

A New Approach to Estimating Effective Resistances and Counting Spanning Trees in Expander Graphs

Lawrence Li
Department of Computer Science
University of Toronto
lawrenceli@cs.toronto.edu

Sushant Sachdeva
Department of Computer Science
University of Toronto
sachdeva@cs.toronto.edu

November 4, 2022

Abstract

We demonstrate that for expander graphs, for all $\varepsilon > 0$, there exists a data structure of size $\tilde{O}(n\varepsilon^{-1})$ which can be used to return $(1 + \varepsilon)$ -approximations to effective resistances in $\tilde{O}(1)$ time per query. Short of storing all effective resistances, previous best approaches could achieve $\tilde{O}(n\varepsilon^{-2})$ size and $\tilde{O}(\varepsilon^{-2})$ time per query by storing Johnson–Lindenstrauss vectors for each vertex, or $\tilde{O}(n\varepsilon^{-1})$ size and $\tilde{O}(n\varepsilon^{-1})$ time per query by storing a spectral sketch.

Our construction is based on two key ideas: 1) ε^{-1} -sparse, ε -additive approximations to σ_u for all u , vectors similar to $DL^+ \mathbf{1}_u$, can be used to recover $(1 + \varepsilon)$ -approximations to the effective resistances, 2) In expander graphs, only $\tilde{O}(\varepsilon^{-1})$ coordinates of σ_u are larger than ε . We give an efficient construction for such a data structure in $\tilde{O}(m + n\varepsilon^{-2})$ time via random walks. This results in an algorithm on expander graphs for computing $(1 + \varepsilon)$ -approximate effective resistances for s vertex pairs that runs in $\tilde{O}(m + n\varepsilon^{-2} + s)$ time, improving over the previously best known running time of $m^{1+o(1)} + (n + s)n^{o(1)}\varepsilon^{-1.5}$ for $s = \omega(n\varepsilon^{-0.5})$.

We employ the above algorithm to compute a $(1 + \delta)$ -approximation to the number of spanning trees in an expander graph, or equivalently, approximating the (pseudo)determinant of its Laplacian in $\tilde{O}(m + n^{1.5}\delta^{-1})$ time. This improves on the previously best known result of $m^{1+o(1)} + n^{1.875+o(1)}\delta^{-1.75}$ time, and matches the best known size of determinant sparsifiers.

1 Introduction

Estimating Laplacian Determinants. Given a graph G with n vertices and m edges, what is the running-time complexity of estimating the number of spanning trees in G ? In the 19th century, the celebrated matrix-tree theorem of Kirchhoff [10] established that the number of spanning trees is equal to $1/n$ times the product of the non-trivial eigenvalues of the graph Laplacian, or equivalently the (pseudo)determinant of the Laplacian.¹ This implies an $O(n^\omega)$ time algorithm for exactly computing the number of spanning trees in G .

This result naturally extends to estimating the total weight of all spanning trees for graphs with non-negative weights w on the edges, where the weight of a tree $w(T)$ is defined as the product of the weights of the edges in the tree $\prod_{e \in T} w(e)$.

The first result to improve on the matrix multiplication time bound was by Durfee, Peebles, Peng, and Rao [6], giving an $\tilde{O}(n^2\delta^{-2})$ time² algorithm for returning a $(1 + \delta)$ -approximation to the Laplacian determinant. Chu, Gao, Peng, Sachdeva, Sawlani, and Wang [4] improved the running time bound to $m^{1+o(1)} + n^{1.875+o(1)}\delta^{-1.75}$.

The current approaches to estimating the determinant are based on the construction of determinant sparsifiers. A determinant sparsifier of a graph is a sparse (reweighted) subgraph that approximately preserves the determinant. [6] showed how to construct $(1 + \delta)$ -determinant sparsifiers for general graphs with $\tilde{O}(n^{1.5}\delta^{-1})$ edges.³ Even considering random subgraphs of the complete graph K_n , the best construction of a determinant sparsifier requires $\Omega(n^{1.5})$ edges [8]. This results in a natural lower bound of $\Omega(n^{1.5})$ for the running time of current approaches to determinant estimation. That leads to the following question:

Can we compute an $O(1)$ -approximation to the Laplacian determinant in $\tilde{O}(n^{1.5})$ time?

Estimating Effective Resistances. As noted above, the current algorithms for estimating Laplacian determinants are far from this $\Omega(n^{1.5})$ time bound. The bottleneck in the current approaches is the time required for constructing determinant sparsifiers. This, in turn, requires estimating the effective resistances of $s = \Omega(n^{1.5})$ edges within a multiplicative factor of $(1 + \varepsilon)$ for $\varepsilon = O(n^{-0.25})$. The effective resistance between $u, v \in V$ is denoted by $R(u, v)$ or $R^G(u, v)$ and is defined as the equivalent effective resistance between the vertices u, v if the graph is thought of as an electrical network with edge e having resistance $1/w_e$. We can also write $R(u, v) = (\mathbf{1}_u - \mathbf{1}_v)^\top \mathbf{L}^+(\mathbf{1}_u - \mathbf{1}_v)$. This motivates the following question:

How efficiently can we compute $(1 + \varepsilon)$ -approximations to effective resistances of s edges?

Table 1 summarizes the known algorithms for estimating effective resistances. The current best algorithm estimates s effective resistances in $m^{1+o(1)} + (n + s)n^{o(1)}\varepsilon^{-1.5}$ time [4], resulting in an $m^{1+o(1)} + n^{1.875+o(1)}$ time bound for estimating the determinant. Observe that any ε^{-1} factors on s will result in a running time worse than $O(n^{1.5})$ for estimating the determinant.

Data-Structure view. Say we're allowed to preprocess G to build a data-structure that occupies small space. After pre-processing, pairs of vertices arrive online and the data-structure must answer a $(1 + \varepsilon)$ -approximate effective resistance in G for each pair. The trivial approach is to compute and store all pairwise effective resistances using $O(n^2)$ space, and answer each query in $O(1)$ time.

The first non-trivial approach is to use Spielman and Srivastava [14] algorithm for estimating effective resistances combining Johnson-Lindenstrauss dimension reduction with fast Laplacian solvers [15], resulting in a data-structure that requires $O(n\varepsilon^{-2} \log n)$ storage⁴. The data structure can be built in $\tilde{O}(\min\{m\varepsilon^{-2}, m + n\varepsilon^{-4}\})$ time, and can answer each query efficiently in $O(\varepsilon^{-2} \log n)$ time. Another approach is to store a spectral sketch [7, 4], a sparse subgraph of G that approximately preserves effective resistances. Such a sketch can be computed in $\tilde{O}(m)$ time, and requires $\tilde{O}(n\varepsilon^{-1})$ space. However, the computation time now is much slower, requiring $\tilde{O}(n\varepsilon^{-1})$ time. There is an immediate open question:

Can we design a data-structure that requires $\tilde{O}(n\varepsilon^{-1})$ space, and can answer effective resistance queries efficiently, say $\tilde{O}(\varepsilon^{-1})$ or $O(1)$ time per query?

¹The usual determinant for the Laplacian is 0 since it has a trivial eigenvalue of 0. In this paper, we overload the notation Laplacian determinant to denote the pseudodeterminant, the product of all non-zero eigenvalues

²The $\tilde{O}(\cdot)$ notation hides poly($\log n$) factors.

³Theorem 1.1 in Durfee, Peebles, Peng, and Rao [6] states the number of edges required as $O(n^{1.5}\delta^{-2})$. The same construction with a better choice of parameters leads to determinant-sparsifiers with $O(n^{1.5}\delta^{-1})$ edges.

⁴For convenience, we are counting the complexity in terms of words of $O(\log n)$ size, so the bit complexity is higher by a factor of $O(\log n)$

Citation	Total running time	Det. Running time	Key ideas
[14]	$m\varepsilon^{-2} + s\varepsilon^{-2}$	$m\sqrt{n} + n^2$	JL + Lap. Solvers
	$m + n\varepsilon^{-4} + s\varepsilon^{-2}$	n^2	Spectral-sparsifiers + JL + Lap. Solvers
[7]	$n^2\varepsilon^{-1}$	$n^{2.25}$	Spectral-sketches + Lap. Solvers
[5]	$m + n\varepsilon^{-2} + s\varepsilon^{-2}$	n^2	Approx. Schur
[4]	$m + n\varepsilon^{-1.5} + s\varepsilon^{-1.5}$	$m + n^{1.875}$	Spectral-sketches + Approx. Schur
This paper	$m + n\varepsilon^{-2} + s$	$m + n^{1.5}$	only for expanders, additive approx. to σ_u

Table 1: Known algorithms for approximating effective resistances of s vertex pairs to within $(1 + \varepsilon)$ in a graph with m edges and n vertices, and the implied running time for estimating determinants. The running times expressed above hide $m^{o(1)}$ factors. JL refers to methods using the Johnson–Lindenstrauss lemma [9], and "Lap. Solvers" indicate using nearly-linear time Laplacian Solvers [15]. "Approx. Schur" indicates using approximate Schur complements [12].

We note that if we can build such a data-structure efficiently, it immediately implies an algorithm for estimating effective resistances. However, building such a data-structure is a harder question than estimating effective resistances.

1.1 Our Results

We make progress on all aforementioned questions on expander graphs.

THEOREM 1.1. *For a graph G with expansion $\tilde{\Omega}(1)$, for any $\varepsilon > 0$, in time $\tilde{O}(m + n\varepsilon^{-2})$, we can compute a data-structure with storage $\tilde{O}(n\varepsilon^{-1})$, that can answer $(1 + \varepsilon)$ -approximations to the effective resistance between any vertex pair with high probability in $\tilde{O}(1)$ time. This gives an algorithm that with high probability returns $(1 + \varepsilon)$ -approximate effective resistances for s vertex pairs in G in $\tilde{O}(m + n\varepsilon^{-2} + s)$ time.*

This establishes for the first time the existence of a data-structure with $\tilde{O}(n\varepsilon^{-1})$ space that can answer each query very efficiently – in $\tilde{O}(1)$ time; the previous best running time with the same amount of space was $\tilde{\Omega}(n\varepsilon^{-1})$. Note that for $\varepsilon = o(1)$, this is the first data-structure that can answer effective resistance queries in $\tilde{O}(1)$ time while only using $o(m)$ space, i.e. without essentially storing them all. For most applications, including graph sparsification and estimating Laplacian determinants, we have $s \geq \tilde{\Omega}(m)$ or $s \geq \tilde{\Omega}(n\varepsilon^{-2})$, resulting in an improvement of $\Omega(\varepsilon^{-1.5})$ in running time compared to the $m^{1+o(1)} + (n + s)n^{o(1)}\varepsilon^{-1.5}$ bound from [4].

Building on this, we give an improved algorithm for estimating Laplacian determinants for expander graphs.

THEOREM 1.2. *For graphs G with $\tilde{\Omega}(1)$ expansion, for any $\delta > 0$, there is an algorithm that returns a $(1 + \delta)$ -approximation of the Laplacian determinant in $\tilde{O}(m + n^{1.5}\delta^{-1})$ with high probability.*

This improves on the previous best running time of $m^{1+o(1)} + n^{1.875+o(1)}\delta^{-1.75}$ from [4]. Ours is the first algorithm that matches the natural $\Omega(n^{1.5})$ lower bound on the size of determinant sparsifiers. Our algorithm extends to estimating determinants of a symmetric $(1 + \alpha)$ -diagonally dominant (DD) matrices (a submatrix \mathbf{M} of any Laplacian that satisfies $\mathbf{M}_{uu} \geq (1 + \alpha) \sum_{v:v \neq u} |\mathbf{M}_{uv}|$).

THEOREM 1.3. *For any $\delta, \alpha > 0$, given a symmetric $(1 + \alpha)$ -DD matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, with m non-zeroes, there is an algorithm that returns a $(1 + \delta)$ -approximate determinant for \mathbf{M} in $\tilde{O}(m + n^{1.5}\delta^{-1}(1 + \alpha^{-3}))$ time with high probability.*

This is significant since the current approach to estimating Laplacian determinants [6, 4] is based on recursing onto two smaller matrices, each with roughly half the vertices, where one of them is 2-DD.

1.2 Technical overview Effective resistance is usually studied as an ℓ_2^2 norm of vectors. Specifically, the approach from Spielman and Srivastava [14] is based on constructing vectors \mathbf{z}_u for every vertex u such that $R(u, v) = \|\mathbf{z}_u - \mathbf{z}_v\|_2^2$ for every u, v . Applying Johnson-Lindenstrauss dimensionality reduction to these vectors gives vectors $\tilde{\mathbf{z}}_u$ in $\tilde{O}(\varepsilon^{-2})$ dimensions such that $R(u, v) = (1 \pm \varepsilon)\|\tilde{\mathbf{z}}_u - \tilde{\mathbf{z}}_v\|_2^2$. It is known that the JL dimension reduction strategy requires $\tilde{\Omega}(\varepsilon^{-2})$ dimensions [13].

Spectral sparsifiers [14, 2] and spectral sketches [7, 4] guarantee that there is a sparse graph H such that for any fixed vector $\mathbf{x} \in \mathbb{R}^V$, $\mathbf{x}^\top \mathbf{L}_H^+ \mathbf{x}$ is the same as $\mathbf{x}^\top \mathbf{L}_G^+ \mathbf{x}$ up to a $(1 \pm \varepsilon)$ factor with high probability. Equivalently, the "error" $|\mathbf{x}^\top \mathbf{L}_H^+ \mathbf{x} - \mathbf{x}^\top \mathbf{L}_G^+ \mathbf{x}|$ is bounded by $\varepsilon \mathbf{x}^\top \mathbf{L}_G^+ \mathbf{x}$, which for expander graphs is $\tilde{O}(\varepsilon \|\mathbf{x}\|_2^2)$. Picking $\mathbf{x} = \mathbf{1}_u - \mathbf{1}_v$ implies that this guarantee is sufficient for approximating effective resistances. While the notion of a spectral sketch is stronger than our goal of approximating all effective resistances, it is currently the only known approach to achieve this goal while requiring only $\tilde{O}(n\varepsilon^{-1})$ space. Furthermore, it is known that every spectral sketch requires $\tilde{\Omega}(n\varepsilon^{-1})$ space [1].

We observe that for estimating effective resistances on expanders, a much weaker error guarantee of $\varepsilon \|\mathbf{x}\|_1^2$ suffices. This is because the ℓ_1 norm and ℓ_2 norm of $\mathbf{1}_u - \mathbf{1}_v$ are within a constant factor, though for arbitrary vectors \mathbf{x} , $\|\mathbf{x}\|_1^2$ could be larger than $\|\mathbf{x}\|_2^2$ by a factor of n . Building on this observation, we prove that for any graph, ε -additive approximations to $\{\mathbf{D}\mathbf{L}^+\mathbf{\Pi}\mathbf{1}_u\}$ for all vertices u , are sufficient for recovering $(1 + O(\varepsilon))$ -approximations to effective resistances between any vertex pair, where $\mathbf{\Pi}$ is the projection orthogonal to the all ones vector.

However, the vectors $\{\mathbf{D}\mathbf{L}^+\mathbf{\Pi}\mathbf{1}_u\}$ do not have small enough ℓ_1 norm for us to obtain a sparsity guarantee with ε -additive approximations, even on expanders. We instead show that there exist vectors $\{\boldsymbol{\sigma}_u\}$ which are similar to $\{\mathbf{D}\mathbf{L}^+\mathbf{\Pi}\mathbf{1}_u\}$ for which the previous argument still holds. Furthermore, in an expander graph, for every vertex u , the vector $\boldsymbol{\sigma}_u$ can only have $\tilde{O}(\varepsilon^{-1})$ coordinates larger than ε . The proof is based on interpreting these vectors in terms of random walks starting at the vertex u , and exploiting that random walks in expanders mix in $\tilde{O}(1)$ steps. Thus, we only need $\tilde{O}(n\varepsilon^{-1})$ space to store ε -additive approximations to all vectors $\{\boldsymbol{\sigma}_u\}_{u \in V}$, and can return $(1 + \varepsilon)$ -approximations to each effective resistance query in $\tilde{O}(1)$ time.

This proof also gives an algorithm for building such a data-structure. Starting from each vertex u , we will take $\tilde{O}(\varepsilon^{-2})$ random walks, each of length $\tilde{O}(1)$. Standard concentration bounds guarantee that this is sufficient to estimate each coordinate of the vector up to an additive ε error with high probability. We preprocess the graph in $O(m)$ time to be able to sample each random walk step in $O(1)$ time. This results in a total preprocessing time of $\tilde{O}(m + n\varepsilon^{-2})$ and an algorithm for computing s pairs of effective resistances in total time $\tilde{O}(m + n\varepsilon^{-2} + s)$.

In order to $(1 + \delta)$ -approximate the Laplacian determinant for expander graphs and $(1 + \alpha)$ -DD matrices, we build on the algorithm from [6]. We modify their algorithm to ensure that every subgraph generated by the algorithm, for which we need to estimate its Laplacian determinant, remains an expander. The runtime bottleneck in our algorithm is the same as in [6], the time for estimating the effective resistance of $s = \tilde{O}(n^{1.5}\delta^{-1})$ edges up to a $(1 + \Theta(n^{-0.25}\delta^{-0.5}))$ multiplicative factor. Thus, with our modification ensuring expander graphs throughout, we can use our new algorithm for estimating effective resistances, resulting in a running time of $\tilde{O}(m + n^{1.5}\delta^{-1})$.

For a $(1 + \alpha)$ -DD matrix \mathbf{M} , by adding a new vertex and connecting it to all existing vertices, this matrix can be extended to a Laplacian. Crucially, the pseudodeterminant of the new matrix is exactly n times the determinant of \mathbf{M} . If we eliminate this new vertex in the Schur complement, we obtain a graph with expansion $\alpha/(1 + \alpha)$ (as measured by the second eigenvalue of its normalized Laplacian). However, this new graph is dense and hence cannot be written down explicitly. Instead, we exploit the fact that we can simulate random walks on this graph by representing the clique implicitly. This allows us to implement our algorithm for estimating effective resistances for expanders on the Schur complement, and hence estimate its determinant in $\tilde{O}(m + n^{1.5}\delta^{-1}(1 + \alpha^{-3}))$ time.

2 Preliminaries

We are given a graph $G(V, E)$ with n vertices and m edges and edge weights bounded by W . Let \mathbf{A} denote the weighted adjacency matrix for G , and \mathbf{D} denote the diagonal matrix of weighted degrees. Then, the Laplacian \mathbf{L} of G is a V by V matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The normalised Laplacian of G is $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. The measure of graph expansion we use is the spectral gap of the normalized Laplacian, or equivalently its second smallest eigenvalue $\nu_2 = \lambda_2(\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2})$. We say that G is an expander if $\nu_2 = \tilde{\Omega}(1)$. By Cheeger's inequality, this is equivalent to G having $\tilde{\Omega}(1)$ conductance.

Given a symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, we will express its spectral decomposition as $\mathbf{M} = \sum_{i=0}^n \lambda_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^\top$, where $\{\boldsymbol{\psi}_i\}_{i=1, \dots, n}$ are orthonormal eigenvectors with eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ respectively. We define its pseudoinverse \mathbf{M}^+ to be: $\mathbf{M}^+ = \sum_{i: \lambda_i \neq 0} \frac{1}{\lambda_i} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^\top$.

In particular, for a connected graph G , its Laplacian \mathbf{L} has kernel spanned exactly by $\mathbf{1}$, the all 1 vector. We will write $\mathbf{A} \preceq \mathbf{B}$ or $\mathbf{B} \succeq \mathbf{A}$ iff $\mathbf{B} - \mathbf{A}$ is positive semi-definite.

The effective resistance between two vertices u and v in a graph G is defined to be

$$R(u, v) = (\mathbf{1}_u - \mathbf{1}_v)^\top \mathbf{L}^+ (\mathbf{1}_u - \mathbf{1}_v).$$

where $\mathbf{1}_u$ is defined to be the vector that is 1 on the u entry, and 0 everywhere else. We will use the following fact about effective resistances, the proof of which is available in the appendix.

FACT 2.1. *The effective resistance of any edge (u, v) is lower bounded by $\frac{1}{2}(\frac{1}{d_u} + \frac{1}{d_v})$, where d_u and d_v are the degrees of u and v respectively.*

The Schur complement of a symmetric matrix $\mathbf{M} = \begin{bmatrix} \mathbf{F} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{bmatrix}$ onto some submatrix \mathbf{C} is defined to be $\mathbf{Sc}(\mathbf{M}, \mathbf{C}) = \mathbf{C} - \mathbf{B}^\top \mathbf{F}^{-1} \mathbf{B}$. In the context of Laplacians, we abuse notation to have $\mathbf{Sc}(G, V_1)$, $\mathbf{Sc}(\mathbf{L}, V_1)$, and $\mathbf{Sc}(\mathbf{L}, \mathbf{C})$ all refer to the Schur complement of \mathbf{L} onto the submatrix \mathbf{C} on the vertex support V_1 . A useful fact about Schur complements is that the Schur complement of a Laplacian is also a Laplacian. When convenient, we may use $\mathbf{Sc}(G, V_1)$ to refer to the graph of the Schur complement from G onto V_1 . Another important fact is that $\mathbf{Sc}(G, V_1)$ preserves the effective resistances of the original graph for pairs of vertices in V_1 .

FACT 2.2. *Consider a graph G , and some arbitrary subset of vertices V_1 . For all $u, v \in V_1$, we have that $R^G(u, v) = R^{\mathbf{Sc}(G, V_1)}(u, v)$.*

A $(1 + \alpha)$ -diagonally dominant matrix \mathbf{M} is a symmetric matrix M with non-positive off diagonal entries satisfying $M_{uu} \geq (1 + \alpha) \sum_{v: v \neq u} |M_{uv}|$. Given such a matrix \mathbf{M} of size n , we can complete it into a Laplacian \mathbf{L}_M by adding only a single column and row. Call the newly added vertex x . We have that:

$$\begin{aligned} (\mathbf{L}_M)_{ux} &= (\mathbf{L}_M)_{xu} = -M_{uu} + \sum_{v: v \neq u} M_{uv}, \\ (\mathbf{L}_M)_{xx} &= -\sum_u (\mathbf{L}_M)_{xu}. \end{aligned}$$

We also refer to a subset V_1 of a graph G as being $(1 + \alpha)$ -diagonally dominant if for each $u \in V_1$, we have:

$$\sum_{v \sim u, v \notin V_1} w_{uv} \geq \alpha \sum_{v \sim u, v \in V_1} w_{uv}.$$

3 Effective Resistances

In this section, we demonstrate that for expanders, a sparse data structure exists, and can be efficiently computed, from which we can query the effective resistances between any two vertices in $\tilde{O}(1)$ time. The main result of this section is a more precise statement of Theorem 1.1 as follows:

THEOREM 3.1. *Given a graph G such that the spectral gap of its normalised Laplacian is ν_2 , there exists an algorithm EFFECTIVERESISTANCESKETCH that computes vectors $\{\tilde{\sigma}_u\}$, in $O(m + n\varepsilon^{-2}\nu_2^{-3} \log n \log^3 \frac{nW}{\varepsilon})$ total time, such that:*

1. *The vectors $\{\tilde{\sigma}_u\}$ are $O(\nu_2^{-1}\varepsilon^{-1} \log nW)$ -sparse.*
2. *For any vertex pair (u, v) EFFECTIVERESISTANCEQUERY can return an estimate of $R(u, v)$ in $\tilde{O}(1)$ time⁵ by querying the vectors $\tilde{\sigma}_u$ and $\tilde{\sigma}_v$. This estimate is a $(1 + \varepsilon)$ -approximation to $R(u, v)$ with high probability.*

3.1 Effective Resistances on Expanders We first begin by proving the existence of these vectors:

LEMMA 3.1. *Suppose G is a graph such that the spectral gap of its normalised Laplacian is ν_2 . There exists vectors $\{\tilde{\sigma}_u\}$ such that the following hold:*

1. *The vectors $\{\tilde{\sigma}_u\}$ are $O(\nu_2^{-1}\varepsilon^{-1} \log nW)$ -sparse.*

⁵The query time is equal to the time required to lookup a constant number of entries. A constant amortized time can be achieved with a good hash function, or $O(\log n)$ time can be achieved through a sorted data set.

2. The effective resistance of an edge (u, v) can be $(1 + \varepsilon)$ -approximated by the vectors $\tilde{\sigma}_u$ and $\tilde{\sigma}_v$ in $\tilde{O}(1)$ time.

For some intuition as to why we consider the vectors that we do, we first consider the set of vectors $DL^+\mathbf{1}_u, u \in V$. It is easy to see that we can query $(1 + \varepsilon)$ -approximations to the effective resistances from ε -additive approximations to these vectors, given the lower bound in Fact 2.1.

If each of these vectors were to have small ℓ_1 norm, on the order of $\tilde{O}(\nu_2^{-1})$, then we would be done, since we could just round off these vectors to the nearest ε to obtain the sparse vectors. However, it is not in general true for expanders that these vectors have small enough ℓ_1 norm, and in fact, for constant expanders, the ℓ_1 norm of $DL^+\mathbf{1}_u$ can scale linearly with n .

Instead, we consider another set of vectors, constructed through writing L^+ as a power series expansion. We first begin with the power series expansion on L^+ :

LEMMA 3.2. Given a graph G , with Laplacian $L = D - A$, projection matrix $\Pi = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ and $\pi = \frac{D\mathbf{1}}{\mathbf{1}^\top D\mathbf{1}}$, the stationary distribution of the random walk on G , we have:

$$L^+(\mathbf{1}_u - \pi) = \frac{1}{2}\Pi D^{-1} \sum_{t=0}^{\infty} \left(\left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t \mathbf{1}_u - \pi \right).$$

Proof. Let $X = \frac{1}{2}I + \frac{1}{2}AD^{-1}$. We have:

$$\frac{1}{2}(D - A)D^{-1} \sum_{t=0}^n \left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t = (I - X) \sum_{t=0}^n X^t = I - X^{n+1}.$$

Consider the action of this matrix on $\mathbf{1}_u - \pi$. As $n \rightarrow \infty$, since X is the lazy random walk matrix, repeated calls of it on $\mathbf{1}_u - \pi$, a vector with entrywise sum 0, sends it to $\mathbf{0}$. As such, we have:

$$\frac{1}{2}(D - A)D^{-1} \sum_{t=0}^{\infty} \left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t (\mathbf{1}_u - \pi) = (I - X) \sum_{t=0}^{\infty} X^t (\mathbf{1}_u - \pi) = \mathbf{1}_u - \pi.$$

Thus, $\mathbf{x} = D^{-1} \sum_{t=0}^{\infty} \left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t (\mathbf{1}_u - \pi)$ is a solution to the equation $(\frac{1}{2}L)\mathbf{x} = \mathbf{1}_u - \pi$. We know that $(\frac{1}{2}L)^+(\mathbf{1}_u - \pi)$ is the unique solution to this equation that is perpendicular to $\mathbf{1}$, so we have:

$$\begin{aligned} 2L^+(\mathbf{1}_u - \pi) &= \left(\frac{1}{2}L \right)^+(\mathbf{1}_u - \pi) = \Pi D^{-1} \sum_{t=0}^{\infty} \left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t (\mathbf{1}_u - \pi) \\ L^+(\mathbf{1}_u - \pi) &= \frac{1}{2}\Pi D^{-1} \sum_{t=0}^{\infty} \left(\left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t \mathbf{1}_u - \pi \right). \end{aligned}$$

as desired. \square

Instead of considering $DL^+\mathbf{1}_u$, we instead consider $\sigma_u = \frac{1}{2} \sum_{t=0}^{\infty} \left(\left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t \mathbf{1}_u - \pi \right)$. This is similar in spirit to $DL^+\mathbf{1}_u$, since $DL^+(\mathbf{1}_u - \pi) = D\Pi D^{-1}\sigma_u$.

However, compared to $DL^+\mathbf{1}_u$, it can be shown that σ_u each have small ℓ_1 norm, and can be used to recover $(1 + \varepsilon)$ -approximations to the effective resistances. We first begin with a by showing that ε -additive approximations to this set of vectors suffices.

LEMMA 3.3. Given an $\frac{1}{4}\varepsilon$ -additive approximation to the vectors $\{\sigma_u = \frac{1}{2} \sum_{t=0}^{\infty} \left(\left(\frac{1}{2}I + \frac{1}{2}AD^{-1} \right)^t \mathbf{1}_u - \pi \right) | u \in V\}$, we can obtain $(1 + \varepsilon)$ -approximations to the effective resistances of the graph G corresponding to the Laplacian L .

Proof. Suppose we have ε -additive approximations $\{\tilde{\sigma}_u\}$ to $\{\sigma_u\}$. For a given edge (u, v) , we have that:

$$\begin{aligned} R(u, v) &= (\mathbf{1}_u - \mathbf{1}_v)^\top L^+(\mathbf{1}_u - \mathbf{1}_v) \\ &= (L^+\mathbf{1}_u)_u - (L^+\mathbf{1}_v)_u - (L^+\mathbf{1}_u)_v + (L^+\mathbf{1}_v)_v. \end{aligned}$$

Since $\mathbf{L}^+(\mathbf{1}_u - \boldsymbol{\pi}) = \mathbf{\Pi D}^{-1} \boldsymbol{\sigma}_u$ from Lemma 3.2 and since the projection to the all 1s vector does not change the difference between any two elements, we have:

$$(\mathbf{L}^+(\mathbf{1}_u - \boldsymbol{\pi}))_u - (\mathbf{L}^+(\mathbf{1}_u - \boldsymbol{\pi}))_v = (\mathbf{\Pi D}^{-1} \boldsymbol{\sigma}_u)_u - (\mathbf{\Pi D}^{-1} \boldsymbol{\sigma}_u)_v = (\mathbf{D}^{-1} \boldsymbol{\sigma}_u)_u - (\mathbf{D}^{-1} \boldsymbol{\sigma}_u)_v.$$

Next, since they are both offset by the same value $(\mathbf{L}^+ \boldsymbol{\pi})_u$, we have:

$$(\mathbf{L}^+ \mathbf{1}_u)_u - (\mathbf{L}^+ \mathbf{1}_v)_u = (\mathbf{L}^+(\mathbf{1}_u - \boldsymbol{\pi}))_u - (\mathbf{L}^+(\mathbf{1}_v - \boldsymbol{\pi}))_u,$$

which gives us:

$$\begin{aligned} R(u, v) &= (\mathbf{L}^+ \mathbf{1}_u)_u - (\mathbf{L}^+ \mathbf{1}_u)_v - (\mathbf{L}^+ \mathbf{1}_v)_u + (\mathbf{L}^+ \mathbf{1}_v)_v \\ &= (\mathbf{L}^+(\mathbf{1}_u - \boldsymbol{\pi}))_u - (\mathbf{L}^+(\mathbf{1}_u - \boldsymbol{\pi}))_v - (\mathbf{L}^+(\mathbf{1}_v - \boldsymbol{\pi}))_u + (\mathbf{L}^+(\mathbf{1}_v - \boldsymbol{\pi}))_v \\ &= (\mathbf{D}^{-1} \boldsymbol{\sigma}_u)_u - (\mathbf{D}^{-1} \boldsymbol{\sigma}_u)_v - (\mathbf{D}^{-1} \boldsymbol{\sigma}_v)_u + (\mathbf{D}^{-1} \boldsymbol{\sigma}_v)_v. \end{aligned}$$

Hence, we can approximate the effective resistance with:

$$\frac{1}{d_u} (\tilde{\boldsymbol{\sigma}}_u)_u - \frac{1}{d_v} (\tilde{\boldsymbol{\sigma}}_u)_v + \frac{1}{d_v} (\tilde{\boldsymbol{\sigma}}_v)_v - \frac{1}{d_u} (\tilde{\boldsymbol{\sigma}}_v)_u.$$

The total error is upper bounded by $\frac{1}{2}(\frac{1}{d_u} + \frac{1}{d_v})\varepsilon$. Since by Fact 2.1, the effective resistance $R(u, v)$ is lower bounded by $\frac{1}{2}(\frac{1}{d_u} + \frac{1}{d_v})$, this gives us a $(1 + \varepsilon)$ -approximation to the effective resistances. \square

Next, we show that these vectors do indeed have small ℓ_1 norm.

LEMMA 3.4. *Given a graph G so that the spectral gap of its normalised Laplacian is ν_2 , the vectors $\{\boldsymbol{\sigma}_u = \frac{1}{2} \sum_{t=0}^{\infty} ((\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1})^t \mathbf{1}_u - \boldsymbol{\pi}) \mid u \in V\}$ each have ℓ_1 norm bounded by $O(\nu_2^{-1} \log nW)$.*

Proof. First, notice that this summation converges. $\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1}$ is the lazy random walk matrix. As such, when applied to $\mathbf{1}_u$, a vector with sum 1, this term goes to $\boldsymbol{\pi}$, the stationary distribution. We explicitly bound the tail terms of this summation, truncating it at t_0 . We have:

$$\sum_{t=0}^{\infty} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right) = \sum_{t=0}^{t_0-1} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right) + \sum_{t=t_0}^{\infty} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right).$$

We first bound the total ℓ_1 norm of the second term. We begin with a well known result on the convergence of random walks, the proof of which is available in the appendix.

LEMMA 3.5. *Given a graph G , and an initial distribution \mathbf{p} , the ℓ_1 norm difference between the distribution of the lazy random walk after t steps and the stationary distribution is bounded by:*

$$\left\| \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1} \right)^t \mathbf{p} - \boldsymbol{\pi} \right\|_1 \leq e^{-t\nu_2/2} \frac{nd_{\max}}{d_{\min}}.$$

where d_{\max} and d_{\min} are the largest and smallest weighted degrees respectively.

This bounds the ℓ_1 norm of the second term:

$$\begin{aligned} \left\| \sum_{t=t_0}^{\infty} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right) \right\|_1 &\leq \sum_{t=t_0}^{\infty} \left\| \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right\|_1 \\ &\leq \frac{2nd_{\max}}{d_{\min}} \sum_{t=t_0}^{\infty} e^{-t\nu_2/2} \\ &= \frac{2nd_{\max}}{d_{\min}} \frac{e^{-\frac{\nu_2 t_0}{2}}}{1 - e^{-\frac{\nu_2}{2}}}. \end{aligned}$$

We have that $\frac{d_{max}}{d_{min}} \leq W$, and $(1 - e^{-\frac{\nu_2}{2}}) \geq \nu_2/4$. We bound ν_2 as follows. The conductance of any graph is minimally $\frac{1}{n^2W}$. Cheeger's inequality then tells us that $\nu_2 \geq (\frac{1}{n^2W})^2/2$. Hence, setting $t_0 = O(\nu_2^{-1} \log nW)$ bounds the error by a constant. The first term can also easily be bounded by $2t_0$:

$$\begin{aligned} \left\| \sum_{t=0}^{t_0-1} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right) \right\|_1 &\leq \sum_{t=0}^{t_0-1} \left\| \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u \right\|_1 + \|\boldsymbol{\pi}\|_1 \\ &= 2t_0, \end{aligned}$$

since the random walk matrix preserves ℓ_1 norms for strictly positive vectors. Combining bounds the ℓ_1 norm of $\sum_{t=0}^{\infty} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right)$ by $O(\nu_2^{-1} \log nW)$ as desired. \square

We now combine Lemma 3.3 and Lemma 3.4 to prove Lemma 3.1:

Proof. [Proof of Lemma 3.1]

We have from Lemma 3.4 that the ℓ_1 norm of $\boldsymbol{\sigma}_u = \sum_{t=t_0}^{\infty} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right)$ is bounded by $O(\nu_2^{-1} \log nW)$. As such, there can be at most $O(\nu_2^{-1} \varepsilon^{-1} \log nW)$ entries in this vector that have absolute value greater than $\varepsilon/4$. Consider the $\varepsilon/4$ -additive approximation to this vector $\tilde{\boldsymbol{\sigma}}_u$ where each entry that has absolute value less than $\varepsilon/4$ is 0. This vector is $O(\nu_2^{-1} \varepsilon^{-1} \log nW)$ sparse.

By Lemma 3.3, we can use these $\varepsilon/4$ -additive approximations to the vectors $\boldsymbol{\sigma}_u$ to obtain $(1 + \varepsilon)$ -approximations to the effective resistances in $\tilde{O}(1)$ time for each query. \square

Finally, we demonstrate that these vectors can be efficiently calculated, proving Theorem 3.1. From the above discussion, we've established that to obtain $(1 + \varepsilon)$ -approximations to the effective resistances, we simply have to produce the vectors $\{\tilde{\boldsymbol{\sigma}}_u\}_{u \in V}$. We can view the algorithm as follows. The random walk starting at any position mixes quickly, and quickly there is no significant difference between its probability distribution and the stationary distribution. Most coordinates approach the stationary distribution quickly enough that even summing across the first $\tilde{O}(\nu_2^{-1})$ steps their contributions are small and can be discarded, while the remaining $\tilde{O}(\varepsilon^{-1})$ coordinates have significant sums from which we can extract the effective resistances.

Proof. [Proof of Theorem 3.1] From our proof of Lemma 3.4, we have that the power series, truncated at $t_0 = O(\nu_2^{-1} \log \frac{nW}{\varepsilon})$ gives us at most $\varepsilon/8$ additive error in each coordinate. As such, we only have to produce a $\varepsilon/8$ -approximation of:

$$\sum_{t=0}^{t_0-1} \left(\left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u - \boldsymbol{\pi} \right).$$

To do so, we simply perform lazy random walks. We calculate an $\varepsilon/8$ -additive approximation to $\left(\sum_{t=0}^{t_0-1} \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u \right)_v$ for all $v \in V$ by performing $s = O(\varepsilon^{-2} t_0 \log n)$ random walks for each length l from 0 to t_0 starting at u . Fix some starting vertex u , and some end vertex v , and let X_{li}^{uv} be the random variable that the i -th random walk of length l starting at u ends at v . Let $S_{uv} = \sum_{i=1}^s \sum_{l=0}^{t_0-1} X_{li}^{uv}$.

By a Hoeffding bound, we have that:

$$\Pr(|S_{uv} - \mathbb{E}[S_{uv}]| \geq s\varepsilon/8) \leq 2e^{-\frac{2(s\varepsilon/8)^2}{st_0}}.$$

This gives us that our approximation to $\left(\sum_{t=0}^{t_0-1} \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{1}_u \right)_v$, namely $\frac{1}{s} S_{uv}$ is an $\varepsilon/8$ -additive approximation with high probability. Now union bounding over all the possible vertices gives us high probability guarantees on all the errors being $\varepsilon/8$ -additive approximations.

For each vertex, there are a total of $s = O(\varepsilon^{-2} t_0 \log n)$ random walks being performed for each length l from 0 to t_0 . We note here that there is a procedure UNSORTEDPROPORTIONALSAMPLING[16, 3] with $O(m)$ total preprocessing time, from which we can query random walk edges in $O(1)$ time each. A simpler approach using a balanced binary search tree would also suffice, but with $O(\log n)$ overhead. This efficient sampling method allows us to run these random walks in total time:

$$O(m + nst_0^2) = O(m + n\varepsilon^{-2} \nu_2^{-3} \log n \log^3 \frac{nW}{\varepsilon}).$$

Algorithm 1 EFFECTIVERESISTANCESKETCH(G, ε, ν_2)

Input: Graph G , with spectral gap of its normalised Laplacian being ν_2

Output $O(\nu_2^{-1}\varepsilon^{-1}\log nW)$ -sparse vectors $\{\sigma_u\}$ from which the effective resistances can be queried in $O(1)$ time.

$S_{\frac{1}{n}} \mathbf{1} \leftarrow \mathbf{0}$

for $u \in V$ **do**

$S_u \leftarrow \mathbf{0}$

for $i = 1$ to $s = O(\varepsilon^{-2}\nu_2^{-1}\log n \log \frac{nW}{\varepsilon})$ **do**

for $l = 0$ to $t_0 - 1 = O(\nu_2^{-1}\log \frac{nW}{\varepsilon})$ **do**

 Perform a length l random walk starting from u , ending at vertex v .

$(S_u)_v \leftarrow (S_u)_v + \frac{1}{s}$

for $u \in V$ **do**

$\tilde{\sigma}_u \leftarrow S_u - t_0\pi_u$

 For each entry of $\tilde{\sigma}_u$, if it smaller than ε in absolute value, set it to 0.

return $\{\tilde{\sigma}_u\}_{u \in V}$ as sparse vectors

Algorithm 2 EFFECTIVERESISTANCEQUERY($\{\tilde{\sigma}_u\}, (x, y)$)

Input: The output of EFFECTIVERESISTANCESKETCH, $\{\tilde{\sigma}_u\}$, and two vertices to be queried (x, y)

Output $(1 + \varepsilon)$ -approximation of the effective resistance between x and y

return $\frac{1}{d_u}(\tilde{\sigma}_u)_u - \frac{1}{d_v}(\tilde{\sigma}_u)_v + \frac{1}{d_v}(\tilde{\sigma}_v)_v - \frac{1}{d_u}(\tilde{\sigma}_v)_u$.

Now, subtracting $t_0\pi$ from the vector yields a $\varepsilon/8$ -additive approximation to $\sum_{t=0}^{t_0-1} ((\frac{1}{2}I + \frac{1}{2}AD^{-1})^t \mathbf{1}_u - \pi)$ as desired.

The full algorithm to sketch these vectors can be seen in Algorithm 1, and the query algorithm in Algorithm 2.

□

3.2 Effective Resistances on $(1 + \alpha)$ -DD Matrices We prove an analogous statement for $(1 + \alpha)$ -DD Matrices.

Given a 2-DD matrix M , we look at its completion into a Laplacian L_M representing graph G_M with new added vertex x . On this new graph G_M , we show that the effective resistances of every edge not involving x can be calculated.

LEMMA 3.6. *Suppose M is a $(1 + \alpha)$ -DD matrix, and let L_M represent its completion into a Laplacian, and G_M be the represented graph with new vertex x . There is an algorithm that builds a data-structure in $O(m + n\varepsilon^{-2}(1 + \alpha^{-3})\log n \log^3 \frac{nW}{\varepsilon})$ time, where m is the number of edges in G_M . This data-structure allows us to query for the effective resistances between any two original vertices in G_M (not x) in $\tilde{O}(1)$ time, and with high probability, it returns a $(1 + \varepsilon)$ -approximation.*

To calculate the effective resistances between vertices in M , we can calculate the effective resistances in the Schur complement of G_M onto $V \setminus \{x\}$. Since we are removing a single vertex x , we are adding a weighted clique back onto the graph. We first show that the Schur complement produced is a ν_2 expander.

Let $G_M[V \setminus \{x\}]$ denote the subgraph of G_M restricted to the vertices $V \setminus \{x\}$. Let the degrees of each vertex u in $G_M[V \setminus \{x\}]$ be d_u , and the weight of the edge (u, x) be d'_u . Since M is an $(1 + \alpha)$ -DD matrix, we have that $d'_u \geq \alpha d_u$. Let $d_x = \sum_u d'_u$ be the total degree of the new vertex x . It is well known⁶ that the weighted clique added by the Schur complement has weights $\frac{d'_u d'_v}{d_x}$ for the edge (u, v) .

Let D' be the diagonal matrix with $D'_{uu} = d'_u$. Consider the Laplacian on just the weighted clique, normalised by the degrees D' , $D'^{-1/2}(\mathcal{S}c(G_M, V \setminus \{x\}) - L_{G_M[V \setminus \{x\}]})D'^{-1/2}$. We have that if λ is an eigenvalue of this

⁶A proof is available in the preliminaries section of [12]

matrix, with eigenvector ψ , that for each u :

$$\begin{aligned} \frac{1}{\sqrt{d'_u}} d'_u \frac{1}{\sqrt{d'_u}} \psi_u - \frac{1}{\sqrt{d'_u}} \sum_v \frac{d'_u d'_v}{d_x} \frac{1}{\sqrt{d'_v}} \psi_v &= \lambda \psi_u, \\ - \sum_v \frac{d'_v}{d_x} \frac{1}{\sqrt{d'_v}} \psi_v &= (\lambda - 1) \frac{1}{\sqrt{d'_u}} \psi_u. \end{aligned}$$

In particular, we have that $(\lambda - 1) \frac{1}{\sqrt{d'_u}} \psi_u = (\lambda - 1) \frac{1}{\sqrt{d'_v}} \psi_v$ for any two u, v . This gives us an eigenvector $\mathbf{D}'^{1/2} \mathbf{1}$, with eigenvalue 0, and any other vectors perpendicular to $\mathbf{D}'^{1/2} \mathbf{1}$ having eigenvalue 1. As such, we have:

$$\mathbf{D}'^{-1/2} (\mathbf{Sc}(G_M, V \setminus \{x\}) - \mathbf{L}_{G_M[V \setminus \{x\}]}) \mathbf{D}'^{-1/2} \succeq \mathbf{I} - \frac{1}{\mathbf{1}^\top \mathbf{D}' \mathbf{1}} \mathbf{D}'^{1/2} \mathbf{1} (\mathbf{D}'^{1/2} \mathbf{1})^\top.$$

Let \mathbf{D} be the diagonal with the degrees in $\mathbf{Sc}(G_M, V \setminus \{x\})$, $\mathbf{D}_u = d_u + d'_u - \frac{(d'_u)^2}{d_x}$. Since $d'_u \geq \alpha d_u$, we have that each diagonal entry of $\mathbf{D}' \mathbf{D}^{-1}$ is lower bounded by $d'_u / (d_u + d'_u) \geq \alpha / (1 + \alpha)$. We now have:

$$\begin{aligned} (\mathbf{Sc}(G_M, V \setminus \{x\}) - \mathbf{L}_{G_M[V \setminus \{x\}]}) &\succeq \mathbf{D}' - \frac{1}{\mathbf{1}^\top \mathbf{D}' \mathbf{1}} \mathbf{D}' \mathbf{1} (\mathbf{D}' \mathbf{1})^\top, \\ \mathbf{D}^{-1/2} (\mathbf{Sc}(G_M, V \setminus \{x\}) - \mathbf{L}_{G_M[V \setminus \{x\}]}) \mathbf{D}^{-1/2} &\succeq \mathbf{D}' \mathbf{D}^{-1} - \mathbf{D}^{-1/2} \frac{1}{\mathbf{1}^\top \mathbf{D}' \mathbf{1}} \mathbf{D}' \mathbf{1} (\mathbf{D}' \mathbf{1})^\top \mathbf{D}^{-1/2}. \end{aligned}$$

In particular, over any subspace perpendicular to $\mathbf{D}^{-1/2} \mathbf{D}' \mathbf{1}$, the eigenvalue of the weighted clique is at least $\alpha / (1 + \alpha)$. As such, since the Laplacian of $G_M[V \setminus \{x\}]$ is positive semi definite, the second smallest normalised eigenvalue of the Schur complement, $\lambda_2(\mathbf{D}^{-1/2} (\mathbf{Sc}(G_M, V \setminus \{x\})) \mathbf{D}^{-1/2})$ is at least $\alpha / (1 + \alpha)$.

We now apply the same technique as in Theorem 3.1, additionally noting that since the Schur complement adds a clique to the original edges, we cannot explicitly write down the whole graph to perform random walks. We modify the random walk process by performing the random walks implicitly.

As a preprocessing step, we first calculate the total weights of each of the edges going to x . Let $d''_u = d'_u - \frac{(d'_u)^2}{d_x}$ be the part of the degree contributed by the Schur complement. We note that $d_u + d''_u$ is the new degree of the vertex u . While performing the random walk, at any vertex u , with probability $\frac{d_u}{d_u + d''_u}$, we perform a random walk step on the $O(m)$ -sparse original graph $G_M[V \setminus \{x\}]$ using UNSORTEDPROPORTIONALSAMPLING. With the remaining probability $\frac{d''_u}{d_u + d''_u}$ we perform a random walk through the weighted clique added by the Schur complement. This is easily achieved by sampling an outgoing edge proportional to the degrees d'_v .

Now, since Schur complements preserve effective resistances, this gives us the effective resistance of all edges not involving x .

4 Approximate Determinants on Expanders

Being able to more efficiently ε -approximate the effective resistances allows us to more efficiently approximate spanning tree counts on expanders. We prove the following more precise version of Theorem 1.2 in this section:

THEOREM 4.1. *Given a graph G such that the spectral gap of its normalised Laplacian is ν_2 , we can calculate a $(1 + \delta)$ -approximation to the number of spanning trees of G in $\tilde{O}(m + n^{1.5} \delta^{-1} \nu_2^{-3})$ time with high probability.*

Proof. We follow the strategy as in [6]. We first begin with a description of the overall strategy as in [6].

We begin with a graph G . The determinant of a Laplacian is always 0, since it has a kernel $\mathbf{1}$. For the sake of convenience, we abuse notation and refer to $\det_+(\mathbf{L}_G)$, the determinant of the Laplacian \mathbf{L}_G with one of its rows and columns removed when talking about the determinant of a Laplacian. To approximate the determinant of \mathbf{L}_G , we find a $(1 + \alpha)$ -DD subset of vertices V_2 using Lemma 3.5 of [11], and $V_1 = V \setminus V_2$ and recursively calculate the following:

$$\det_+(\mathbf{L}_G) = \det(\mathbf{L}_{[v_2, v_2]}) \cdot \det_+(\mathbf{Sc}(\mathbf{L}, V_1)).$$

This decomposes into two parts, the determinant of a $(1 + \alpha)$ -DD matrix, and $\det_+(\mathbf{Sc}(\mathbf{L}, V_1))$. The former of these two terms is the submatrix of a Laplacian, and we can calculate it recursively by adding a new row and column, completing it into a Laplacian.

The latter term is the determinant of a Schur complement. The Schur complement is also a Laplacian, so ideally we would like to simply recurse on this half as well. However, taking Schur complements can result in the number of edges blowing up to $\Theta(n^2)$, so an explicit construction of the Schur complement would not be fast enough. The authors of [6] get around this by implicitly constructing a determinant sparsifier of the Schur complement.

In particular, they demonstrate that a $(1+\delta)$ -determinant sparsifier of a graph can be constructed by sampling some s edges of G , proportional to their leverage scores with ε multiplicative error, and then reweighting each edge by a factor of $\exp(\frac{n^2}{2(n-1)s})$. For the guarantees to hold, s and ε are picked so that they satisfy $\frac{n^2\varepsilon^2}{s}, \frac{n^3}{s^2} \leq \delta^2$. For the best time complexity, we pick $s = n^{1.5}\delta^{-1}, \varepsilon = n^{-0.25}\delta^{0.5}$.

In fact, the bottleneck in this algorithm is precisely the time required to sample these $n^{1.5}\varepsilon^{-1}$ $(1+\varepsilon)$ -approximate effective resistances, for $\varepsilon = n^{-0.25}\delta^{-0.5}$. Chu, Gao, Peng, Sachdeva, Sawlani, and Wang [4] rewrites the algorithm in [6], showing the following lemma:

LEMMA 4.1. ([6, 4]) *Let $T(m, n, s, \varepsilon)$ be the time required to find s $(1+\varepsilon)$ -approximations to the effective resistances of some query edges in a graph G with m edges and n vertices. There is an algorithm BETTERSCHURSPARSE that takes a 2-DD subset of vertices V_2 , and constructs a $(1+\delta)$ -determinant sparsifier of $\mathbf{Sc}(G, V_1)$ with $\tilde{O}(n^{1.5}\delta^{-1})$ edges in time:*

$$\tilde{O}(m + T(m, n^{1.5}\delta^{-1}, n^{-0.25}\delta^{-0.5})),$$

and that in fact the determinant approximation algorithm DETAPPROX has the same time complexity.

We adapt their algorithm to our effective resistance sampler that only works on $(1+\alpha)$ -DD matrices or expanders. For our approach to go through, we require that taking the Schur complement preserves expansion so that our recursion holds. We first begin by proving this lemma.

LEMMA 4.2. *Given a graph G with Laplacian \mathbf{L} such that the second smallest eigenvalue of its normalised laplacian is ν_2 , for any set C of vertices, the Schur complement $\mathbf{S} = \mathbf{Sc}(G, C)$ of G from $F = V \setminus C$ onto C , normalised to the original degrees, also has second smallest eigenvalue at least ν_2 .*

Proof. We adopt the strategy as in the proof of the eigenvalue interlacing theorem. Let $\mathbf{L} = \begin{bmatrix} \mathbf{F} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{bmatrix}$, and

$\mathbf{D}_F^{-1/2}$ and $\mathbf{D}_C^{-1/2}$ be the degrees on F and C respectively. We have:

$$(4.1) \quad \begin{bmatrix} \mathbf{D}_F^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_C^{-1/2} \end{bmatrix} \begin{bmatrix} \mathbf{F} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{D}_F^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_C^{-1/2} \end{bmatrix}$$

$$(4.2) \quad = \begin{bmatrix} \mathbf{D}_F^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_C^{-1/2} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{B}^\top \mathbf{F}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{F}^{-1} \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_F^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_C^{-1/2} \end{bmatrix}.$$

Consider some vector $\mathbf{v} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$.

We have that:

$$\begin{bmatrix} \mathbf{I} & \mathbf{F}^{-1} \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_F^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_C^{-1/2} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_F^{-1/2} \mathbf{x} + \mathbf{F}^{-1} \mathbf{B} \mathbf{D}_C^{-1/2} \mathbf{y} \\ \mathbf{D}_C^{-1/2} \mathbf{y} \end{bmatrix}.$$

Now for any \mathbf{y} , consider its extension into the whole space $\mathbf{v}(\mathbf{y}) = \begin{bmatrix} -\mathbf{D}_F^{-1/2} \mathbf{F}^{-1} \mathbf{B} \mathbf{D}_C^{-1/2} \mathbf{y} \\ \mathbf{y} \end{bmatrix}$. Notice that $\mathbf{F}^{-1}, \mathbf{D}_C^{-1/2}$ and $\mathbf{D}_F^{-1/2}$ are all well defined, since \mathbf{F} is positive definite, while \mathbf{D}_C and \mathbf{D}_F are diagonal and non-zero. We also have that $\begin{bmatrix} \mathbf{I} & \mathbf{F}^{-1} \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_F^{-1/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_C^{-1/2} \end{bmatrix} \mathbf{v}(\mathbf{y}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_C^{-1/2} \mathbf{y} \end{bmatrix}$, which combined with equation (2) gives us that $\mathbf{v}(\mathbf{y})^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{v}(\mathbf{y}) = \mathbf{y}^\top \mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2} \mathbf{y}$. This gives us a bijection from the quadratic form of the normalised Laplacian $\mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$ to the quadratic form of the Schur complement $\mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2}$, normalised to the old degrees.

Now let \mathbf{v}_1 and \mathbf{v}_2 be the smallest two eigenvectors of $\mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2}$, and let $W = \text{span}\{\mathbf{v}_1, \mathbf{v}_2\}$. We have:

$$\lambda_2(\mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2}) = \max_{\mathbf{y} \in W} \frac{\mathbf{y}^\top \mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}}$$

$$\begin{aligned}
&= \max_{\mathbf{y} \in W} \frac{\mathbf{v}(\mathbf{y})^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{v}(\mathbf{y})}{\mathbf{y}^\top \mathbf{y}} \\
&\geq \max_{\mathbf{y} \in W} \frac{\mathbf{v}(\mathbf{y})^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{v}(\mathbf{y})}{\mathbf{v}(\mathbf{y})^\top \mathbf{v}(\mathbf{y})}.
\end{aligned}$$

But if \mathbf{y} is in the two dimensional subspace W , $\mathbf{v}(\mathbf{y})$ also lies in some two dimensional vector subspace $W' = \mathbf{v}(W)$. To see this, notice that W' is indeed a vector subspace since $\mathbf{v}(\mathbf{x} + \mathbf{y}) = \mathbf{v}(\mathbf{x}) + \mathbf{v}(\mathbf{y})$. Also, W' is exactly 2-dimensional, since it can be spanned by $\mathbf{v}(\mathbf{w}_1), \mathbf{v}(\mathbf{w}_2)$ and $\mathbf{v}(\mathbf{w}_1) \neq c\mathbf{v}(\mathbf{w}_2)$, for any basis $\mathbf{w}_1, \mathbf{w}_2$ of W . Hence, we have that:

$$\begin{aligned}
\lambda_2(\mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2}) &\geq \max_{\mathbf{y} \in W} \frac{\mathbf{v}(\mathbf{y})^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{v}(\mathbf{y})}{\mathbf{v}(\mathbf{y})^\top \mathbf{v}(\mathbf{y})} \\
&= \max_{\mathbf{v} \in \mathbf{v}(W)} \frac{\mathbf{v}^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \\
&\geq \min_{\dim(U)=2} \max_{\mathbf{v} \in U} \frac{\mathbf{v}^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \\
&= \lambda_2(\mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}).
\end{aligned}$$

□

A consequence of this lemma is that if G has second smallest normalised eigenvalue ν_2 , then the Schur complement $\mathbf{S}_C(G, S)$, normalised to its new degrees also has second smallest normalised eigenvalue at least ν_2 . Since the new degrees of the Schur complement, represented by say the matrix \mathbf{D}_S , are entrywise smaller than \mathbf{D}_C , we have that $\lambda_2(\mathbf{D}_S^{-1/2} \mathbf{S} \mathbf{D}_S^{-1/2}) \geq \lambda_2(\mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2})$.

Let $\mathbf{f}(\mathbf{x}) = \mathbf{D}_C^{1/2} \mathbf{D}_S^{-1/2} \mathbf{x}$. Notice that the map \mathbf{f} forms a bijection between vector spaces of dimension 2, since $\mathbf{f}(U)$ is 2-dimensional if U is 2-dimensional, and \mathbf{f} is invertible. We also have that $|\mathbf{f}(\mathbf{x})| \geq |\mathbf{x}|$, since it is entrywise larger. Hence:

$$\begin{aligned}
\lambda_2(\mathbf{D}_S^{-1/2} \mathbf{S} \mathbf{D}_S^{-1/2}) &= \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{x}^\top \mathbf{D}_S^{-1/2} \mathbf{S} \mathbf{D}_S^{-1/2} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \\
&\geq \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{x}^\top \mathbf{D}_S^{-1/2} \mathbf{S} \mathbf{D}_S^{-1/2} \mathbf{x}}{\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})} \\
&= \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{f}(\mathbf{x})^\top \mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2} \mathbf{f}(\mathbf{x})}{\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})} \\
&= \lambda_2(\mathbf{D}_C^{-1/2} \mathbf{S} \mathbf{D}_C^{-1/2}).
\end{aligned}$$

The algorithm in [6] also constructs determinant sparsifiers. For our recursive guarantees to hold, we also have to demonstrate that spectral gap of the normalised Laplacian can not change that much after the sketching process.

LEMMA 4.3. *Consider a graph G so that the spectral gap of its normalised Laplacian is ν_2 . Let H be the graph produced by determinant sparsifier DETSPARSIFY in [6] with some $s = n^{1.5} \delta^{-1}$ edges. Then, the spectral gap of the normalised Laplacian for H is at least some $(1 - O(\delta^{1/4}))\nu_2$.*

Proof. Note that DetSparsify (Algorithm 2 of [6]) essentially samples $s = n^{1.5} \delta^{-0.5}$ edges proportional to the edge leverage scores, and rescaling by a factor of $\exp(\frac{n^2}{2(n-1)s})$. As such, the graph constructed, with weights rescaled back, ie $\exp(-\frac{n^2}{2(n-1)s})H$ is a $\delta^{1/4}$ -spectral sparsifier of G (See Theorem 1 of [14]).

Let $\mathbf{f}(\mathbf{x}) = \mathbf{D}_G^{1/2} \mathbf{D}_H^{-1/2} \mathbf{x}$. \mathbf{f} is a mapping that forms a bijection between two dimensional subspaces $U \rightarrow \mathbf{f}(U)$. Since H is a $\delta^{1/4}$ -spectral sparsifier of G , their degrees differ by a factor of at most $1 + \delta^{1/4}$, and as such $\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \geq (1 - 2\delta^{1/4})\mathbf{x}^\top \mathbf{x}$. We have:

$$\lambda_2(\mathbf{D}_H^{-1/2} \mathbf{H} \mathbf{D}_H^{-1/2}) = \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{x}^\top \mathbf{D}_H^{-1/2} \mathbf{H} \mathbf{D}_H^{-1/2} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$$

$$\begin{aligned}
&= \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{f}(\mathbf{x})^\top D_G^{-1/2} H D_G^{-1/2} \mathbf{f}(\mathbf{x})}{\mathbf{x}^\top \mathbf{x}} \\
&\geq (1 - 2\delta^{1/4}) \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{f}(\mathbf{x})^\top D_G^{-1/2} H D_G^{-1/2} \mathbf{f}(\mathbf{x})}{\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})} \\
&\geq (1 - 3\delta^{1/4}) \min_{\dim(U)=2} \max_{\mathbf{x} \in U} \frac{\mathbf{f}(\mathbf{x})^\top D_G^{-1/2} G D_G^{-1/2} \mathbf{f}(\mathbf{x})}{\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})} \\
&= \lambda_2(D_G^{-1/2} G D_G^{-1/2}).
\end{aligned}$$

□

We now demonstrate that we can solve determinants on expanders. We use the same recursive strategy as in [6]. We begin with a graph G with spectral gap of the normalised Laplacian being ν_2 . To calculate the determinant of \mathbf{L}_G , we find a 2-DD subset of vertices V_2 using Lemma 3.5 of [11], and $V_1 = V \setminus V_2$ and recursively calculate the following:

$$\det_+(\mathbf{L}_G) = \det(\mathbf{L}_{[V_2, V_2]}) \cdot \det_+(\mathbf{Sc}(\mathbf{L}, V_1)).$$

This decomposes into two parts, the determinant of a 2-DD matrix, and $\det_+(\mathbf{Sc}(\mathbf{L}, V_1))$, defined to be the determinant of the Schur complement onto the rest of the vertices with a row and column removed. The entirety of the algorithm is the same, with the only difference being the subroutine used to approximate effective resistances. This leads to the following differences:

1. In [6], $\det(\mathbf{L}_{[V_2, V_2]})$ is calculated by simply completing $\mathbf{L}_{[V_2, V_2]}$ into a Laplacian and recursing. Since our algorithm involves expansion, for our recursion to hold, we have to demonstrate that this new graph, with all of V_1 being contracted to a single vertex, does not have ν_2 that decreases by a large amount. Instead of doing this, we slightly modify the recursion, and use Lemma 3.6.
2. In the second half of the recursion, $\det_+(\mathbf{Sc}(\mathbf{L}, V_1))$ can potentially have too many edges if explicitly represented, so the determinant is calculated by first constructing a determinant sparsifier, and then recursing. If there were no sketches involved, Lemma 4.2 would tell us that taking Schur complements can only increase ν_2 , allowing us to recurse. However, we have to demonstrate that the sketching process does not reduce ν_2 by too much.

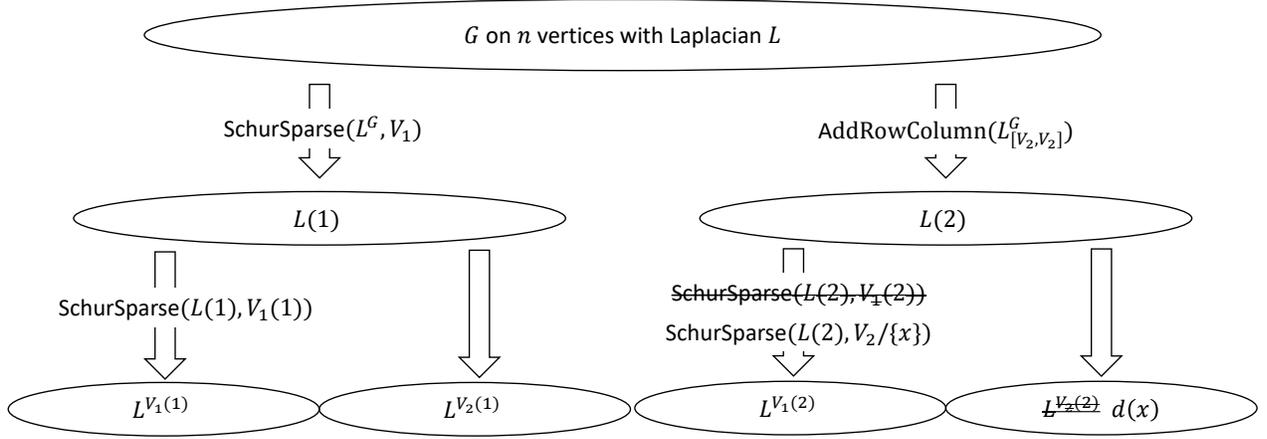
4.1 2-DD matrix Let \mathbf{L}^{V_2} be the completion of $\mathbf{L}_{[V_2, V_2]}$ into a Laplacian, and let the new vertex be x . By removing the single vertex x , we have the following:

$$\begin{aligned}
\det(\mathbf{L}_{[V_2, V_2]}) &= \det_+(\mathbf{L}^{V_2}) \\
&= \deg(x) \cdot \det_+(\mathbf{Sc}(\mathbf{L}^{V_2}, V_2)).
\end{aligned}$$

By Lemma 3.6, we have that we can calculate the effective resistances on all edges of the 2-DD matrix that are not adjacent to the new vertex x , which gives us the effective resistances of the edges in $\mathbf{Sc}(\mathbf{L}^{V_2}, V_2)$. This allows us to construct a determinant preserving sparsifier of the Schur complement H^{V_2} using SchurSparse as in [6], taking total time $\tilde{O}(m + n\varepsilon^{-2})$ to calculate the effective resistances, and $\tilde{O}(s)$ time to sample s edges. In particular we require choices of s and ε such that $\frac{n^2\varepsilon^2}{s}, \frac{n^3}{s^2} \leq \delta^2$, for the guarantees of SchurSparse to hold, so we pick $s = n^{1.5}\delta^{-1}, \varepsilon = n^{-0.25}\delta^{0.5}$, giving us total time $\tilde{O}(s + n\varepsilon^{-2}) = \tilde{O}(n^{1.5}\delta^{-1})$.

We now have a sketch H^{V_2} of size $n^{1.5}\delta^{-1}$. From Lemma 3.6, we know that $\mathbf{Sc}(\mathbf{L}^{V_2}, V_2)$ is a graph with expansion at least some constant. Lemma 4.3 then guarantees that after the sketching process, the ν_2 decreases by at most some $(1 - O(\delta^{1/4}))$ factor. This guarantees that H^{V_2} is also a graph with constant expansion, allowing us to continue by recursing. We are essentially picking a specific 2-DD set to recurse on, so error guarantees still hold through a similar argument as in [6]. We see below in Figure 1 the new recursive structure. [5] prove that the total error in each layer is sufficiently small. We note that the fact that the choice of subset was not important in the error analysis, so we have the same error guarantees. As a result of our recursion on the right half reducing the problem size by only 1 as opposed to a factor of 2, our recursion will have twice the depth, but this also does not affect error guarantees.

Figure 1: Recursive structure of DetApprox from [6], with overlaid changes



4.2 Schur Complement As in the case of [6], we do not have to calculate effective resistances on the Schur complement. Since Schur complements preserve effective resistances, we can instead calculate the effective resistances on L instead of $\mathbf{Sc}(L^{V_2}, V_1)$.

As such, the only difference is in maintaining expansion guarantees through the sketching process. Again, just as in the 2-DD case, by Lemma 4.3, the determinant sparsifier produced H^{V_2} is a rescaled spectral sparsifier of $\mathbf{Sc}(L^{V_2}, V_1)$, and as such the sampled graph H^{V_2} 's second smallest normalised eigenvalue can decrease by at most a factor of $(1 - 2\delta^{1/4})$. Suppose the graph at the topmost level has spectral gap of the normalised Laplacian ν_2 . The resulting graph also always has spectral gap at least some $\tilde{O}(\nu_2)$, since its expansion drops by at most a factor of $2\delta^{1/4}$, or gets set back to a constant value in the case of it being generated from the 2-DD component.

Putting this all together, we see that at each level of our recursion, the costs are dominated by the time required to produce the effective resistance estimates, taking $\tilde{O}(s + n\varepsilon^{-2}\nu_2^{-3}) = \tilde{O}(n^{1.5}\delta^{-1}\nu_2^{-3})$ time. Given that there are only $O(\log n)$ layers, the total time complexity of the algorithm comes out to $\tilde{O}(m + n^{1.5}\delta^{-1}\nu_2^{-3})$. \square

A corollary of the above method is that we can calculate the determinant of any $(1 + \alpha)$ -diagonally dominant matrix M that is a submatrix of a Laplacian (Theorem 1.3).

Proof. (of Theorem 1.3) The proof of Lemma 3.6 gives us that the Schur complement of L_M onto $V \setminus \{x\}$ has ν_2 at least $\alpha/(1 + \alpha)$. Using the same recursive step as in Section 4.2, we essentially work on a graph with ν_2 at least $\alpha/(1 + \alpha)$, giving us a time complexity of $\tilde{O}(m + n^{1.5}\delta^{-1}(1 + \alpha^{-3}))$. \square

References

- [1] Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. “On Sketching Quadratic Forms”. In: *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*. ITCS '16. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 311–319. ISBN: 9781450340571 (cit. on p. 3).
- [2] Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. “Twice-Ramanujan Sparsifiers”. In: *SIAM Journal on Computing* 41.6 (2012), pp. 1704–1721. ISSN: 0097-5397 (cit. on p. 3).
- [3] Karl Bringmann and Konstantinos Panagiotou. “Efficient Sampling Methods for Discrete Distributions”. In: *Algorithmica* (2016) (cit. on p. 7).
- [4] 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: *SIAM Journal on Computing* 0.0 (2018), FOCS18-85-FOCS18-157 (cit. on pp. 1, 2, 3, 10).
- [5] David Durfee, Rasmus Kyng, John Peebles, Anup B. Rao, and Sushant Sachdeva. “Sampling Random Spanning Trees Faster than Matrix Multiplication”. In: STOC 2017. Montreal, Canada: Association for Computing Machinery, 2017, pp. 730–742. ISBN: 9781450345286 (cit. on pp. 2, 12).
- [6] David Durfee, John Peebles, Richard Peng, and Anup B. Rao. “Determinant-Preserving Sparsification of SDDM Matrices”. In: *SIAM Journal on Computing* 49.4 (2017), FOCS17-350-FOCS17-408 (cit. on pp. 1, 2, 3, 9, 10, 11, 12, 13).
- [7] Arun Jambulapati and Aaron Sidford. “Efficient $\tilde{O}(n/\epsilon)$ Spectral Sketches for the Laplacian and Its Pseudoinverse”. In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '18. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2018, pp. 2487–2503. ISBN: 9781611975031 (cit. on pp. 1, 2, 3).
- [8] Svante Janson. “The Numbers of Spanning Trees, Hamilton Cycles and Perfect Matchings in a Random Graph”. In: *Combinatorics, Probability and Computing* 3.1 (1994), pp. 97–126 (cit. on p. 1).
- [9] William B. Johnson and Joram Lindenstrauss. “Extensions of Lipschitz mappings into a Hilbert space”. In: *Conference on Modern Analysis and Probability* (1984), pp. 189–206 (cit. on p. 2).
- [10] Gustav Kirchhoff. “Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der Linearen Vertheilung Galvanischer ströme geführt wird”. In: *Annalen der Physik und Chemie* 148.12 (1847), pp. 497–508 (cit. on p. 1).
- [11] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A. Spielman. “Sparsified Cholesky and multigrid solvers for connection laplacians”. In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing* (2016) (cit. on pp. 9, 12).
- [12] Rasmus Kyng and Sushant Sachdeva. “Approximate Gaussian Elimination for Laplacians - Fast, Sparse, and Simple”. In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. 2016, pp. 573–582 (cit. on pp. 2, 8).
- [13] Kasper Green Larsen and Jelani Nelson. “Optimality of the Johnson-Lindenstrauss lemma”. In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 633–638 (cit. on p. 2).
- [14] Daniel A. Spielman and Nikhil Srivastava. “Graph Sparsification by Effective Resistances”. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. Victoria, British Columbia, Canada: Association for Computing Machinery, 2008, pp. 563–568. ISBN: 9781605580470 (cit. on pp. 1, 2, 3, 11).
- [15] Daniel A. Spielman and Shang-Hua Teng. “Nearly-linear Time Algorithms for Graph Partitioning, Graph Sparsification, and Solving Linear Systems”. In: *STOC*. 2004 (cit. on pp. 1, 2).
- [16] Alastair J. Walker. “An Efficient Method for Generating Discrete Random Variables with General Distributions”. In: *ACM Trans. Math. Softw.* 3.3 (1977), pp. 253–256. ISSN: 0098-3500 (cit. on p. 7).

A Deferred Proofs

We provide some proofs of relatively standard facts here. We first prove a lower bound on the effective resistances.

FACT 2.1. *The effective resistance of any edge (u, v) is lower bounded by $\frac{1}{2}(\frac{1}{d_u} + \frac{1}{d_v})$, where d_u and d_v are the degrees of u and v respectively.*

Proof. For any vector \mathbf{x} , we have that the quadratic form $\mathbf{x}^\top(\mathbf{D} + \mathbf{A})\mathbf{x}$ can be written as:

$$\mathbf{x}^\top(\mathbf{D} + \mathbf{A})\mathbf{x} = \sum_{(u,v) \in E} w_{uv}(\mathbf{x}_u + \mathbf{x}_v)^2.$$

Thus, $\mathbf{D} + \mathbf{A} \succeq \mathbf{0}$, and $\mathbf{L} = \mathbf{D} - \mathbf{A} \preceq 2\mathbf{D}$. Hence, for vectors perpendicular to $\mathbf{1}$, we have $\frac{1}{2}\mathbf{D}^{-1} \preceq \mathbf{L}^+$. This gives us:

$$\begin{aligned} R(u, v) &= (\mathbf{1}_u - \mathbf{1}_v)^\top \mathbf{L}^+ (\mathbf{1}_u - \mathbf{1}_v) \\ &\geq (\mathbf{1}_u - \mathbf{1}_v)^\top \frac{1}{2} \mathbf{D}^{-1} (\mathbf{1}_u - \mathbf{1}_v) \\ &= \frac{1}{2} \left(\frac{1}{d_u} + \frac{1}{d_v} \right). \end{aligned}$$

An alternative proof is as follows. By Rayleigh's Monotonicity Law, the effective resistance between u and v will only decrease if the resistances of edges are lowered. WLOG, let $d_u < d_v$. Construct the graph G' where we decrease the resistance of every single edge not adjacent to u to 0, noting that by Rayleigh's Monotonicity Law, the resistance of u and v in this graph is a lower bound for the resistance between u and v in G . Notice now we can collapse all vertices that are not u into a single vertex, since they are all connected by 0 resistance edges, resulting in a circuit with parallel edges from u to v . This gives us that the effective resistance between u and v is just $\frac{1}{d_u}$, lower bounding the effective resistance between any two vertices by $\max\{\frac{1}{d_u}, \frac{1}{d_v}\}$. \square

Next we prove a result on the convergence of random walks.

LEMMA 3.5. *Given a graph G , and an initial distribution \mathbf{p} , the ℓ_1 norm difference between the distribution of the lazy random walk after t steps and the stationary distribution is bounded by:*

$$\left\| \left(\frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{A}\mathbf{D}^{-1} \right)^t \mathbf{p} - \boldsymbol{\pi} \right\|_1 \leq e^{-t\nu_2/2} \frac{nd_{\max}}{d_{\min}}.$$

where d_{\max} and d_{\min} are the largest and smallest weighted degrees respectively.

Proof. Let $\mathbf{N} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ be the normalized Laplacian. Since this matrix is symmetric, it has an orthogonal basis of eigenvectors. Let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ denote an orthonormal eigenbasis for \mathbf{N} with eigenvalues $\nu_1 \leq \dots \leq \nu_n$ respectively. We have that:

$$\frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{A}\mathbf{D}^{-1} = \mathbf{I} - \frac{1}{2}\mathbf{D}^{1/2}\mathbf{N}\mathbf{D}^{-1/2},$$

which gives us that for each i if $\mathbf{D}^{1/2}\boldsymbol{\psi}_i$ is an eigenvector of $\frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{A}\mathbf{D}^{-1}$ with eigenvalue $(1 - \frac{1}{2}\nu_i)$. Given an initial distribution \mathbf{p} , we write $\mathbf{D}^{-1/2}\mathbf{p}$ as a linear combination of the orthogonal eigenvectors of \mathbf{N} , i.e., for some $\alpha_i \in \mathbb{R}$, we have $\mathbf{D}^{-1/2}\mathbf{p} = \sum_i \alpha_i \boldsymbol{\psi}_i$. Thus, $\mathbf{p} = \sum_i \alpha_i \mathbf{D}^{1/2}\boldsymbol{\psi}_i$, and,

$$\left(\frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{A}\mathbf{D}^{-1} \right) \mathbf{p} = \sum_i \alpha_i \left(1 - \frac{1}{2}\nu_i \right) \mathbf{D}^{1/2}\boldsymbol{\psi}_i.$$

In particular, we have that $\nu_1 = 0$ and $\boldsymbol{\psi}_1 = \frac{1}{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \mathbf{D}^{1/2} \mathbf{1}$. Thus, its coefficient α_1 is

$$\alpha_1 = \boldsymbol{\psi}_1^\top \sum_i \alpha_i \boldsymbol{\psi}_i = \boldsymbol{\psi}_1^\top \mathbf{D}^{-1/2} \mathbf{p} = \frac{1}{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \mathbf{1}^\top \mathbf{p} = \frac{1}{\mathbf{1}^\top \mathbf{D} \mathbf{1}},$$

We bound each individual term after t steps. Let $\boldsymbol{\pi}$ be the stationary distribution of the lazy random walk, $(\boldsymbol{\pi})_u = \frac{d_u}{\sum_u d_u}$. Thus $\boldsymbol{\pi} = \frac{1}{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \mathbf{D} \mathbf{1} = \alpha_1 \mathbf{D}^{1/2} \boldsymbol{\psi}_1$. We have that the ℓ_1 distance between the stationary distribution and the random walk distribution after t steps is bounded by:

$$\begin{aligned}
\left\| \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{p} - \boldsymbol{\pi} \right\|_1 &= \left\| \sum_i \alpha_i \left(1 - \frac{1}{2} \nu_i \right)^t \mathbf{D}^{1/2} \boldsymbol{\psi}_i - \boldsymbol{\pi} \right\|_1 \\
&= \left\| \sum_{i>1} \alpha_i \left(1 - \frac{1}{2} \nu_i \right)^t \mathbf{D}^{1/2} \boldsymbol{\psi}_i + \alpha_1 \mathbf{D}^{1/2} \boldsymbol{\psi}_1 - \boldsymbol{\pi} \right\|_1 \\
&= \left\| \sum_{i>1} \alpha_i \left(1 - \frac{1}{2} \nu_i \right)^t \mathbf{D}^{1/2} \boldsymbol{\psi}_i \right\|_1 \\
&= \left\| \mathbf{D}^{1/2} \sum_{i>1} \alpha_i \left(1 - \frac{1}{2} \nu_i \right)^t \boldsymbol{\psi}_i \right\|_1
\end{aligned}$$

Applying Cauchy-Schwarz, we have that this is bounded by,

$$\begin{aligned}
\left\| \left(\frac{1}{2} \mathbf{I} + \frac{1}{2} \mathbf{A} \mathbf{D}^{-1} \right)^t \mathbf{p} - \boldsymbol{\pi} \right\|_1 &\leq \left\| \mathbf{D}^{1/2} \mathbf{1} \right\|_2 \left\| \sum_{i>1} \alpha_i \left(1 - \frac{1}{2} \nu_i \right)^t \boldsymbol{\psi}_i \right\|_2 \\
&\leq \sqrt{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \sqrt{\sum_{i>1} \alpha_i^2 \left(1 - \frac{1}{2} \nu_i \right)^{2t}} \\
&\leq e^{-t\nu_2/2} \sqrt{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \sqrt{\sum_{i>1} \alpha_i^2} \\
&\leq e^{-t\nu_2/2} \sqrt{\mathbf{1}^\top \mathbf{D} \mathbf{1}} \sqrt{\mathbf{p}^\top \mathbf{D}^{-1} \mathbf{p}} \\
&\leq e^{-t\nu_2/2} \frac{nd_{\max}}{d_{\min}},
\end{aligned}$$

where the last two inequalities follow from $\sum_i \alpha_i^2 = \mathbf{p}^\top \mathbf{D}^{-1} \mathbf{p} \leq 1$.

□