Spectral Sparsification via Bounded-Independence Sampling

Dean Doron
Department of Computer Science, Stanford University, CA, USA
ddoron@stanford.edu

Jack Murtagh
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
jmurtagh@g.harvard.edu

Salil Vadhan
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
salil_vadhan@harvard.edu

David Zuckerman
Department of Computer Science, University of Texas at Austin, TX, USA
diz@cs.utexas.edu

Abstract

We give a deterministic, nearly logarithmic-space algorithm for mild spectral sparsification of undirected graphs. Given a weighted, undirected graph G on n vertices described by a binary string of length N, an integer $k \leq \log n$ and an error parameter $\varepsilon > 0$, our algorithm runs in space $\widetilde{O}(k \log(N \cdot w_{\max}/w_{\min}))$ where $w_{\max}$ and $w_{\min}$ are the maximum and minimum edge weights in G, and produces a weighted graph H with $\widetilde{O}(n^{1+2/k}/\varepsilon^2)$ edges that spectrally approximates G, in the sense of Spielmen and Teng [52], up to an error of ε.

Our algorithm is based on a new bounded-independence analysis of Spielman and Srivastava's effective resistance based edge sampling algorithm [51] and uses results from recent work on space-bounded Laplacian solvers [41]. In particular, we demonstrate an inherent tradeoff (via upper and lower bounds) between the amount of (bounded) independence used in the edge sampling algorithm, denoted by k above, and the resulting sparsity that can be achieved.

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Spectral sparsification, Derandomization, Space complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.39

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at [19], `https://eccc.weizmann.ac.il/report/2020/026/`.

Funding *Dean Doron*: Research supported by a Motwani Postdoctoral Fellowship.
Jack Murtagh: Research supported by NSF grant CCF-1763299.
Salil Vadhan: Research supported by NSF grant CCF-1763299 and a Simons Investigator Award.
David Zuckerman: Research supported in part by NSF Grant CCF-1705028 and a Simons Investigator Award (#409864).

Acknowledgements We thank Jelani Nelson for his insights on spectral sparsification via k-wise independent sampling. We also thank Jarosław Błasiok for helpful discussions about random matrices. The first author would like to thank Tselil Schramm and Amnon Ta-Shma for interesting conversations.

1 Introduction

The graph sparsification problem is the following: given a weighted, undirected graph G, compute a graph H that has very few edges but is a close approximation to G for some definition of approximation. In general, graph sparsifiers are useful for developing more efficient graph-theoretic approximation algorithms. Algorithms whose complexity depend on the number of edges in the graph will be more efficient when run on the sparser graph H, and if H approximates G in an appropriate way, the result on H may be a good approximation to the desired result on G. In this work, we present an algorithm that can be implemented deterministically in small space and achieves sparsification in the *spectral sense* of Spielman and Teng [52]. (See Section 1.2 below for a more formal statement of our main result.)

1.1 Background

Motivated by network design and motion planning, Chew [10] studied *graph spanners*, which are sparse versions of graphs that approximately preserve the shortest distance between each pair of vertices. Benczúr and Karger [6] defined *cut sparsifiers* whose notion of approximation is that every cut of H has size within a $(1 \pm \varepsilon)$ factor of the size of the corresponding cut in G. They showed that every graph G on n vertices has a cut sparsifier H with $O(n \cdot \log n/\varepsilon^2)$ edges and gave a randomized algorithm for computing such cut sparsifiers. Their algorithm runs in nearly linear time (i.e., $\widetilde{O}(m)$ where m is the number of edges in G and the $\widetilde{O}(\cdot)$ notation hides polylogarithmic factors) and they used it to give a faster algorithm for approximating minimum s-t cuts.

Spielman and Teng introduced *spectral sparsifiers*, which define approximation between the graph and its sparsifier in terms of the quadratic forms of their *Laplacians* [52]. The Laplacian of an undirected graph is the matrix $L = D - A$ where A is the adjacency matrix of the graph and D is the diagonal matrix of vertex degrees (i.e. D_{ii} equals the weighted degree of vertex i). H is said to be an *ε-spectral approximation* of G if for all vectors $v \in \mathbb{R}^n$, we have that $v^\top \widetilde{L} v \in (1 \pm \varepsilon) \cdot v^\top L v$, where $\widetilde{L}$ and L are the Laplacians of H and G, respectively. Spectral sparsifiers generalize cut sparsifiers, which can be seen by observing that when $v \in \{0,1\}^n$, v is the characteristic vector of some set of vertices $S \subseteq [n]$ and $v^\top L v$ equals the sum of the weights of the edges cut by S.

Spielman and Teng showed that all graphs have spectral sparsifiers with $O(n \cdot \log^{O(1)} n/\varepsilon^2)$ edges and gave a nearly linear time randomized algorithm for computing them with high constant probability. Their spectral sparsifiers were a key ingredient that they used to develop the first nearly linear time algorithm for solving Laplacian systems. These fast Laplacian solvers spawned a flurry of improvements and simplifications [16, 26, 28, 27, 30, 31, 34, 44] as well as extensions to directed graphs [15, 14, 13] and to the space-bounded setting [18, 41, 1]. Spectral sparsification and the nearly linear time Laplacian solvers that use them have been critical primitives that have enabled the development of faster algorithms for a wide variety of problems including max flow [37, 11, 17, 24, 33], random generation of spanning trees, [25, 40, 49], and other problems in computer science [43, 29].

Spielman and Srivastava [51] gave a spectral sparsification algorithm that both simplified and improved upon the algorithm of Spielman and Teng. They show that randomly sampling edges, independently with probabilities proportional to their *effective resistances* produces a good spectral sparsifier with high probability. Viewing a graph as an electrical network, the effective resistance of an edge (a, b) is the potential difference induced between them when a unit of current is injected at a and extracted at b (or vice versa). More formally, the effective

resistance of an edge (a,b) in a graph with Laplacian L is $R_{ab} = (e_a - e_b)^\top L^+ (e_a - e_b)$, where e_i denotes the ith standard basis vector and L^+ denotes the *Moore-Penrose pseudoinverse* of L[1]. Spielman and Srivastava proved the following theorem.

▶ **Theorem 1** (spectral sparsification via effective resistance sampling [51, 50]). *Let $G = (V, E, w)$ be a weighted graph on n vertices and for each edge $(a,b) \in E$ with weight w_{ab}, define $p_{ab} = \min\{1, 4 \cdot \log n \cdot w_{ab} \cdot R_{ab}/\varepsilon^2\}$, where R_{ab} is the effective resistance of (a,b) as defined above. Construct a sparsifier H by sampling edges from G independently such that each edge (a,b) in G is added to H with probability p_{ab}. For edges that get added to H, reweight them with weight w_{ab}/p_{ab}. Let L and $\widetilde{L}$ be the Laplacians of G and H, respectively. Then, with high probability,*

1. *H has $O(n \cdot (\log n)/\varepsilon^2)$ edges, and,*
2. *$\widetilde{L}$ ε-spectrally approximates L.*

Furthermore, this procedure can be implemented to run in time $\widetilde{O}(\frac{m}{\varepsilon^2} \cdot \log(w_{\max}/w_{\min}))$, where m is the number of edges in G and $w_{\max}, w_{\min}$ are the maximum and minimum edge weights of G, respectively.[2]

The sparsity achieved by the Spielman and Srivastava sparsifiers was improved by Batson, Spielman and Srivastava [5], who gave a deterministic algorithm for computing ε-spectral sparsifiers with $O(n/\varepsilon^2)$ edges, which is asymptotically optimal, however, their algorithm is less efficient, running in time $O(m \cdot n^3/\varepsilon^2)$. Work on these optimal sparsifiers continued with another slightly faster deterministic algorithm [53] followed by an $O(n^{2+\varepsilon})$-time randomized algorithm [2], and culminating in the randomized algorithms of Lee and Sun who achieved almost-linear time [35] and finally nearly-linear time [36].

1.2 Our Main Result

In this work we study the deterministic *space complexity* of computing spectral sparsifiers. Our main result is a deterministic, nearly-logarithmic space algorithm for computing *mild spectral sparsifiers*, that is, graphs with $O(n^{1+\alpha}/\varepsilon^2)$ edges for any constant $\alpha > 0$.

▶ **Theorem 2** (see also Theorem 19). *Let G be a connected, weighted, undirected graph on n vertices, $k \in \mathbb{N}$ an independence parameter and $\varepsilon > 0$ an error parameter. There is a deterministic algorithm that on input G, k, and ε, outputs a weighted graph H that is an ε-spectral sparsifier of G and has $O(n^{1+2/k} \cdot (\log n)/\varepsilon^2)$ edges. The algorithm runs in space $O(k \log(N \cdot w) + \log(N \cdot w) \log\log(N \cdot w))$, where $w = w_{\max}/w_{\min}$ is the ratio of the maximum and minimum edge weights in G and N is the length of the input.*

The closest analogue to spectral sparsifiers in the space-bounded derandomization literature is the *derandomized square* of Rozenman and Vadhan [47], a graph operation that produces a sparse approximation to the square of a graph.[3] The derandomized square was introduced to give an alternative proof to Reingold's celebrated result that Undirected S-T Connectivity can be solved in deterministic logspace [46]. Murtagh, Sidford, Reingold,

[1] L^+ is a matrix with the same kernel as L that acts as an inverse of L on the orthogonal complement of the kernel. See Section 2.2 for a formal definition.

[2] In their original paper, [51], they fix the number of edges in the sparsifier in advance resulting in a slightly different theorem statement and analysis. The version we cite here and what we model our algorithm after was presented later in [50].

[3] The *square* of a graph G is a graph on the same vertex set whose edges correspond to all walks of length 2 in G.

and Vadhan [41] showed that the derandomized square actually produces a *spectral sparsifier* of the square of a graph and this was a key observation they used to develop a deterministic, nearly logarithmic space algorithm for solving Laplacian systems. Later the sparsification benefits of the derandomized square were also used in nearly logarithmic space algorithms for deterministically approximating random walk probabilities and for solving Laplacian systems in Eulerian directed graphs [42, 1].

For a d-regular graph G on n vertices, its square G^2 has degree d^2 and the derandomized square computes an ε-spectral approximation to G^2 with degree $O(d/\varepsilon^2)$. On the other hand, applying our sparsification to G^2 results in an ε-spectral approximation with on average $O(n^\alpha/\varepsilon^2)$ edges adjacent to each vertex for any constant α, which is independent of d and much sparser when $d = \omega(n^\alpha)$. Also, our algorithm can sparsify any undirected graph, not just squares. Our algorithm does not replace the derandomized square, however, because the derandomized square can be iterated very space efficiently, a property that is used in all of its applications thus far. Nevertheless, given the success of spectral sparsification and Laplacian solvers in the nearly-linear time context and the fruit borne of porting these techniques to the logspace setting, we are hopeful that our spectral sparsifiers will have further applications in derandomization of space-bounded computation.

1.3 Techniques

Our deterministic space-efficient algorithm is modeled after the effective resistance based sampling algorithm of Spielman and Srivastava (Theorem 1). Although the Spielman and Srivastava procedure is randomized and does not achieve optimal sparsity, the known algorithms that do ([5, 53, 2, 35, 36]) are more involved and often sequential in nature so do not seem as amenable to small-space implementations.

To derandomize the Spielman-Srivastava algorithm, we follow the standard approach of first reducing the number of random bits used to logarithmic, and then enumerating over all random choices of the resulting algorithm. Following [39, 3], a natural way to reduce the number of random bits used is to do the edge sampling only k-wise independently for some $k \ll |E|$ rather than sampling every edge independently from all other edges.

Let k be our bounded-independence parameter. Namely, we are only guaranteed that every subset of k edges is chosen independently (with the right marginals), however there may be correlations between the choices in tuples of size $k+1$. It is well known that such a sampling can be performed using fewer random bits. By [51], we know that $k = |E|$ will, with high probability, produce an ε-spectral sparsifier with $O(n \cdot \log n/\varepsilon^2)$ edges in expectation. What about much smaller values of k? In Section 3, we prove the following:

▶ **Theorem 3** (informal; see Theorem 9). *Let G be a connected weighted undirected graph on n vertices with Laplacian L, $k \in \mathbb{N}$ an independence parameter and $\varepsilon > 0$ an error parameter. Let H be the graph which is the output of Spielman and Srivastava's sampling-based sparsification algorithm (Theorem 1), when the edge sampling is done in a k-wise independent manner, and let $\widetilde{L}$ be the Laplacian of H. Then, with high constant probability, $\widetilde{L}$ ε-approximates L and H has $O(n^{1+2/k} \cdot (\log n)/\varepsilon^2)$ edges.*

A first thing to observe is that $k = \log n$ gives the same result as in [51]. More importantly, the above shows that the result *interpolates*: Even for a constant k, Theorem 3 gives a *mild sparsification* that sparsifies dense graphs to $O(n^{1+\alpha})$ expected edges, where $\alpha > 0$ is an arbitrarily small constant.

We prove Theorem 3 by extending the arguments in [51, 50]. For every edge $(a, b) \in E$, we define a random matrix X_{ab} that corresponds to the choice made by the sparsification algorithm, in such a way that $X = \sum_{(a,b)\in E} X_{a,b}$ relates to the resulting Laplacian $\widetilde{L}$.[4] Let Π be the orthogonal projection onto the image of L. Following [51, 50], we show that $\widetilde{L}$ ε-spectrally approximates L (equivalently, that H is an ε-spectral sparsifier for G) with high probability if $X - \Pi$ has bounded moments. Deriving a tail bound that relies on the first k moments alone, we can proceed with the analysis as if the X_{ab}'s were *truly independent*. More specifically, we bound $\mathrm{Tr}(\mathbb{E}_X[(X - \Pi)^k])$ using a matrix concentration result due to Chen, Gittens and Tropp [9]. For the complete details, as well as how our argument differs from [51, 50], see Section 3.

Getting a Deterministic Algorithm

Theorem 3 readily gives a simple, randomness-efficient algorithm, as k-wise independent sampling of edges only requires $O(k \cdot \log(N \cdot w))$ random bits [23, 3] (See Lemma 8). However, more work is needed to obtain a space-efficient deterministic algorithm. First, we need to be able to compute the marginal sampling probabilities, which depend on the effective resistances R_{ab}. Fortunately, the recent work of Murtagh et al. [41] allows us to approximate the effective resistances using only $O(\log(N \cdot w) \log\log(N \cdot w))$ space and we show that the k-wise independent sampling procedure can tolerate the approximation.

Next, to obtain a deterministic algorithm, we can enumerate over all possible random choices of the algorithm in space $O(k \cdot \log(N \cdot w))$ and compute a candidate sparsifier H for each. We are guaranteed that at least one (indeed, most) of the resulting graphs H is a good sparsifier for G but how can we identify which one? To do this, it suffices for us, given Laplacians L and $\widetilde{L}$, to distinguish the case that $\widetilde{L}$ is an ε-spectral approximation of L from the case that $\widetilde{L}$ is not a $2 \cdot \varepsilon$-spectral approximation of L. We reduce that problem to that of approximating the spectral radius of $M = ((\widetilde{L} - L)L^+/\varepsilon)^2$, where L^+ is the pseudoinverse of L, which can be approximated in nearly logarithmic space by [41]. In fact, it will be sufficient to check whether the trace of a logarithmically high power of M is below a certain threshold to deduce that the spectral radius of M does not exceed 1. In Section 5.2, we show that the latter case implies that $\widetilde{L}$ indeed ε-approximates L.

The deterministic, nearly logarithmic space Laplacian solver of [41] only worked for *multigraphs*, i.e. graphs with integer edge weights. To get our result for arbitrary weighted graphs, we extend the work of [41] and give a deterministic, nearly logarithmic space Laplacian solver for arbitrary undirected weighted graphs. Combining this extension with the k-wise independent analysis of the edge sampling algorithm (Theorem 3) and the verification procedure described above lets us prove our main result Theorem 2.

1.4 Lower Bounds for Bounded-Independence Sampling

Having established an upper bound on the amount of independence required for the edge-sampling procedure (Theorem 3), a natural goal would be to come up with a corresponding lower bound. Theorem 3 tells us that in order to sparsify to $\widetilde{O}(n^{1+\alpha})$ expected edges, we can use k-wise independent sampling for $k = 2/\alpha$. Can a substantially smaller choice of k perform just as well? In Section 4, we show that our upper bound of $k = 2/\alpha$ is tight up to a small constant factor.

[4] Specifically, $X = L^{+/2}\widetilde{L}L^{+/2}$, where $L^{+/2}$ is the square-root of the pseudoinverse of L.

▶ **Theorem 4** (informal; see Theorem 11). *For every small enough $\alpha > 0$ there exist infinitely many connected graphs $G = (V = [n], E)$ with all effective resistances equal that are d-regular with $d = \Omega(n^{\alpha})$ and a distribution $\mathcal{D} \sim \{0,1\}^{|E|}$ that is k-wise independent for $k = \lfloor 4/3\alpha \rfloor$ with marginals $1/2$ that would fail to produce an ε-spectral sparsifier of G to within any $\varepsilon > 0$ with high probability.*

Our family of "bad graphs" will be dense graphs having large girth. Namely, given a girth g and an integer $d \geq 3$, we consider graphs $G = (V = [n], E)$ satisfying $d \geq n^{\gamma/g} + 1$ for some constant $0 < \gamma < 2$ [32]. Getting an infinite family of graphs with γ approaching 2 (and specifically attaining the *Moore bound*), even non-explicitly, has been the subject of extensive study (see [21] and references therein). See also Section 4.1 for a further discussion. Given a sparsification parameter $\alpha > 0$, we set $k \approx \gamma/\alpha$ and take a graph G on n vertices with girth $g = k + 1$ and degree $d > n^{\gamma/g} + 1$.

Our construction of the distribution $\mathcal{D}$ is inspired by Alon and Nussboim [4]: choose a partition of the vertices $V = V_0 \uplus V_1$ uniformly at random, and for every edge $e = (u, v) \in E$, include it in the sample if and only if either $u, v \in V_0$ or $u, v \in V_1$. Clearly, sampling edges according to $\mathcal{D}$ results in a disconnected graph almost surely. However, we show that $\mathcal{D}$ is indeed k-wise independent, relying on the fact that the girth of G is $k + 1$.

To obtain Theorem 4 we use the family of graphs given by Lazebnik et al. [32] who obtained $\gamma = 4/3$. Indeed, any improvement in γ would bring our upper bound of $k \approx 2/\alpha$ and lower bound of $k \approx \gamma/\alpha$ closer together.

1.5 Open Problems

An interesting open problem is to achieve improved sparsity, e.g. $O(n \cdot (\log n)/\varepsilon^2)$ matching [51]. Our algorithm would require space $\Omega(\log^2 n)$ to achieve this sparsity, due to setting $k = \Omega(\log n)$. We remark that previous work implies that this can be done in *randomized* logarithmic space. Indeed, Doron et al. [18] gave a randomized algorithm for solving Laplacian systems in logarithmic space (without $\log\log(\cdot)$ factors), and this implies that one can approximate effective resistances and hence implement the Spielman-Srivastava edge sampling with full independence in randomized logspace. It is also an interesting question whether there is a nearly logspace algorithm (even randomized) that produces spectral sparsifiers of optimal sparsity (i.e., $O(n/\varepsilon^2)$ edges).

Finally, while it is not known how to compute spectral sparsifiers of arbitrary *directed* graphs, there has been progress on sparsifying Eulerian digraphs in the nearly-linear time literature [15, 14, 13, 12]. Given the recent advance of a nearly-logarithmic space solver for Eulerian Laplacian systems [1], an interesting question is sparsifying Eulerian graphs in small space.

2 Preliminaries

We will work with undirected weighted graphs, $G = (V, E, w)$, where w is a vector of length $|E|$ and each edge $(a, b) \in E$ is associated with a positive weight $w_{ab} > 0$. At times we refer to undirected *multigraphs*, which are weighted graphs where all of the weights are integers. The adjacency matrix of G is a symmetric, real-valued matrix A in which $A_{ij} = w_{ij}$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise.

For any matrix A, its *spectral norm* $\|A\|$ is $\max_{\|x\|=1} \|Ax\|_2$, which is also the largest singular value of A. For any square matrix A, its *spectral radius*, denoted $\rho(A)$, is the largest absolute value of its eigenvalues. When A is real and symmetric, the spectral norm equals the spectral radius. The spectral norm is sub-multiplicative, i.e., $\|AB\| \leq \|A\| \|B\|$. We denote by $A^\top$ the transpose of A. We denote by $\mathbf{1}$ the all-ones vector, by $\mathbf{0}$ the all-zeros vector, and e_a is the vector with 1 in the a-th coordinate and 0 elsewhere, where e_a's dimension will be understood from context (i.e., e_a is the a-th standard basis vector).

The *trace* of a matrix $A \in \mathbb{R}^{n\times n}$, is $\mathrm{Tr}(A) = \sum_{i\in[n]} A_{ii}$, which also equals the sum of its eigenvalues. The trace is invariant under cyclic permutations, i.e., $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$. The expectation of a *random matrix* is the matrix of the coordinate-wise expectations. More formally, if A is a random matrix, then $\mathbb{E}[A] = \widehat{A}$ where $\widehat{A}_{ij} = \mathbb{E}[A_{ij}]$ for all $i, j \in [n]$. The trace and the expectation are both linear functions of a matrix and they commute. That is, for all random matrices A, we have $\mathrm{Tr}(\mathbb{E}[A]) = \mathbb{E}[\mathrm{Tr}(A)]$ (see, e.g.,[45]).

2.1 PSD Matrices and Spectral Approximation

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *positive semi-definite* (PSD), denoted $A \succeq 0$, if for every $x \in \mathbb{R}^n$ it holds that $x^\top Ax \geq 0$, or equivalently, if all its eigenvalues are non-negative. We write $A \succeq B$ if $A - B \succeq 0$.

▶ **Definition 5.** *Let A and B be $n \times n$ symmetric PSD matrices. For a real $\varepsilon > 0$, we say that A is an* ε-spectral approximation *of B, denoted $A \approx_\varepsilon B$, if*

$$(1-\varepsilon)B \preceq A \preceq (1+\varepsilon)B.$$

When A and B share an eigenvector basis $v_1, \ldots, v_n$, Definition 5 is equivalent to requiring $(1-\varepsilon)\mu_i \leq \lambda_i \leq (1+\varepsilon)\mu_i$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A corresponding to $v_1, \ldots, v_n$ and $\mu_1, \ldots, \mu_n$ are the eigenvalues of B corresponding to $v_1, \ldots, v_n$.

2.2 The Moore-Penrose Pseudoinverse

Let A be any linear operator. The *Moore-Penrose pseudoinverse* of A, denoted A^+, is defined as follows. If $A = U\Sigma V^\top$ is the singular value decomposition (SVD) of A, the pseudoinverse is given by $A^+ = V\Sigma^+U^\top$ where Σ^+ is the matrix obtained by taking the reciprocal of each nonzero diagonal element of Σ, and leaving the zeros intact. When A is a symmetric PSD matrix, the SVD coincides with the eigen-decomposition and so if $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A then A^+ shares the same eigenvector basis and has eigenvalues $\lambda_1^+, \ldots, \lambda_n^+$, where $\lambda_i^+ = 1/\lambda_i$ if $\lambda_i \neq 0$ and $\lambda_i^+ = 0$ otherwise.

Also note that if A is real then A^+ is real-valued as well.

A *square root* of a matrix A is any matrix X that satisfies $X^2 = A$. When A is symmetric and PSD, it has a unique symmetric PSD square root, which we write as $A^{1/2}$. If $A = U\Sigma U^\top$ is the eigen-decomposition of A then $A^{1/2} = U\sqrt{\Sigma}U^\top$ where $\sqrt{\Sigma}$ is obtained by taking the square root of each diagonal element of Σ. We denote by $A^{+/2}$ the matrix $(A^+)^{1/2} = (A^{1/2})^+$.

2.3 The Graph Laplacian and Effective Resistance

Given a graph G on n vertices with an adjacency matrix A and degree matrix D (i.e., D is a diagonal matrix where $D_{ii} = \sum_{j=1}^n A_{ij}$ equals the weighted degree of vertex i in G), the *Laplacian* of G is the matrix

$$L = D - A.$$

For every undirected weighted graph $G = (V, E, w)$, its Laplacian L is symmetric and PSD, with smallest eigenvalue 0. The zero eigenvalue has multiplicity one if and only if G is connected. In this case, $\ker(L_G) = \text{span}(\{\mathbf{1}\})$. For every edge $(a, b) \in E$, define the *edge Laplacian* of (a, b) to be

$$L_{ab} = (e_a - e_b)(e_a - e_b)^\top = (e_b - e_a)(e_b - e_a)^\top.$$

Note that $L = \sum_{(a,b)\in E} w_{ab} \cdot L_{ab}$.

It is often helpful to associate G with an electric circuit, where an edge $(a, b) \in E$ corresponds to a resistor of resistance $1/w_{ab}$. For each pair of vertices a and b, the *effective resistance* between them, denoted by R_{ab}, is the energy of the electrical flow that sends one unit of current from a to b. The effective resistance can be calculated using the pseudoinverse of the Laplacian:

$$R_{ab} = (e_a - e_b)^\top L^+ (e_a - e_b).$$

(See [8] for more information on Laplacians and viewing graphs as electrical networks). A useful fact about effective resistances is Foster's Theorem:

▶ **Theorem 6** ([22]). *For every undirected weighted graph $G = (V, E, w)$ on n vertices it holds that*

$$\sum_{(a,b)\in E} w_{ab} \cdot R_{ab} = n - 1.$$

2.4 Bounded-Independence Sampling

Given a probability vector $p \in [0, 1]^m$, let Bernoulli(p) denote the distribution X over $\{0, 1\}^m$ where the bits are independent and for each $i \in [m]$, $\mathbb{E}[X_i] = p_i$. For a set $I \subseteq [m]$ and a string $z \in \{0, 1\}^m$, we let $z|_I \in \{0, 1\}^{|I|}$ be the restriction of z to the indices in I.

▶ **Definition 7.** *We say a distribution $X \sim \{0, 1\}^m$ is k-wise independent with marginals $p \in [0, 1]^m$ if for every set $I \subseteq [m]$ with $|I| \leq k$, it holds that $X|_I = \text{Bernoulli}(p|_I)$. We refer to X as a k-wise independent sample space with marginals p.*

Consider $G = (V, E, w)$ with $|E| = m$. Throughout, when we say *sampling edges in a k-wise independent manner*, we refer to the process of picking an element $x \in \{0, 1\}^m$ from a k-wise independent sample space uniformly at random and taking those edges $e \in E$ for which $x_e = 1$.

For $p \in [0, 1]^m$ and a positive integer t, we define $\lfloor p \rfloor_t$ to be the vector p' obtained by truncating every element of p after t bits. Thus, for each $i \in [m]$, $p'_i = 2^{-t}\lfloor 2^t p_i \rfloor$, and so $|p_i - p'_i| \leq 2^{-t}$. The following lemma states that we can construct small k-wise independent sample spaces with any specified marginals.

▶ **Lemma 8** (following [23, 3]). *For every $m, k, t \in \mathbb{N}$ and $p \in [0, 1]^m$ there exists an explicit k-wise independent distribution $X \sim \{0, 1\}^m$ with marginals $\lfloor p \rfloor_t$, that can be sampled with $r = O(k \cdot \max\{t, \log m\})$ truly random bits. Furthermore, given $\rho \in \{0, 1\}^r$, the element $x \in \text{Supp}(X)$ corresponding to the random bits ρ can be computed in $O(k \cdot \max\{t, \log m\})$ space.*

3 Sparsification via Bounded-Independence Sampling

In Section 1, we briefly introduced the Spielman-Srivastava sparsification algorithm [51] based on (truly) independent edge sampling, with probabilities proportional to the effective resistances of the edges. In this section, we explore the tradeoff between the amount of independence used in the edge sampling process and the resulting sparsity that can be achieved.

In particular, we analyze the algorithm Sparsify (see Algorithm 1). The algorithm gets as input an undirected, weighted, dense graph $G = (V, E, w)$ on n vertices, approximate effective resistances $\widetilde{R}_{ab}$ for each edge $(a, b) \in E$, a bounded independence parameter $k \leq \log n$, a desired approximation error $\varepsilon > 0$, and a parameter $\delta > 0$ governing the success probability, and outputs a sparser graph H whose Laplacian ε-spectral approximates the Laplacian of G with probability at least $1 - 2\delta$.

Algorithm 1 Computing a spectral sparsifier via bounded independence sampling.

Sparsify$(G = (V, E, w), \{\widetilde{R}_{ab}\}_{(a,b)\in E}, k, \varepsilon, \delta)$.

1. Initialize H to be the empty graph on $n = |V(G)|$ vertices.
2. Set $s \leftarrow \frac{18e \log n}{\varepsilon^2} \cdot \left(\frac{n}{\delta}\right)^{2/k}$.
3. For every edge $(a, b) \in E$, set $p_{ab} \leftarrow \min\left\{1, w_{ab} \cdot \widetilde{R}_{ab} \cdot s\right\}$.
4. For every edge $(a, b) \in E$, add (a, b) to H with weight w_{ab}/p_{ab} with probability p_{ab}. Do this sampling *in a k-wise independent manner*, following Lemma 8.
5. Return H.

First we will analyze Sparsify for the case where the effective resistances are given exactly, i.e. $\widetilde{R}_{ab} = R_{ab}$ for all $(a, b) \in E$. Then, in Section 3.2 we will analyze the more general case where we are given approximations to the effective resistances. This latter case is useful algorithmically because more efficient algorithms are known for estimating effective resistances than for computing them exactly, both in the time-bounded and space-bounded settings [51, 41].

3.1 Sparsification With Exact Effective Resistances

In this section we give our main theorem about Sparsify.

▶ **Theorem 9** (spectral sparsification via bounded independence). *Let $G = (V, E, w)$ be an undirected connected weighted graph on n vertices with Laplacian L and effective resistances $R = \{R_{ab}\}_{(a,b)\in E}$. Let $0 < \varepsilon < 1$, $0 < \delta < 1/2$ and let $k \leq \log n$ be an even integer. Let H be the output of* Sparsify$(G, R, k, \varepsilon, \delta)$ *and let $\widetilde{L}$ be its Laplacian. Then, with probability at least $1 - 2\delta$ we have:*

1. *$\widetilde{L} \approx_\varepsilon L$, and,*
2. *H has $O\left(\frac{1}{\delta^{1+2/k}} \cdot \frac{\log n}{\varepsilon^2} \cdot n^{1+\frac{2}{k}}\right)$ edges.*

Spielman and Srivastava showed that by using truly independent sampling (i.e., $k = |E|$) in Sparsify, one can compute an ε-spectral sparsification of G with $O(n \cdot \log n/\varepsilon^2)$ edges, with high constant probability [51]. One immediate consequence of Theorem 9 is that $\log n$-wise independent sampling suffices to match the sparsity that truly independent sampling achieves. Another consequence of Theorem 9 is that for any constant $0 < \alpha < 1$ and any constant $\gamma < \alpha/2$, for $k \approx 2/(\alpha - 2\gamma)$, k-wise independent sampling achieves a spectral sparsifier with error $\varepsilon = n^{-\gamma}$ and $O(n^{1+\alpha})$ expected edges, with high constant probability.

The proof of Theorem 9 is modeled after Spielman and Srivastava's argument [51]. One difference is that the sparsification algorithm in [51] fixes the number of edges to be sampled in advance rather than having the number of edges be a random variable. They then prove spectral approximation by reducing the problem to a question about concentration of random matrices, which they resolve with a matrix Chernoff bound due to Rudelson and Vershynin [48]. We follow a variant of this argument for the case where the number of edges in the sparsifier is random and use a matrix concentration bound of Chen, Gittens, and Tropp [9]. This variant, for truly independent sampling, has appeared before in [50]. Our argument is modified to address the fact that we only use k-wise independent sampling, and the proof is given in the full version of the paper.

3.2 Sparsification With Approximate Effective Resistances

Spielman and Srivastava showed that the original version of spectral sparsification through effective resistance sampling (with fully independent sampling and fixing the number of edges in advance) is robust to small changes in the sampling probabilities. In this section we show the same is true of Sparsify. As said, this is useful because more efficient algorithms are known for estimating effective resistances than for computing them exactly, and we will also use this fact for our space-bounded algorithm for sparsification in Section 5.

The lemma below says that if we only have small multiplicative approximations to the effective resistances then the guarantees of Theorem 9 still hold with a small loss in the sparsity.

▶ **Lemma 10.** *Let $G = (V, E, w)$ be an undirected connected weighted graph on n vertices with Laplacian L. Let $0 < \varepsilon < 1$, $0 < \delta < 1/2$ and let $k \leq \log n$ be an even integer. For each $(a, b) \in E$, let $\widetilde{R}_{ab}$ be such that*

$$(1 - \alpha) \cdot R_{ab} \leq \widetilde{R}_{ab} \leq (1 + \alpha) \cdot R_{ab},$$

where R_{ab} is the effective resistance of (a, b) and $0 < \alpha < 1$. Let $\widetilde{R} = \{\widetilde{R}_{ab}/(1 - \alpha)\}_{(a,b)\in E}$. Let H be the output of Sparsify$(G, \widetilde{R}, k, \varepsilon, \delta)$ *and let $\widetilde{L}$ be its Laplacian. Then, with probability at least $1 - 2\delta$ we have:*

1. *$\widetilde{L} \approx_\varepsilon L$, and,*
2. *H has $O\left(\frac{1+\alpha}{1-\alpha} \cdot \frac{1}{\delta^{1+2/k}} \cdot \frac{\log n}{\varepsilon^2} \cdot n^{1+\frac{2}{k}}\right)$ edges.*

A proof of Lemma 10 can be found in the full version. Note that we could equivalently define Sparsify to take approximate sampling probabilities as input (i.e., $(1 - \alpha)p_{ab} \leq \widetilde{p}_{ab} \leq (1 + \alpha)p_{ab}$) rather than α-approximate effective resistances and the same lemma applies.

4 Lower Bounds for Bounded-Independence Sampling

In this section we give a lower bound for sampling-based bounded independence sparsification. Our lower bound will hold even for unweighted, simple, regular graphs in which all the effective resistances are the same, so for this section, assume $G = (V = [n], E)$ is such a graph. In Section 3 we measure sparsity in terms of the number of edges in the graph. We use this measure rather than average degree because in weighted graphs, the degree of a vertex v typically refers to the sum of the weights of the edges incident to v, whereas in sparsification algorithms we are trying to minimize the number of edges incident to v, regardless of their weight. In this section, we will sometimes refer to average degree rather than number of edges. When we refer to the average degree of a weighted graph, we mean the average number of edges incident to each vertex. For simple, unweighted graphs, these quantities are the same.

Fix some $\alpha > 0$. Theorem 9 tells us that if we want to sparsify G to within error ε and expected degree $s = O\left(n^{\alpha} \cdot \log n/\varepsilon^2\right)$, we can do so by sampling each edge with probability $p = s \cdot (n-1)/|E|$ in a k-wise independent manner, where $k = 2/\alpha$ (rounded to an even integer).[5] We now prove that $k \geq 4/3\alpha$ is essential for such a sampling procedure, at least for constant α.

▶ **Theorem 11** (lower bound for spectral sparsification via bounded independence)**.** *Fix $c > 0$. For every $\alpha \in (0, 4/15]$, there exist infinitely many n's for which the following holds.*

There exists a connected graph $G = (V = [n], E)$ whose effective resistances are all equal and a distribution $\mathcal{D} \sim \{0,1\}^{|E|}$ that is k-wise independent for $k = \lfloor 4/3\alpha \rfloor$ with marginals $1/2$ that would fail to sparsify G to within any error $\varepsilon > 0$ and expected degree $s = c \log n \cdot n^{\alpha_0}$, where $\alpha_0 \geq (1-2\alpha)\alpha$.

More specifically, sampling a subgraph of G according to $\mathcal{D}$ would result in a disconnected graph with probability at least $1 - 2/2^n$.

We note that a disconnected graph fails to be a good spectral sparsifier of a connected graph, which is implicit in Theorem 11. Formally:

▷ Claim 12. Let G and $\widetilde{G}$ be undirected graphs on n vertices with Laplacians L and $\widetilde{L}$, respectively. If G is connected and $\widetilde{G}$ is disconnected then $\widetilde{L} \not\approx_{\varepsilon} L$ for any $\varepsilon > 0$.

We give a proof of Claim 12 in the full version of the paper.

4.1 Moore-Like Graphs With a Given Girth

Toward proving Theorem 11, we will need, for every bounded-independence parameter k, an infinite family of graphs satisfying certain properties. Recall that the *girth* of a graph G is the length of the shortest cycle in G. We will need an infinite family of girth-g graphs having large degree. Formally:

▶ **Definition 13.** *Given $\gamma > 0$ and $g\colon \mathbb{N} \to \mathbb{N}$, an infinite family of graphs $\{G_i = (V_i = [n_i], E_i)\}_{i \in \mathbb{N}}$ is* (g,γ)-Moorish *if for every $i \in \mathbb{N}$, G_i is connected, has girth at least $g(n_i)$ and is d_i-regular for some $d_i \geq n_i^{\gamma/g(n_i)} + 1$.*

The problem of finding such families of graphs, or even proving their existence in some regime of parameters, has been widely studied in extremal graph theory. A simple counting argument ([20], see also [8]) shows that (g,γ)-Moorish families of graphs can only exist when $\gamma \leq 2$:

▶ **Lemma 14** (the Moore bound, see, e.g., [8])**.** *Every d-regular graph of girth g on n vertices satisfies $n \geq 2 \cdot ((d-1)^{g/2} - 1)/(d-2)$.*

Still, no families with γ approaching 2 for arbitrary girths are known. The Ramanujan graphs of Lubotzky, Phillips and Sarnak [38] were shown to obtain $\gamma \geq 4/3$ by Biggs and Boshier [7]. Lazebnik, Ustimenko and Woldar [32] slightly improved upon [38] in the lower-order terms, but more importantly for us, the family they construct consists of *edge-transitive graphs*.

[5] We used the fact that for every $(a,b) \in E$, $p_{ab} \leftarrow \min\{1, R_{ab}s\} = R_{ab}s = R \cdot s$, which can be argued as follows. When all effective resistances equal R, we have $R = (n-1)/|E|$ due to Theorem 6. Now, if G has $n \cdot s$ edges or fewer, then it already achieves the desired sparsity so without loss of generality we can assume that $|E| > n \cdot s$. Hence, $R \cdot s < (n-1)s/ns < 1$. Also, the resulting graph should indeed be a weighted one, however all its weights will be the same, $1/p$.

▶ **Theorem 15** ([32]). *For every prime power d and even integer $g \geq 6$ there exists a d-regular explicit simple, edge-transitive graph with $n \leq 2d^{g-\lfloor \frac{g-3}{4} \rfloor - 4}$ vertices and girth g. In particular, for every prime power d there exists a $(g, \gamma = 4/3)$-Moorish family of edge-transitive graphs, where* $\mathrm{Im}(g) = \{6, 8, \ldots\}$.

Intuitively, in an edge-transitive graph the local environment of every edge (i.e., the vertices and edges adjacent to it) looks the same. More formally, an edge-transitive graph is one in which any two edges are equivalent under some element of its automorphism group. As the computation of the effective resistance is not affected by an automorphism, we can conclude the following claim.

▷ Claim 16. Let $G = (V, E)$ be an unweighted edge-transitive graph. Then, for every two edges $e = (a, b)$ and $e' = (a', b')$ in E it holds that $R_{ab} = R_{a'b'}$.

4.2 The Lower Bound Proof

We next prove our main result for this section, showing that Moorish edge-transitive graphs cannot be sparsified via bounded-independence edge sampling when k is too small. Our proof can be seen as an extension of an argument by Alon and Nussboim [4], who studied the bounded independence relaxation of the usual Erdős-Rényi random graph model, where it is only required that the distribution of any subset of k edges is independent. They provide upper and lower bounds on the minimal k required to maintain properties that are satisfied by a truly random graph, and in particular they show that there exists a pairwise independent distribution $\mathcal{D}$ over edges with marginals $1/2$ such that a random graph sampled from $\mathcal{D}$ is disconnected almost surely.

As a warm-up, we extend the argument in [4] and show that *3-wise* independence also does not suffice, even for the special case of sparsifying the complete graph.

▶ **Lemma 17.** *Let $G = (V = [n], E)$ be the complete graph. There exists a distribution $\mathcal{D} \sim \{0,1\}^{|E|}$ that is 3-wise independent with marginals $1/4$ such that sampling a subgraph of G according to $\mathcal{D}$ would result in a disconnected graph with probability at least $1 - 2/2^n$.*

Proof. We first set some notations. Let $\mathcal{G}(A, p)$ be the usual Erdős-Rényi model, in which each edge between two vertices in A is included in the graph with probability p. Let $\mathcal{B}(A)$ be the natural distribution over complete bipartite graphs: Choose a partition $A = A_1 \uplus A_2$ uniformly at random and include all edges between A_1 and A_2.

We construct $\mathcal{D} \sim \{0,1\}^{|E|}$ as follows. Choose a partition $[n] = V_0 \uplus V_1$ uniformly at random. On V_0, draw a graph from $\mathcal{G}(V_0, 1/2)$ and on V_1, draw a graph from $\mathcal{B}(V_1)$. Clearly, sampling G' according to $\mathcal{D}$ would result in a disconnected graph unless $V_0 = \varnothing$ or $V_1 = \varnothing$, which occurs with probability at most $2/2^n$, so what is left to show is that $\mathcal{D}$ is 3-wise independent with marginals $1/4$.

This is equivalent to saying that for every $T \subseteq E$ with $|T| \leq 3$, $\Pr[\forall e \in T,\ \mathcal{D}(e) = 1] = 1/4^{|T|}$. Let us first consider the case $|T| = 1$, i.e. $T = e$ for a single edge $e \in E$. Notice that $\mathcal{D}(e) = 1$ only if both endpoints of e appear in the same side of the partition $V_0 \uplus V_1$, which occurs with probability $1/2$, and given that this occurs, e appears in $\mathcal{G}(V_0, 1/2)$ or $\mathcal{B}(V_1)$ with probability $1/2$. Thus, $\Pr[\mathcal{D}(e) = 1] = 1/4$, as desired.

Next, fix a set $T \subseteq E$ of $t \in \{2, 3\}$ edges and note that we can assume without loss of generality that these edges form either a path or a triangle (for $t = 3$), as disjoint paths will occur independently. If T forms a path, then similarly,

$$\Pr[T \in \mathcal{D}^{-1}(1)] = \Pr[V(T) \subseteq V_0] \cdot 2^{-t} + \Pr[V(T) \subseteq V_1] \cdot 2^{-t} = 2^{-(t+1)} \cdot 2^{-t} + 2^{-(t+1)} \cdot 2^{-t} = 4^{-t},$$

which is what we want. If T forms a triangle, then using the fact that a bipartite graph is triangle-free,

$$\Pr[T \in \mathcal{D}^{-1}(1)] = \Pr[V(T) \subseteq V_0] \cdot \frac{1}{8} = 4^{-3},$$

concluding the proof. ◀

The above lemma shows that one cannot sparsify the complete graph via $(k = 3)$-wise independent edge sampling. For a general k, we indeed need to resort to Moore-like graphs.

Proof of Theorem 11. Recalling that $k = \lfloor 4/3\alpha \rfloor$, let $g = k + 1$ or $g = k + 2$, whichever is even. Set d_0 to be the first prime power larger than

$$1 + \max\left\{2^{6^2/\alpha^8}, (2c)^{6/\alpha^2}\right\}.$$

By Theorem 15, for every prime power $d \geq d_0$ there exists $n = n(g, d)$ and a girth-g, edge-transitive, d-regular graph graph $G = (V = [n], E)$ (note that by our assumption that $\alpha \leq 4/15$, indeed $g \geq 6$). From here onwards, fix such a d and $n = n(g, d)$, observing that $\{n(g, d)\}_{d \geq d_0}$ is infinite.

Choose α_0 so that $c \log n \cdot n^{\alpha_0} = d/2$ so that marginals $1/2$ correspond to expected sparsity s. Using the fact that $d \geq n^{4/3g} + 1$, it can be verified that $\alpha_0 \geq (1 - 2\alpha) \cdot \alpha$. Using the fact that $d \geq n^{4/3g} + 1$,

$$\alpha_0 \geq \frac{4}{3g} - \frac{\log(2c \log n)}{\log n} \geq \frac{4}{\frac{4}{\alpha} + 6} - \frac{\log(2c \log n)}{\log n} \geq \left(1 - \frac{3\alpha}{2}\right)\alpha - \frac{\log(2c \log n)}{\log n}.$$

As $n \leq (d-1)^{\frac{3g}{4}}$ and $n \geq 2 \cdot \frac{(d-1)^{\frac{g}{2}} - 1}{d-2} \geq (d-1)^{\frac{g}{2}-1}$, the latter being the Moore bound, we have

$$\frac{\log(2c \log n)}{\log n} \leq \frac{\log(2c) + \log \frac{3g}{4} + \log\log(d-1)}{\left(\frac{g}{2} - 1\right)\log(d-1)} \leq \frac{\log(2c)}{\log(d-1)} + 2 \cdot \frac{\log\log(d-1)}{\log(d-1)} \leq \frac{\alpha^2}{2},$$

where we used $\frac{\log(2c)}{\log(d-1)} \leq \frac{\alpha^2}{6}$, $\log \frac{3g}{4} \leq \frac{g}{2} - 1$ and $\frac{\log\log(d-1)}{\log(d-1)} \leq \frac{\alpha^2}{6}$. Thus, overall, $\alpha_0 \geq (1 - 2\alpha)\alpha$.

We now give a k-wise independent distribution with marginals $1/2$ that fails to yield a good spectral sparsifier for G, namely it will be disconnected with high probability.

To do so, construct $\mathcal{D} \sim \{0,1\}^{|E|}$ as follows. Choose a partition $[n] = V_0 \uplus V_1$ uniformly at random. Each random partition gives rise to an element $D \sim \mathcal{D}$ in which for $e = (u, v) \in E$, $D(e) = 1$ (i.e., the edge e is chosen to survive) if and only if either $u, v \in V_0$ or $u, v \in V_1$.

▷ Claim 18. The distribution $\mathcal{D}$ is k-wise independent with marginals $1/2$.

Proof. As in the proof of Lemma 17, it suffices to show that for every set $T \subseteq E$ of $t \leq k$ edges of G we have $\Pr[T \subseteq \mathcal{D}^{-1}(1)] = 2^{-t}$. First, similar to Lemma 17, note that we can assume without loss of generality that T is a connected component, since whenever T_1 and T_2 are over disjoint sets of vertices, $\Pr[T_1 \cup T_2 \subseteq \mathcal{D}^{-1}(1)] = \Pr[T_1 \subseteq \mathcal{D}^{-1}(1)] \cdot \Pr[T_2 \subseteq \mathcal{D}^{-1}(1)]$. As the girth of G is larger than t, it must be the case that A is a tree.

In such a case, where T contains no cycles, $\Pr[T \in \mathcal{D}]$ is equal to the probability that all $t + 1$ vertices in T belong to the same partition, which is $2 \cdot 2^{-(t+1)} = 2^{-t}$. ◁

By the way $\mathcal{D}$ was constructed, it is clear that sampling G' according to $\mathcal{D}$ would result in a disconnected graph unless $V_0 = \varnothing$ or $V_1 = \varnothing$, which occurs with probability $1 - 2/2^n$, meaning that G' almost surely does not ε-approximate G, for *any* ε. ◀

We again stress that by the work in Section 3, we know that *any* k-wise independent distribution over the edges of G with marginals $s \cdot (n-1)/|E|$ for $k = \lceil 2/\alpha \rceil$ would produce an ε-spectral sparsifier with expected degree $O(s)$ with high constant probability.

The above also implies that any improvement upon Moorish families of edge-transitive graphs will improve our lower bound. Assuming the existence of a $(g, \gamma = 2)$-Moorish family of edge-transitive graphs we are able to show that the result of Section 3 is essentially tight.

5 Spectral Sparsifiers in Deterministic Small Space

In this section we show that Sparsify can be derandomized space efficiently.

▶ **Theorem 19** (deterministic small-space sparsification)**.** *Let G be an undirected, connected, weighted graph on n vertices with Laplacian L. There is a deterministic algorithm that, when given G, an even integer k and $0 < \varepsilon < 1$ outputs a weighted graph H with Laplacian $\widetilde{L}$ satisfying:*

1. *$\widetilde{L} \approx_\varepsilon L$, and,*
2. *H has $O\left(\frac{\log n}{\varepsilon^2} n^{1+2/k}\right)$ edges.*

The algorithm runs in space $O(k \log(N \cdot w) + \log(N \cdot w) \log\log(N \cdot w))$, where $w = w_{\max}/w_{\min}$ is the ratio of the maximum and minimum edge weights in G and N is the bitlength of the input.

We use the standard model of space-bounded computation. The machine has a read-only input tape, a constant number of read/write work tapes, and a write-only output tape. We say the machine runs in space s if throughout the computation, it only uses s total tape cells on the work tapes. The machine may write outputs to the output tape that are larger than s (in fact as large as $2^{O(s)}$) but the output tape is write-only. We use the following fact about the composition of space-bounded algorithms.

▶ **Lemma 20.** *Let f_1 and f_2 be functions that can be computed in space $s_1(n), s_2(n) \geq \log n$, respectively, and f_2 has output of length $\ell_1(n)$ on inputs of size n. Then $f_2 \circ f_1$ can be computed in space*

$$O(s_2(\ell_1(n)) + s_1(n)).$$

The natural way to derandomize Sparsify would be to iterate over all elements of the corresponding k-wise independent sample space. More formally, given $\{p_{ab}\}_{(a,b)\in E}$, let I_{ab} be the indicator random variable that is 1 if and only if edge (a, b) is chosen. If the I_{ab}'s are k-wise independent so that $\Pr[I_{ab} = 1] = p_{ab}$ (or some good approximation of p_{ab}), we are guaranteed to succeed with nonzero probability. Hence, at least one assignment to the I_{ab}'s taken from the k-wise independent is guaranteed to work. From Section 2.4 we know the sample space is small enough that we can afford to enumerate over all elements in it. Towards proving Theorem 19, there are still three issues to consider:

1. Approximating the effective resistances R_{ab} for every $(a, b) \in E$, space efficiently. Fortunately, we can do this with high accuracy using the result of Murtagh, Reingold, Sidford, and Vadhan [41] for approximating the pseudoinverse of a Laplacian, which we state shortly.

2. Verifying that a given set of random choices in Sparsify provides a sparse and accurate approximation to the input graph. The sparsity requirement is easy to check. To check that $\widetilde{L} \approx_\varepsilon L$, we devise a verification algorithm that uses the algorithm of [41]. The details are given in Lemma 25.
3. The Laplacian solver of [41] only works for multigraphs (graphs with integer edge weights) and we want an algorithm that works for general weighted graphs. To fix this, we extend the work of [41] by giving a simple reduction from the weighted case to the multigraph case. The details can be found in the full version of the paper.

5.1 Algorithm for Approximating Effective Resistances

A key ingredient in our deterministic sparsification algorithm is a deterministic nearly logarithmic space algorithm for approximating the pseudoinverse of an undirected Laplacian.

▶ **Theorem 21** ([41]). *Given an undirected, connected multigraph G with Laplacian $L = D - A$ and $\varepsilon > 0$, there is a deterministic algorithm that computes a symmetric PSD matrix $\widetilde{L^+}$ such that $\widetilde{L^+} \approx_\varepsilon L^+$, and uses space $O(\log N \cdot \log\log \frac{N}{\varepsilon})$, where N is the bitlength of the input (as a list of edges).*

Note that the space complexity above assumes that the multigraph is given as a list of edges. If we instead think of parallel edges as integer edge weights, then N should be replaced by $N \cdot w_{\max}$, where $w_{\max}$ is the maximum edge weight in G since an edge of weight w gets repeated w times in the edge-list representation. To work with general weighted graphs, we extend the result of [41].

▶ **Lemma 22** (small space laplacian solver for weighted graphs). *Given an undirected connected weighted graph $G = (V, E, w)$ with Laplacian $L = D - A$, and $0 < \varepsilon < 1$, there exists a deterministic algorithm that computes a symmetric PSD matrix $\widetilde{L^+}$ such that $\widetilde{L^+} \approx_\varepsilon L^+$, and uses space $O(\log(N \cdot w) \log\log(N \cdot w/\varepsilon))$, where $w = w_{\max}/w_{\min}$ is the ratio of the maximum and minimum edge weights in G and N is the bitlength of the input.*

A proof of Lemma 22 can be found in the full version. Lemma 22 immediately gives an algorithm for computing strong multiplicative approximations to effective resistances.

▶ **Lemma 23.** *Let $G = (V, E, w)$ be an undirected, connected, weighted graph and let R_{ab} be the effective resistance of $(a, b) \in E$. There is an algorithm that computes a real number $\widetilde{R}_{ab}$ such that*

$$(1 - \varepsilon) \cdot R_{ab} \leq \widetilde{R}_{ab} \leq (1 + \varepsilon) \cdot R_{ab}$$

and uses space $O(\log(N \cdot w) \cdot \log\log \frac{N \cdot w}{\varepsilon})$, where $w = w_{\max}/w_{\min}$ is the ratio of the maximum and minimum edge weights in G and N is the bitlength of the input.

See the full version for a proof of Lemma 23.

Next we show how we test whether two matrices spectrally approximate each other. We will need the following claim about the space complexity of matrix multiplication.

▷ Claim 24. Given $n \times n$ matrices $M_1, \ldots, M_k$, their product $M_1 \cdot \ldots \cdot M_k$ can be computed using $O(\log N \cdot \log k)$ space, where N is the bitlength of $(M_1, \ldots, M_k)$.

The proof of Claim 24 uses the natural divide and conquer algorithm and the fact that two matrices can be multiplied in logarithmic space. A detailed proof can be found in [41].

5.2 Testing for Spectral Proximity

In this section we give our deterministic, small-space procedure for verifying that two Laplacians spectrally approximate one another.

▶ **Lemma 25.** *There exists a deterministic algorithm that, given undirected, connected, weighted graphs $\widetilde{G}$ and G with Laplacians $\widetilde{L}, L$, and $\varepsilon, \alpha > 0$, outputs* `YES` *or* `NO` *such that*

1. *If $\widetilde{L} \approx_\varepsilon L$, then the algorithm outputs* `YES`*, and,*
2. *If $\widetilde{L} \not\approx_{\varepsilon\cdot\sqrt{1+\alpha}} L$ then the algorithm outputs* `NO`*.*

The algorithm uses space $O(\log(N \cdot w) \cdot \log\log \frac{N\cdot w}{\alpha\varepsilon} + \log(N \cdot w) \cdot \log \frac{1}{\alpha})$, where $w = w_{\max}/w_{\min}$ is the ratio of the maximum and minimum edge weights in G and $\widetilde{G}$ and N is the bitlength of the input.

The high level idea for the proof is that testing whether two matrices L and $\widetilde{L}$ spectrally approximate each other can be reduced to approximating the spectral radius of a particular matrix

$$M = \left(\frac{(\widetilde{L} - L)L^+}{\varepsilon}\right)^2.$$

In fact, it will be sufficient to check whether the trace of a sufficiently high power of M is below a certain threshold to deduce whether the spectral radius of M does not exceed 1. For intuition, replace the matrices with scalars m, ℓ, and $\widetilde{\ell}$ where

$$m = \frac{(\widetilde{\ell} - \ell)^2}{(\varepsilon \cdot \ell)^2}.$$

Then, $m \leq 1$ implies $\sqrt{m} \leq 1$, which implies $|\widetilde{\ell} - \ell| \leq \varepsilon \cdot \ell$ – the kind of relative closeness we want between the matrices $\widetilde{L}$ and L when aiming for spectral approximation. The proof of Lemma 25 can be found in the full version.

5.3 Completing the Proof of Theorem 19

We can now prove the main result of this section. As noted above, the algorithm proceeds by first approximating the sampling probabilities and then sparsifying G where the surviving edges are chosen from a small k-wise independent sample space whose marginals are set properly. Each potential sparsifier is checked using the algorithm given in Section 5.2.

Proof of Theorem 19. Set $\delta = \frac{1}{4}$, $\widehat{\varepsilon} = \frac{4\varepsilon}{5}$ and

$$s = \frac{18e \log n}{\widehat{\varepsilon}^2} \cdot \left(\frac{n}{\delta}\right)^{2/k},$$

for α soon to be determined. These parameters are chosen in accordance with the parameters required for Sparsify to succeed with probability $1/2$ and approximation error $\widehat{\varepsilon}$ (see Lemma 10). Set $\alpha' = \alpha/(4+\alpha)$. We compute approximate effective resistances $\widetilde{R}_{ab}$ for each edge (a, b) in G using Lemma 23, so that

$$(1 - \alpha')R_{ab} \leq \widetilde{R}_{ab} \leq (1 + \alpha')R_{ab}.$$

This takes $O(\log(N \cdot w) \log\log((N \cdot w)/\alpha))$ space. Then, we compute approximate sampling probabilities as follows:

$$\widetilde{p}_{ab} = \alpha' \cdot \left\lfloor \frac{1}{\alpha'} \cdot \min\left\{1, w_{ab} \cdot \widetilde{R}_{ab} \cdot s/(1 - \alpha')\right\}\right\rfloor$$

That is, we truncate the required (approximate) sampling probabilities to $\log \frac{1}{\alpha'}$ bits of precision. In particular, denoting the precise sampling probabilities by $p^\star_{ab} = \min\{1, w_{ab} \cdot R_{ab} \cdot s\}$, we have

$$\begin{aligned}\min\{1, w_{ab} \cdot \widetilde{R}_{ab} \cdot s/(1-\alpha')\} - p^*_{ab} &\leq w_{ab} \cdot s \cdot R_{ab} \cdot \left(\frac{1+\alpha'}{1-\alpha'} - 1\right) \\ &= p^*_{ab} \cdot \frac{2\alpha'}{1-\alpha'} \leq \alpha/2\end{aligned}$$

Furthermore, we have an additional error of α' due to the truncation so $|\widetilde{p}_{ab} - p^\star_{ab}| \leq \alpha/2 + \alpha' \leq \alpha$.

We want to set α so that $\widetilde{p}_{ab}$ is a multiplicative approximation to p^*_{ab} for all $(a,b) \in E$, which requires α to be smaller than $\min_{(a,b)\in E)}\{p^*_{ab}\}$.

▷ Claim 26. Let $d_{\max}$ be the maximum weighted degree over all vertices in G. Then, for all $(a,b) \in E$, $p^\star_{ab} \geq 1/d_{\max}$.

Proof. Since $s > 1$ and $w_{ab} \geq 1$ (all edge weights are positive integers) we have $p^\star_{ab} \geq R_{ab}$. Let $\lambda_{\min}(C)$ denote the minimal nonzero eigenvalue of a matrix C. To lower bound R_{ab}, we use the variational characterization of eigenvalues and the definition of effective resistance to write

$$\begin{aligned}R_{ab} &= (e_a - e_b)^\top L^+ (e_a - e_b) \geq \lambda_{\min}(L^+) \cdot \|e_a - e_b\|^2 \\ &= \frac{2}{\|L\|} \geq \frac{1}{d_{\max}}.\end{aligned}$$

Note that we can indeed consider the minimal nonzero eigenvalue of L^+ because $e_a - e_b$ is perpendicular to the one-dimensional kernel of L (the all-ones vector). ◁

In light of the above, we can set α so that $1/\alpha = 2 \cdot d_{\max} = O(N \cdot w)$ and get a 1/2-multiplicative approximation to the sampling probabilities.

Now, consider the k-wise independent sample space $\mathcal{D} \subseteq \{0,1\}^{|E|}$ guaranteed to us by Lemma 8, substituting $t = \lceil \log(1/\alpha') \rceil$. By Lemma 8, each element of $\mathcal{D}$ can be sampled using

$$O(k \cdot \max\{\log(1/\alpha'), \log|E|\}) = O(k \cdot \log(N \cdot w))$$

space. For each element of $\mathcal{D}$, construct the corresponding sparse graph. Note that the space used to cycle through each element can be reused. Lemma 10 tells us that at least $1 - 2\delta = 1/2$ of the Laplacians of the resulting graphs $\widehat{\varepsilon}$-approximate the Laplacian of G and have

$$O\left(\frac{1+1/2}{1-1/2} \cdot \frac{1}{\delta^{1+2/k}} \cdot \frac{\log n}{\widehat{\varepsilon}^2} \cdot n^{1+\frac{2}{k}}\right) = O\left(\frac{\log n}{\varepsilon^2} \cdot n^{1+\frac{2}{k}}\right)$$

edges. For each of these graphs, we run the verification algorithm with accuracy parameter 9/16, which is guaranteed to find a graph with the above sparsity whose Laplacian approximates the Laplacian of G with error

$$\widehat{\varepsilon} \cdot \sqrt{1 + \frac{9}{16}} = \frac{4\varepsilon}{5} \cdot \sqrt{\frac{25}{16}} = \varepsilon$$

in space

$$O\left(\log(N \cdot w) \log\log \frac{16N \cdot w}{9\widehat{\varepsilon}} + \log(N \cdot w) \log \frac{16}{9}\right) = O\left(\log(N \cdot w) \log\log \frac{N \cdot w}{\varepsilon}\right).$$

Again, the space used for the verification process can be reused. Adding up the space complexities gives us a total of

$$O\left(k\log(N\cdot w)+\log(N\cdot w)\log\log\frac{N\cdot w}{\varepsilon}\right)$$

space. Note that the final result is vacuous when $\varepsilon \leq 1/n$ so we can without loss of generality assume that $\varepsilon \geq 1/n$. This gives a total space complexity of $O(k\log(N\cdot w)+\log(N\cdot w)\log\log(N\cdot w))$. ◀

References

1 AmirMahdi Ahmadinejad, Jonathan Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil Vadhan. High-precision estimation of random walks in small space. *arXiv preprint*, 2019. `arXiv:1912.04524`.

2 Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC 2015)*, pages 237–245. ACM, 2015.

3 Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of algorithms*, 7(4):567–583, 1986.

4 Noga Alon and Asaf Nussboim. k-wise independent random graphs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 813–822. IEEE, 2008.

5 Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Review*, 56(2):315–334, 2014.

6 András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\widetilde{O}(n^2)$ time. In *STOC*, volume 96, pages 47–55. Citeseer, 1996.

7 Norman L. Biggs and Alan G. Boshier. Note on the girth of Ramanujan graphs. *Journal of Combinatorial Theory, Series B*, 49(2):190–194, 1990.

8 Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

9 Richard Y. Chen, Alex Gittens, and Joel A. Tropp. The masked sample covariance estimator: An analysis using matrix concentration inequalities. *Information and Inference: A Journal of the IMA*, 1(1):2–20, 2012.

10 L. Paul Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.

11 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 *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC 2011)*, pages 273–282. ACM, 2011.

12 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 *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2018)*, pages 361–372. IEEE, 2018.

13 Michael B. Cohen, Jonathan Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B. Rao, and Aaron Sidford. Solving directed Laplacian systems in nearly-linear time through sparse LU factorizations. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2018)*, pages 898–909. IEEE, 2018.

14 Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for Markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, pages 410–419. ACM, 2017.

15 Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Aaron Sidford, and Adrian Vladu. Faster algorithms for computing the stationary distribution, simulating random walks, and more. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 583–592. IEEE, 2016.

16 Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup B. Rao, and Shen Chen Xu. Solving SDD linear systems in nearly $m \log^{1/2} n$ time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 343–352. ACM, 2014.

17 Michael B Cohen, Aleksander Madry, Piotr Sankowski, and Adrian Vladu. Negative-weight shortest paths and unit capacity minimum cost flow in $\widetilde{O}(m^{10/7} \log W)$ time. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, pages 752–771. SIAM, 2017.

18 Dean Doron, François Le Gall, and Amnon Ta-Shma. Probabilistic logarithmic-space algorithms for Laplacian solvers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, pages 41:1–41:20. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

19 Dean Doron, Jack Murtagh, Salil Vadhan, and David Zuckerman. Spectral sparsification via bounded-independence sampling. *Electronic Colloquium on Computational Complexity (ECCC)*, 2020.

20 Paul Erdős and Horst Sachs. Reguläre graphen gegebener taillenweite mit minimaler knotenzahl. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 12(251-257):22, 1963.

21 Geoffrey Exoo and Robert Jajcay. Dynamic cage survey. *Electron. J. Combin*, 15(16):4, 2008.

22 Ronald M. Foster. The average impedance of an electrical network. *Contributions to Applied Mechanics (Reissner Anniversary Volume)*, pages 333–340, 1949.

23 A. Joffe. On a set of almost deterministic k-independent random variables. *The Annals of Probability*, 2(1):161–162, 1974.

24 Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014)*, pages 217–226. SIAM, 2014.

25 Jonathan A. Kelner and Aleksander Madry. Faster generation of random spanning trees. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*, pages 13–21. IEEE, 2009.

26 Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 911–920. ACM, 2013.

27 Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly-$m \log n$ time solver for SDD linear systems. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2011)*, pages 590–598. IEEE, 2011.

28 Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD linear systems. *SIAM Journal on Computing*, 43(1):337–354, 2014.

29 Ioannis Koutis, Gary L. Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. In *International Symposium on Visual Computing*, pages 1067–1078. Springer, 2009.

30 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 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 842–850. ACM, 2016.

31 Rasmus Kyng and Sushant Sachdeva. Approximate Gaussian elimination for Laplacians-fast, sparse, and simple. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 573–582. IEEE, 2016.

32 Felix Lazebnik, Vasiliy A. Ustimenko, and Andrew J. Woldar. A new series of dense graphs of high girth. *Bulletin of the American mathematical society*, 32(1):73–79, 1995.

33 Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 755–764. ACM, 2013.

34 Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 147–156. IEEE, 2013.

35 Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. In *Proceedings of the 56th IEEE Annual Symposium on Foundations of Computer Science (FOCS 2015)*, pages 250–269. IEEE, 2015.

36 Yin Tat Lee and He Sun. An SDP-based algorithm for linear-sized spectral sparsification. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, pages 678–687. ACM, 2017.

37 Yang P. Liu and Aaron Sidford. Faster energy maximization for faster maximum flow. *arXiv preprint*, 2019. `arXiv:1910.14276`.

38 Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

39 Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, 1986.

40 Aleksander Madry, Damian Straszak, and Jakub Tarnawski. Fast generation of random spanning trees and the effective resistance metric. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014)*, pages 2019–2036. SIAM, 2014.

41 Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil P. Vadhan. Derandomization beyond connectivity: Undirected Laplacian systems in nearly logarithmic space. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS 2017)*, pages 801–812. IEEE, 2017.

42 Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil P. Vadhan. Deterministic approximation of random walks in small space. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, pages 42:1–42:22. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

43 Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K. Vishnoi. Approximating the exponential, the Lanczos method and an $\widetilde{O}(m)$-time spectral algorithm for balanced separator. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 1141–1160. ACM, 2012.

44 Richard Peng and Daniel A. Spielman. An efficient parallel solver for SDD linear systems. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 333–342. ACM, 2014.

45 Robert Qiu and Michael Wicks. *Cognitive networked sensing and big data*. Springer, 2014.

46 Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4), 2008.

47 Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 436–447. Springer, 2005.

48 Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21, 2007.

49 Aaron Schild. An almost-linear time algorithm for uniform random spanning tree generation. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 214–227. ACM, 2018.

50 Daniel A. Spielman. Yale Applied Mathematics 561/Computer Science 662, Lecture Notes: Spectral Graph Theory, November 2015. URL: `http://www.cs.yale.edu/homes/spielman/561/lect17-15.pdf`.

51 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

52 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 81–90, 2004.

53 Anastasios Zouzias. A matrix hyperbolic cosine algorithm and applications. In *International Colloquium on Automata, Languages, and Programming (ICALP 2012)*, pages 846–858. Springer, 2012.