{\mathbf{A}}\|_2^{k-2}\left\|\bar{\mathbf{A}}\right\|_F \leqslant n^{k/2+1}$ whp. and further $n^{k/2+1} \leqslant n^{3(k-3)}$ for all $k > 3$. Therefore, using the definition of δ as in the theorem we get that whp.,

$$\left|\mathrm{tr}\left(\bar{\mathbf{A}}^k\right)\right| \delta^{k-3}\left|\frac{c_k}{c_3}\right| \leqslant \frac{1}{10^{k-3}} \text{ for all } k > 3.$$

Plugging into Equation (7), we get

$$\left|\sum_{k=4}^{\infty} c_k\delta^k \mathrm{tr}\left(\bar{\mathbf{A}}^k\right)\right| \leqslant \frac{|c_3|\,\delta^3}{9}.$$

The rest of the proof is essentially identical to the steps in the proof of Theorem 15 in [33], but we reproduce them here for completeness.

Using the simple facts $\mathrm{tr}\left(\bar{\mathbf{A}}^0\right) = n$, $\mathrm{tr}\left(\bar{\mathbf{A}}\right) = 0$ and $\mathrm{tr}\left(\bar{\mathbf{A}}^2\right) \leqslant n^2$, we have

$$c_0\mathrm{tr}\left(\bar{\mathbf{A}}^0\right) + c_1\mathrm{tr}\left(\bar{\mathbf{A}}\right) + c_2\mathrm{tr}\left(\bar{\mathbf{A}}^2\right) \leqslant |c_3|\,\delta^3\left(c_0 n/(c_3\delta^3) + c_2 n^2/(c_3\delta)\right) \leqslant \frac{|c_3|\,\delta^3}{\epsilon_1}.$$

Given X, in $O(\mathrm{nnz}(\bar{\mathbf{A}}))$ time, we can compute

$$\begin{aligned} X - c_0 n - c_2\delta^2\mathrm{tr}\left(\bar{\mathbf{A}}^2\right) &= c_3\delta^3\mathrm{tr}\left(\bar{\mathbf{A}}^3\right) \pm \frac{|c_3|\,\delta^3}{9} \pm \frac{\epsilon_1}{9}\left(\frac{|c_3|\,\delta^3}{9} + c_3\delta^3\mathrm{tr}\left(\bar{\mathbf{A}}^3\right) + \frac{|c_3|\,\delta^3}{\epsilon_1}\right) \\ &= c_3\delta^3\left[\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)\left(1 \pm \frac{1}{20}\right) \pm \frac{1}{3}\right]. \end{aligned}$$

This is sufficient to detect if $\left|\mathrm{tr}\left(\bar{\mathbf{A}}\right)\right| = 0$ or if $\left|\mathrm{tr}\left(\bar{\mathbf{A}}^3\right)\right| \geqslant 1$. The final result then follows by applying Lemma 8. ☐

We now prove the main result Theorem 5.

Theorem 5. *Given a combinatorial algorithm which on input $\mathbf{B} \in \mathbb{R}^{n\times n}$ outputs a spectral sum estimate $Y \approx_\epsilon \mathcal{S}_f(\mathbf{B})$ in $O(n^\gamma\epsilon^{-c})$ time with $\gamma \geqslant 2$ for the spectral sums in Table 3, we can produce a randomized combinatorial algorithm that can detect a triangle in an n-node graph whp. in $\widetilde{O}(n^{\gamma+\alpha c})$ time, where α is a scaling that depends on properties of the function f (see Table 3 for values of α for several spectral sums.)*

Proof. We apply Theorem 11 to the specific spectral sums in Table 3.

Schatten 3-norm We have $f(x) = x^3$. Therefore, $c_k = 0$ for $k > 3$. So we apply Theorem 11 with $h = 0$ and hence $\delta = \widetilde{O}(\frac{1}{\sqrt{n}})$ and $\epsilon_1 = \widetilde{O}(\frac{1}{n^{2.5}})$.

Schatten p-norm $p \neq 1, 2$ We have $f(x) = x^p$. Using the Taylor series about 1, we have $\frac{c_k}{c_3} \leqslant p^{k-3}$ for all $k > 3$ as well as $\left|\frac{c_0}{c_3}\right| = \left|\frac{1}{p(p-1)(p-2)}\right| \leqslant \left|\frac{1}{2\min\{p,(p-1),(p-2)\}}\right|$ and similarly $\left|\frac{c_2}{c_3}\right| \leqslant \left|\frac{1}{2\min\{p,(p-1)\}}\right|$. Therefore, with $h = p$, we apply Theorem 11 with $\delta = \widetilde{O}(\frac{1}{n^3 p})$ and $\epsilon_1 = \frac{c_3\delta^3}{c_0 n} = \widetilde{O}(\frac{|\min\{p,(p-1),(p-2)\}|}{n^{10}p^3})$, which gives the result.

SVD Entropy We have $f(x) = x\log x$. For $x \in (0, 2)$, using the Taylor Series about 1 we can write $x\log x = \sum_{k=0}^{\infty} c_k(x-1)^k$ where $c_0 = 1\log(1) = 0$, $c_1 = \log(1) + 1 = 1$, and $|c_k| = \frac{(k-2)!}{k!} \leqslant 1$ for $k \geqslant 2$. So we have $c_k < c_3$ for all $k > 3$, $\frac{c_0}{c_3} = 0$ and $\frac{c_2}{c_3} = \frac{1}{3}$. So with $h = 1$, Applying Theorem 11 with $\delta = \widetilde{O}(\frac{1}{n^3})$ and $\epsilon_1 = \frac{\delta}{3n^2} = \widetilde{O}(\frac{1}{n^5})$ gives the result.

Log Determinant We have $f(x) = \log x$. For $x \in (0, 2)$, using the Taylor Series about 1 we can write $\log x = \sum_{k=0}^{\infty} c_k(x-1)^k$ where $c_0 = 0$, $|c_i| = 1/i$ for $i \geqslant 1$. Again we have $c_k < c_3$ for all $k > 3$ and $\frac{c_0}{c_3} = 0$ while $\frac{c_2}{c_3} = \frac{3}{2}$. So with $h = 1$, Applying Theorem 11 with $\delta = \widetilde{O}(\frac{1}{n^3})$ and $\epsilon_1 = \frac{\delta}{3n^2} = \widetilde{O}(\frac{1}{n^5})$ gives the result.

Trace of Exponential We have $f(x) = e^x$. Using the Taylor Series about 1 we can write $e^x = \sum_{k=0}^{\infty} \frac{e(x-1)^k}{k!}$. We have $\frac{c_0}{c_3} = 6$, $\frac{c_2}{c_3} = 3$, and $c_k < c_3$ for all $k \geqslant 3$. So with $h = 1$, Applying Theorem 11 with $\delta = \widetilde{O}(\frac{1}{n^3})$ and $\epsilon_1 = \frac{c_3\delta^3}{c_0 n} = \widetilde{O}(\frac{1}{n^{10}})$ gives the result. $\square$