Eulerian Graph Sparsification by Effective Resistance Decomposition

Arun Jambulapati
University of Michigan
jmblpati@gmail.com

Sushant Sachdeva
University of Toronto
sachdeva@cs.toronto.edu

Aaron Sidford
Stanford University
sidford@stanford.edu

Kevin Tian
University of Texas at Austin
kjtian@cs.utexas.edu

Yibin Zhao
University of Toronto
ybzhao@cs.toronto.edu

Abstract

We provide an algorithm that, given an n-vertex m-edge Eulerian graph with polynomially bounded weights, computes an $\breve{O}(n\log^2 n\cdot\varepsilon^{-2})$-edge ε-approximate Eulerian sparsifier with high probability in $\breve{O}(m\log^3 n)$ time (where $\breve{O}(\cdot)$ hides polyloglog(n) factors). Due to a reduction from [Peng-Song, STOC '22], this yields an $\breve{O}(m\log^3 n+n\log^6 n)$-time algorithm for solving n-vertex m-edge Eulerian Laplacian systems with polynomially-bounded weights with high probability, improving upon the previous state-of-the-art runtime of $\Omega(m\log^8 n+n\log^{23} n)$. We also give a polynomial-time algorithm that computes $O(\min(n\log n\cdot\varepsilon^{-2}+n\log^{5/3} n\cdot\varepsilon^{-4/3}, n\log^{3/2} n\cdot\varepsilon^{-2}))$-edge sparsifiers, improving the best such sparsity bound of $O(n\log^2 n\cdot\varepsilon^{-2}+n\log^{8/3} n\cdot\varepsilon^{-4/3})$ [Sachdeva-Thudi-Zhao, ICALP '24]. Finally, we show that our techniques extend to yield the first $O(m\cdot\text{polylog}(n))$ time algorithm for computing $O(n\varepsilon^{-1}\cdot\text{polylog}(n))$-edge graphical spectral sketches, as well as a natural Eulerian generalization we introduce.

In contrast to prior Eulerian graph sparsification algorithms which used either short cycle or expander decompositions, our algorithms use a simple efficient effective resistance decomposition scheme we introduce. Our algorithms apply a natural sampling scheme and electrical routing (to achieve degree balance) to such decompositions. Our analysis leverages new asymmetric variance bounds specialized to Eulerian Laplacians and tools from discrepancy theory.

Contents

1 Introduction

Over the past decade, ideas from spectral graph theory have led to a revolution in graph algorithms. A major frontier for such developments is the design of spectral algorithms for directed graphs. Such algorithms have wide-ranging applications from fast algorithms for processing Markov chains (see e.g., [CKPPSV16; AJSS19]) to deterministic low-space computation (see e.g., [AKMPSV20]). A fundamental challenge in this setting is the fairly involved machinery used in spectral directed graph algorithms, which include efficient constructions of expander decompositions [CKPPRSV17] and short cycle decompositions [CGPSSW18]. In this paper we focus on the central topic of *spectral sparsification of directed graphs*, for which, this challenge is particularly manifest.

A sparsifier of an undirected graph $G = (V, E, \mathbf{w})$ or directed graph $\vec{G}$ is another graph supported on the same set of vertices with fewer edges, that approximately preserves some property. Several notions of sparsification for undirected graphs have been studied in the literature, e.g., spanners [BS03; TZ05], which approximately preserve shortest path distances, and cut sparsifiers [BK96], which approximately preserve cut sizes. Spectral sparsification [ST04] has been particularly influential in the design of graph algorithms. An ε-approximate undirected spectral sparsifier (henceforth, *ε-approximate undirected sparsifier*) $H = (V, E', \mathbf{w}')$ of undirected G approximately preserves the quadratic form of G's graph Laplacian, i.e., for all $\mathbf{x} \in \mathbb{R}^V$,

$$(1-\varepsilon)\mathbf{x}^\top \mathbf{L}_G \mathbf{x} \leq \mathbf{x}^\top \mathbf{L}_H \mathbf{x} \leq (1+\varepsilon)\mathbf{x}^\top \mathbf{L}_G \mathbf{x}, \text{ where } \mathbf{x}^\top \mathbf{L}_G \mathbf{x} = \sum_{e=(u,v)\in E} \mathbf{w}_e (\mathbf{x}_u - \mathbf{x}_v)^2, \tag{1}$$

where $\mathbf{L}_G$ and $\mathbf{L}_H$ are the undirected Laplacian matrices of G and H (see Section 2 for notation), and (1) is equivalent to $(1-\varepsilon)\mathbf{L}_G \preceq \mathbf{L}_H \preceq (1+\varepsilon)\mathbf{L}_G$. Spectral sparsification generalizes cut sparsification and was key to the advent of nearly-linear time Laplacian systems solvers [ST04].

Simple and efficient algorithms for computing undirected spectral sparsifiers with nearly-optimal guarantees are known. Spielman and Srivastava [SS11] showed that independently sampling (and reweighting) $O(n\varepsilon^{-2}\log n)$ edges of an n-vertex graph, with probability proportional to their *effective resistances* (a graph-theoretic analog of leverage scores), produces a spectral sparsifier. All effective resistances can be estimated in $\breve{O}(m \log n)$ time[1] using fast Laplacian system solvers [JS21] (see Lemma 13) – this step dominates the runtime for undirected spectral sparsification. Additionally, Batson, Spielman, and Srivastava [BSS12] showed spectral sparsifiers with $O(n\varepsilon^{-2})$ edges exist, which is optimal [BSS12; CKST19] and constructible in near-linear time [LS17; JRT23].

Obtaining correspondingly simple and fast sparsification algorithms and optimal sparsity bounds for directed graphs remains elusive. Even proposing useful notions of directed sparsification was challenging; any sparsifier of the complete, directed, bipartite graph, i.e., the graph with a directed edge from every node in one side of the bipartition to the other, that approximately preserves all directed cuts cannot delete any edges. The influential work [CKPPRSV17] overcame this bottleneck by restricting their attention to directed Eulerian graphs (where every vertex has equal weighted in-degree and out-degree). Further, [CKPPRSV17] showed that their sparsification notion suffices for numerous applications, including fast solvers for all directed Laplacian linear systems (not necessarily corresponding to an Eulerian graph), overviewed in Section 8. In this paper, we consider the following definition of Eulerian sparsification closely related to that of [CKPPRSV17].[2]

[1] When discussing a graph clear from context with n vertices and edge weight ratio bounded by U, we use the $\breve{O}$ notation to hide polyloglog(nU) factors for brevity (in runtimes only).

[2] The key difference is that we add the $E(\vec{H}) \subseteq E$ restriction.

Definition 1 (Eulerian sparsifier)**.** *$\vec{H}$ is an ε-*approximate Eulerian sparsifier *of $\vec{G} = (V, E, \mathbf{w})$ if $\vec{H}$ and $\vec{G}$ are both Eulerian, $V(\vec{H}) = V$, and for $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$, we have*

$$\left\| \mathbf{L}_G^{\dagger\frac{1}{2}} \left(\vec{\mathbf{L}}_{\vec{G}} - \vec{\mathbf{L}}_{\vec{H}} \right) \mathbf{L}_G^{\dagger\frac{1}{2}} \right\|_{\text{op}} \leq \varepsilon, \text{ and } E(\vec{H}) \subseteq E. \tag{2}$$

Definition 1 generalizes the notion of undirected sparsification (Fact 7). While useful in applications, Definition 1 poses computational challenges. Eulerian sparsifiers preserve exact degree balance, so in contrast to undirected sparsifiers, one cannot simply sample edges independently to compute sparsifiers. There have been two broad approaches for addressing this key challenge.

The first approach leverages *expander decompositions* and is related to one used in [ST04] to sparsify undirected graphs. [CKPPRSV17] followed such an approach and their algorithm consists of decomposing the Eulerian graph $\vec{G}$ into expanders, sampling edges independently inside the expanders, and then fixing the resulting degree imbalance by adding edges; this resulted in sparsifiers that did not necessarily satisfy the $E(\vec{H}) \subseteq E$ property in (2). This approach was refined in [APPSV23] (using cycle decompositions as in the second approach below, but not necessarily short ones), resulting in an algorithm for constructing Eulerian sparsifiers with $O(n\varepsilon^{-2}\log^{20} n)$ edges in $O(m\log^7 n)$ time. Existing near-linear time expander decomposition methods [SW19; ADK23] incur several logarithmic factors in the running time and (inverse) expansion quality, leading to these large, difficult to improve, polylogarithmic factors in the running time and sparsity.

The second approach leverages that most the edges in $\vec{G}$ can be decomposed into edge-disjoint short cycles, termed a *short cycle decomposition*. [CGPSSW18] pioneered this approach and sampled the edges in a coordinated manner within each cycle to preserve degree balance. Advances in short cycle decompositions [LSY19; PY19; STZ24] resulted in an $m^{1+o(1)}$-time algorithm for constructing Eulerian sparsifiers with $O(n\varepsilon^{-2}\log^3 n)$ edges. Short cycle decompositions yield Eulerian sparsifier constructions with significantly improved sparsity compared to the expander decomposition approach, at the cost of large $m^{o(1)}$ factors in running time.

In summary, all prior algorithms for constructing Eulerian sparsifiers use either expander decomposition or short cycle decomposition, which result in substantial polylogarithmic factors (or larger) in sparsities and runtimes. More broadly, large gaps seem to remain in our understanding of efficient algorithms for constructing Eulerian sparsifiers and the optimal sparsity achievable.

1.1 Our results

We present a new sparsification framework that allows one to preserve exact degree balance while sampling, as in Eulerian sparsification, and yet analyze the sampling error as if the edges were sampled independently. Our framework is simple and intuitive, as it is based on randomly signing multiplicative reweightings to edges, and using electrical flows to fix the degree balance. Combining our framework with a lightweight graph-theoretic construction, *effective resistance decomposition* (Definition 9), we obtain the following Eulerian sparsification result.

Theorem 2. *Given Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $|V| = n$, $|E| = m$, integral $\mathbf{w} \in [1, \text{poly}(n)]^E$ and $\varepsilon \in (0, 1)$,* FastSparsify *(Algorithm 7) in $\breve{O}\left(m\log^3 n\right)$ time returns Eulerian $\vec{H}$ that w.h.p.,[3] is an ε-approximate Eulerian sparsifier of $\vec{G}$ with $|E(\vec{H})| = O\left(n\varepsilon^{-2}\log^2(n)\log^2\log(n)\right)$.*

[3] In the introduction only, we use the abbreviation "w.h.p." ("with high probability") to mean that a statement holds with n^{-C} failure probability for an arbitrarily large constant C (which affects other constants in the statement). In the formal variants of theorem statements later in the paper, we state precise dependences on failure probabilities.

Theorem 2 constructs Eulerian sparsifiers with sparsity within a $\breve{O}(\log^2 n)$ factor of optimal [CKST19], in time $\breve{O}(m \log^3 n)$. Our algorithm simultaneously achieves a substantially faster runtime than prior Eulerian sparsification schemes and improves the state-of-the-art sparsity bound (see Table 1). For instance, the prior state-of-the-art Eulerian sparsification algorithm with both $O(n\varepsilon^{-2} \cdot \text{polylog}(n))$ edges and a $O(m \cdot \text{polylog}(n))$ runtime has (up to $O(\text{poly} \log \log n)$) factors an extra $\Omega(\log^{18} n)$ factor in sparsity and an $\Omega(\log^4 n)$ factor in the runtime compared to Theorem 2.

As a corollary of our fast sparsification algorithm (Theorem 2), reductions due to Peng and Song [PS22] and earlier works on solving (variants of) directed Laplacian systems [CKPPSV16; CKPPRSV17; AJSS19], we obtain a host of additional results. The following is a straightforward corollary obtained by a direct reduction given in the main result of [PS22].

Corollary 3 (Eulerian Laplacian solver)**.** *There is an algorithm which given input Eulerian* $\vec{G} = (V, E, \mathbf{w})$ *with* $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, \text{poly}(n)]^E$, *and* $\mathbf{b} \in \mathbb{R}^V$, *in* $\breve{O}\left(m \log^3(n) + n \log^6(n)\right)$ *time returns* $\mathbf{x} \in \mathbb{R}^V$ *satisfying, w.h.p.,* $\|\mathbf{x} - \vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\|_{\mathbf{L}_G} \leq \varepsilon \|\vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\|_{\mathbf{L}_G}$ *for* $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$.

The runtime of Corollary 3 improves upon the prior state-of-the-art claimed in the literature of $\breve{O}(m \log^8 n + n \log^{23} n)$ (see Appendix C, [PS22]). Up to small polylogarithmic factor overheads in runtimes, our Eulerian Laplacian solver also implies a solver for all directed Laplacians (Corollary 43), and fast high-accuracy approximations for directed graph primitives such as computation of stationary distributions, mixing times, Personalized PageRank vectors, etc., as observed by [CKPPSV16; AJSS19]. We state these additional applications in Section 8.

We further ask: what is the optimal number of edges in an Eulerian sparsifier? By combining our new approach with recent advances in discrepancy theory due to Bansal, Jiang, and Meka [BJM23], we obtain the following improved sparsity bound over Theorem 2.

Theorem 4. *Given Eulerian* $\vec{G} = (V, E, \mathbf{w})$ *with* $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, \text{poly}(n)]^E$ *and* $\varepsilon \in (0, 1)$, ExistentialSparsify *(Algorithm 3) in* $\text{poly}(n, \varepsilon^{-1})$ *time returns Eulerian* $\vec{H}$ *such that w.h.p.* $\vec{H}$ *is an* ε*-approximate Eulerian sparsifier of* $\vec{G}$ *with*

$$|E(\vec{H})| = O\left(\min\left\{\frac{n \log n}{\varepsilon^2} + \frac{n \log^{5/3} n}{\varepsilon^{4/3}}, \frac{n \log^{3/2} n}{\varepsilon^2}\right\}\right).$$

For $\varepsilon \leq \log^{-1} n$, Theorem 4 establishes that $O(n\varepsilon^{-2} \log n)$-edge Eulerian sparsifiers exist and are constructible in polynomial time. Moreover for any ε, the sparsity is at most $n\varepsilon^{-2} \log^{\frac{3}{2}} n$. In Appendix C, we discuss potential directions towards showing the existence of even sparser Eulerian sparsifiers, e.g., with only $O(n\varepsilon^{-2})$ nonzero edge weights (matching the optimal sparsity for undirected graph sparsifiers [BSS12; CKST19]).

We further demonstrate the power of our framework by giving an efficient construction of graphical spectral sketches [ACKQWZ16; JS18; CGPSSW18], i.e., sparse graphs which satisfy (1) for any fixed vector $\mathbf{x} \in \mathbb{R}^V$ w.h.p. (rather than for all $\mathbf{x} \in \mathbb{R}^V$). The only previously known construction of graphical spectral sketches was based on short cycle decompositions [CGPSSW18; LSY19; PY19]. We provide an algorithm that efficiently computes sparse weighted subgraphs that are simultaneously graphical spectral sketches, spectral sparsifiers (for a larger value of ε), and sketches of the pseudoinverse in a suitable sense.

Theorem 5. *There is an algorithm that, given undirected graph* $G = (V, E, \mathbf{w})$ *with* $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, \text{poly}(n)]^E$ *and* $\varepsilon \in (0, \frac{1}{100})$, *in* $\breve{O}\left(m \log^9(n)\right)$ *time returns an undirected graph* H *such that* $|E(H)| = O\left(n\varepsilon^{-1} \log^9(n) \log^2 \log(n)\right)$ *and the following properties hold.*

1. H *is a* $\sqrt{\varepsilon}$*-approximate spectral sparsifier of* G *w.h.p.*

Method	Sparsity	Runtime	Approach
[CKPPRSV17]	$n\varepsilon^{-2}\log^{C} n$	$m\log^{C} n$	expanders
[CGPSSW18]	$n\varepsilon^{-2}\log^{4} n$	mn	short cycles
[CGPSSW18; LSY19; PY19]	$n\varepsilon^{-2}\log^{k} n$	$m + n^{1+O(\frac{1}{k})}$	short cycles
[PY19]	$n^{1+o(1)} + n\varepsilon^{-2}\log^{4} n$	$m\log^{C} n$	short cycles
[APPSV23]	$n\varepsilon^{-2}\log^{12} n$	existential	SV sparsification
[APPSV23]	$n\varepsilon^{-2}\log^{20} n$	$m\log^{7} n$	SV sparsification
[PY19; STZ24]	$n\varepsilon^{-2}\log^{3} n$	$m^{1+\delta}$	short cycles
[STZ24]	$n\varepsilon^{-2}\log^{2} n + n\varepsilon^{-4/3}\log^{8/3} n$	n^{C}	short cycles
Theorem 2	$n\varepsilon^{-2}\log^{2} n$	$m\log^{3} n$	ER decomposition
Theorem 4	$n\varepsilon^{-2}\log n + n\varepsilon^{-4/3}\log^{5/3} n$	n^{C}	ER decomposition
Theorem 4	$n\varepsilon^{-2}\log^{3/2} n$	n^{C}	ER decomposition

Table 1: **Eulerian sparsification algorithms.** All results apply to Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $n \stackrel{\text{def}}{=} |V|$ and $m \stackrel{\text{def}}{=} |E|$. For simplicity, $\mathbf{w} \in [1, \text{poly}(n)]^E$ and all algorithms fail with probability $\text{poly}(\frac{1}{n})$. C denotes an unspecified (large) constant, δ denotes an arbitrarily small constant, and we hide polyloglog(n) factors. The third row requires $k \geq 4$. The [CKPPRSV17] sparsifiers were not reweighted subgraphs of the original graph, but all other sparsifiers in this table are.

2. *H is a ε-approximate graphical sketch of G, i.e., for an arbitrarily fixed vector $\mathbf{x} \in \mathbb{R}^V$, w.h.p. over H, $\left|\mathbf{x}^\top(\mathbf{L}_H - \mathbf{L}_G)\mathbf{x}\right| \leq \varepsilon \cdot \mathbf{x}^\top \mathbf{L}_H \mathbf{x}$.*

3. *H is a ε-approximate inverse sketch of G, i.e., for an arbitrarily fixed vector $\mathbf{x} \in \mathbb{R}^V$, w.h.p. over H, $\left|\mathbf{x}^\top(\mathbf{L}_H^\dagger - \mathbf{L}_G^\dagger)\mathbf{x}\right| \leq \varepsilon \cdot \mathbf{x}^\top \mathbf{L}_H^\dagger \mathbf{x}$.*

While this more general guarantee was also achieved by the short-cycle decomposition based constructions, the previous best construction of a graphical spectral sketch with $n\varepsilon^{-1} \cdot \text{polylog}(n)$ edges required $m^{1+o(1)}$ time [PY19]. Additionally, in Section 9 we generalize this notion of graphical spectral sketches to Eulerian graphs (Definition 45) and provide analogous runtimes and sparsity bounds for such sketches (Theorem 56); these are the first such results to the best of our knowledge.

1.2 Overview of approach

In this paper, we provide a new, simpler framework for sparsifying Eulerian graphs. Despite its simplicity, our approach yields Eulerian sparsification algorithms which improve upon prior work in both runtime and sparsity. We briefly overview our framework and technical contributions here; see Section 3 for a more detailed technical overview.

Our framework is motivated by the following simple undirected graph sparsification algorithm.

- For all edges $e \in E$ with an effective resistance (ER) smaller than ρ, toss an independent coin and either drop the edge or double its weight.
- Repeat until there are no edges left with a small ER.

It is straightforward to show that this algorithm produces a spectral sparsifier. In each iteration, the algorithm's relative change to the Laplacian (in a multiplicative sense) is $\sum_{e \in E} \mathbf{s}_e \mathbf{A}_e$, where $\mathbf{s}_e$ is a random ± 1 sign and $\mathbf{A}_e = \mathbf{w}_e \mathbf{L}_G^{\dagger/2} \mathbf{b}_e \mathbf{b}_e^\top \mathbf{L}_G^{\dagger/2}$ denotes the normalized contribution of the edge

Laplacian. The key step of the analysis is bounding the total matrix variance $\sum_{e\in E} \mathbf{A}_e\mathbf{A}_e^\top$, across all iterations. When setting $\rho = c\frac{n}{m}$ where m is the current number of edges and c is a sufficiently large constant, the variance contribution for each edge forms an increasing geometric progression (as m decreases geometrically) where the sum is bounded by the last term. Moreover, each edge Laplacian only contributes if its leverage score is at most ρ, so $\mathbf{A}_e\mathbf{A}_e^\top \preceq \rho\mathbf{A}_e$. Summing over all edges, the total matrix variance is $\preceq \rho\mathbf{I}$. Stopping when $\rho = O(\frac{\varepsilon^2}{\log n})$ for an appropriate constant, standard matrix concentration bounds then show the total relative spectral error is $O(\sqrt{\rho \log n}) \cdot \mathbf{I} = \varepsilon\mathbf{I}$.

Emulating such a strategy for Eulerian graphs faces an immediate obstacle: adding and dropping edges independently might result in a non-Eulerian graph, i.e., one that does not satisfy the degree balance constraints of an Eulerian graph. In fact, there may be no setting of $\mathbf{s} \in \{\pm 1\}^E$ for which the relative change in edge weights, $\mathbf{w} \circ \mathbf{s}$, satisfies the necessary degree balance. As mentioned previously, one approach to Eulerian sparsification [CKPPRSV17] independently samples ± 1 signs for edges inside an expander, fixes the resulting degree imbalance, and uses the expansion property to bound the resulting error. Another approach, based on short cycle decomposition [CGPSSW18], toggles cycles, keeping either only the clockwise or counterclockwise edges, thus ensuring degrees are preserved. Additionally, [APPSV23] samples ± 1 signs for cycles (not necessarily short) inside an expander. Each of these results in large polylogarithmic factors or worse in their guarantees, due to limitations in algorithms for expander or short-cycle decomposition.

To obtain faster and simpler algorithms with improved sparsity guarantees, we take an alternative approach. As a starting point, consider sampling a random signing $\mathbf{s}$ on edge Laplacians, and projecting $\mathbf{s}$ down to the degree balance-preserving subspace. We make the simple, yet crucial, observation: this projection step does not increase the matrix variance (Lemma 17)! This fact, which lets us bound spectral error as we would if all edge signings were independent, has not been exploited previously for efficient degree balance-preserving sparsification to our knowledge.

Our second key contribution is recognizing that to bound the variance of an independent edge Laplacian signing in a subgraph, requiring the subgraph to be an expander is stronger than necessary. In Lemma 19, we show it suffices to work in subgraphs with bounded ER diameter (implied by expansion in high-degree unweighted graphs, cf. Lemma 52). Decomposing a graph into low ER diameter pieces can be achieved more simply, efficiently, and with better parameters (for our purposes) as compared to expander or short cycle decompositions (Proposition 10).

To implement this approach to Eulerian sparsification efficiently, we overcome several additional technical hurdles. The first one is ensuring (in nearly-linear time) that the updated edge weight vector is nonnegative; negative weight edges could occur when projecting a large vector to the degree-preserving space. In previous discrepancy works, e.g., [Rot17], this problem was alleviated by projecting the random vector to the intersection of the subspace with the ± 1 hypercube. This projection is expensive; on graphs it could be implemented with oblivious routings, but unfortunately, the fastest routings of sufficient quality in the literature do not run in nearly-linear time. We show that by scaling down the step size by a polylogarithmic factor and appealing to sub-Gaussianity of random projection vectors, we can ensure the nonnegativity of weights.

Secondly, since the weight updates are small in magnitude, there is no immediate reduction in sparsity. Using a careful two-stage step size schedule (see discussion in Section 7), we give a potential argument showing that after adding roughly $\log^2(n)$ random signings, each projected by solving an undirected Laplacian system, suffices to make a constant fraction of the weights tiny. These tiny edge weights can then be rounded to zero, decreasing the sparsity by a constant factor. Combining our framework with state-of-the-art undirected Laplacian solvers gives our overall runtime of $\breve{O}(m \log^3(n))$ in Theorem 2.

1.3 Related work

Undirected sparsifiers and Laplacian solvers. The first nearly-linear time algorithm for solving undirected Laplacian linear systems was obtained in groundbreaking work of Spielman and Teng [ST04]. Since then, there has been significant work on developing faster undirected Laplacian solvers [KMP14; KMP11; PS14; CKMPPRX14; KLPSS16; KS16; JS21; FGLPSY22; SZ23], culminating in an algorithm that runs in $\breve{O}(m \log \frac{1}{\varepsilon})$ time for approximately solving undirected Laplacian linear systems up to expected relative error ε (see Proposition 12 for a formal statement).

The first spectral sparsifiers for undirected graphs were constructed by Spielman and Teng [ST04], which incurred significant polylogarithmic overhead in their sparsity. Spielman and Srivastava [SS11] then gave a simple algorithm for constructing undirected spectral sparsifiers with $O(n\varepsilon^{-2} \log n)$ edges in nearly-linear time. Batson, Spielman, and Srivastava [BSS12] gave a polynomial time algorithm for constructing undirected spectral sparsifiers with $O(n\varepsilon^{-2})$ edges, and established that this sparsity bound is optimal. Faster algorithms for $O(n\varepsilon^{-2})$-edge undirected sparsifiers were later given in [LS17; LS18; JRT23]. We also mention an additional notion of sparsification in undirected graphs, *degree-preserving sparsification*, which has been studied in the literature as an intermediary between undirected and Eulerian sparsification [CGPSSW18; JRT23]. Degree-preserving undirected sparsifiers of sparsity $O(n\varepsilon^{-2})$ were recently shown to exist and be constructible in almost-linear time by [JRT23], motivating our work in the related Eulerian sparsification setting.

Eulerian sparsifiers and directed Laplacian solvers. The study of efficient directed Laplacian solvers was initiated by Cohen, Kelner, Peebles, Peng, Sidford, and Vladu [CKPPSV16], who established that several computational problems related to random walks on directed graphs can be efficiently reduced to solving linear systems in Eulerian Laplacians. This work also gave an algorithm for solving Eulerian Laplacian linear systems in $O((mn^{2/3} + m^{3/4}n) \cdot \text{polylog}(n))$ time, the first such solver with a runtime faster than that known for linear system solving in general. Subsequently, the aforementioned authors and Rao [CKPPRSV17] introduced the notion of Eulerian sparsifiers and gave the first $O(m \cdot \text{polylog}(n))$-time algorithm for constructing Eulerian sparsifiers with $O(n\varepsilon^{-2} \cdot \text{polylog}(n))$ edges, based on expander decompositions. They used their method to give the first $m^{1+o(1)}$ time algorithm for solving linear systems in directed Eulerian Laplacians. A follow-up work by the aforementioned authors and Kyng [CKKPPRS18] later gave an improved $O(m \cdot \text{polylog}(n))$-time solver for directed Laplacian linear systems.

As an alternative approach to Eulerian sparsification, Chu, Gao, Peng, Sachdeva, Sawlani, and Wang [CGPSSW18] introduced the short cycle decomposition, and used it to give an $O(mn)$ time algorithm for computing Eulerian sparsifiers with $O(n\varepsilon^{-2} \log^4 n)$ edges. Improved short cycle decomposition constructions by Liu, Sachdeva, and Yu [LSY19], as well as Parter and Yogev [PY19] resulted in an improved running time of $O(m^{1+\delta})$ for any constant $\delta > 0$, for the same sparsity.

Very recently, Sachdeva, Thudi, and Zhao [STZ24] gave an improved analysis of the short cycle decomposition-based construction of Eulerian sparsifiers from [CGPSSW18], improving the resulting sparsity to $O(n\varepsilon^{-2} \log^3 n)$ edges. They complemented their algorithmic construction with an existential result showing that Eulerian sparsifiers with $\breve{O}(n\varepsilon^{-2} \log^2 n + n\varepsilon^{-4/3} \log^{8/3} n)$ edges exist, using recent progress on the matrix Spencer's conjecture [BJM23]. Our fast algorithm in Theorem 2 yields an improved sparsity compared to the strongest existential result in [STZ24] with a significantly improved runtime, and departs from the short cycle decomposition framework followed by that work. Moreover, our existential result in Theorem 4, which also applies [BJM23] (combined with our new framework), improves [STZ24]'s existential result by a logarithmic factor.

Finally, we note that our applications in Section 8 follow from known implications in the literature, e.g., [CKPPSV16; AJSS19; PS22]. In particular, our directed Laplacian linear system solver

follows from reductions in [CKPPSV16; PS22], who showed that an efficient Eulerian sparsification algorithm implies efficient solvers for all directed Laplacian linear systems. Building upon this result, our other applications follow [CKPPSV16; AJSS19], which show how various other primitives associated with Markov chains can be reduced to solving appropriate directed Laplacian systems.

Discrepancy-theoretic approaches to sparsification. The use of discrepancy-theoretic techniques for spectral sparsification has been carried out in several prior works. First, [RR20] showed how to use matrix variance bounds in undirected graphs with the partial coloring framework of [Rot17] to construct linear-sized sparsifiers. Subsequently, this partial coloring-based sparsification algorithm was sped up to run in nearly-linear time by [JRT23] and [STZ24] showed how to adapt these techniques to the Eulerian sparsification setting, by using an improved analysis of the matrix variance induced by algorithms using short cycle decompositions.

Our strongest existential sparsification result (cf. Theorems 4, 26) follows the discrepancy-based partial coloring approach to sparsification pioneered in these works, combining it with our new matrix variance bounds via ER decomposition (Lemma 19) instead of short cycles, as was done in [STZ24]. Recently, concurrent and independent work of [LWZ24] gave a derandomized partial colouring framework for spectral sparsification using the "deterministic discrepancy walk" approach from [PV23], and applied it to obtain polynomial-time deterministic Eulerian sparsifiers satisfying a stronger notion of spectral approximation known as "singular value (SV) approximation" [APPSV23]. This result of [LWZ24] complements, but is largely orthogonal to, our results: it yields directed sparsifiers with larger sparsities and runtimes than ours, but which satisfy stronger notions of sparsification (i.e., SV sparsification) and are obtained deterministically.

1.4 Roadmap

In Section 2, we introduce notation and useful technical tools used throughout the paper. In Section 3 we then provide a technical overview of the rest of the paper. Next, we give our effective resistance decomposition algorithm in Section 4, a key building block in our sparsification methods. In Section 5, we then show how to take advantage of this decomposition by proving a new matrix variance bound for directed edge Laplacians after an electric projection. Crucially, this bound is parameterized by the effective resistance diameter of decomposition pieces.

The remainder of the paper contains applications of our sparsification framework. In Section 6, we prove Theorem 4, our result with the tightest sparsity guarantees. In Section 7, we prove Theorem 2, which obtains a significantly improved runtime at the cost of slightly worse sparsity. In Section 8, we combine our sparsification methods with existing reductions in the literature and overview additional applications of our algorithms for directed graph primitives. Finally, in Section 9, we show how to apply our sparsification subroutines to design state-of-the-art graphical spectral sketches, proving Theorem 5 and an extension to Eulerian graphs that we introduce.

2 Preliminaries

General notation. All logarithms are base e unless otherwise specified. When discussing a graph clear from context with n vertices and edge weight ratio bounded by U, we use the $\breve{O}$ notation to hide $\text{polyloglog}(nU)$ factors for brevity (in runtimes only). We let $[n] \stackrel{\text{def}}{=} \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$.

Vectors. Vectors are denoted in lower-case boldface. $\mathbf{0}_d$ and $\mathbf{1}_d$ are the all-zeroes and all-ones vector respectively of dimension d. $\mathbf{e}_i$ denote the i^{th} basis vector. $\mathbf{u} \circ \mathbf{v}$ denotes the entrywise product of $\mathbf{u}, \mathbf{v}$ of equal dimension.

Matrices. Matrices are denoted in upper-case boldface. We refer to the i^{th} row and j^{th} column of matrix $\mathbf{M}$ by $\mathbf{M}_{i:}$ and $\mathbf{M}_{:j}$ respectively. We use $[\mathbf{v}]_i$ to index into the i^{th} coordinate of vector $\mathbf{v}$, and let $[\mathbf{M}]_{i:} \stackrel{\text{def}}{=} \mathbf{M}_{i:}$, $[\mathbf{M}]_{:j} \stackrel{\text{def}}{=} \mathbf{M}_{:j}$, and $[\mathbf{M}]_{ij} \stackrel{\text{def}}{=} \mathbf{M}_{ij}$ in contexts where $\mathbf{v}$, $\mathbf{M}$ have subscripts.

$\mathbf{I}_d$ is the $d \times d$ identity matrix. For $\mathbf{v} \in \mathbb{R}^d$, $\mathbf{diag}\,(\mathbf{v})$ denotes the associated diagonal $d \times d$ matrix. For linear subspace S of $\mathbb{R}^d$, $\dim(S)$ is its dimension and $\mathbf{P}_S$ is the orthogonal projection matrix onto S. We let $\ker(\mathbf{M})$ and $\mathbf{M}^\dagger$ denote the kernel and pseudoinverse of $\mathbf{M}$. We denote the operator norm (largest singular value) of matrix $\mathbf{M}$ by $\|\mathbf{M}\|_{\text{op}}$, and the Frobenius norm (entrywise ℓ_2 norm) of $\mathbf{M}$ by $\|\mathbf{M}\|_{\text{F}}$. The number of nonzero entries of a matrix $\mathbf{M}$ (resp. vector $\mathbf{v}$) is denoted $\text{nnz}(\mathbf{M})$ (resp. $\text{nnz}(\mathbf{v})$), and the subset of indices with nonzero entries is $\text{supp}(\mathbf{M})$ (resp. $\text{supp}(\mathbf{v})$).

We use $\preceq$ to denote the Loewner partial order on $\mathbb{S}^d$, the symmetric $d \times d$ matrices. We let $\mathbb{U}^d$ denote the set of $d \times d$ real unitary matrices. For $\mathbf{M} \in \mathbb{S}^d$ and $i \in [d]$, we let $\lambda_i(\mathbf{M})$ denote the i^{th} smallest eigenvalue of $\mathbf{M}$, so $\lambda_1(\mathbf{M}) \leq \lambda_2(\mathbf{M}) \leq \ldots \leq \lambda_d(\mathbf{M})$. For positive semidefinite $\mathbf{A} \in \mathbb{S}^d$, we define the seminorm induced by $\mathbf{A}$ by $\|\mathbf{x}\|_{\mathbf{A}}^2 \stackrel{\text{def}}{=} \mathbf{x}^\top \mathbf{A}\mathbf{x}$.

Distributions. $\text{Geom}(p)$ for $p \in (0, 1]$ denotes the geometric distribution on $\mathbb{N}$ with mean $\frac{1}{p}$. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. γ_d denotes the Gaussian measure in dimension d, i.e., for $\mathcal{K} \subseteq \mathbb{R}^d$, $\gamma_d(\mathcal{K}) \stackrel{\text{def}}{=} \Pr_{g \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)}[g \in \mathcal{K}]$; when S is a linear subspace of $\mathbb{R}^d$, we define $\gamma_S(\mathcal{K}) \stackrel{\text{def}}{=} \Pr_{g \sim \mathcal{N}(\mathbf{0}_d, \mathbf{P}_S)}[g \in \mathcal{K}]$.

Graphs. All graphs throughout this paper are assumed to be simple without loss of generality, as collapsing parallel multi-edges does not affect (undirected or directed) graph Laplacians. We denote undirected weighted graphs without an arrow and directed weighted graphs with an arrow, i.e., $G = (V, E, \mathbf{w})$ is an undirected graph with vertices V, edges E, and weights $\mathbf{w} \in \mathbb{R}^E_{\geq 0}$, and $\vec{G}$ is a directed graph. A directed Eulerian graph is a directed graph where weighted in-degree equals weighted out-degree for every vertex. We refer to the vertex set and edge set of a graph G (resp. $\vec{G}$) by $V(G)$ and $E(G)$ (resp. $V(\vec{G})$ and $E(\vec{G})$). We associate a directed edge e from u to v with the tuple (u, v), and an undirected edge with (u, v) and (v, u) interchangeably. We define $h(e) = u$ and $t(e) = v$ to be the head and tail of a directed edge $e = (u, v)$.

Finally, when we are discussing Eulerian sparsification of a graph $\vec{G}$ in the sense of Definition 1, we will always assume henceforth that $G = \text{und}(\vec{G})$ is connected. This is without loss of generality: otherwise, we can define an instance of Definition 1 on each connected component of G. The left and right kernels of $\vec{\mathbf{L}}_{\vec{G}}$ and $\mathbf{L}_G$ are spanned by the all-ones vectors indicating each connected component of G. Moreover, each connected component in G still corresponds to an Eulerian graph. Therefore, satisfying Definition 1 for each component individually implies the same inequality holds for the entire graph, by adding all the component Laplacians.

Subgraphs and graph operations. We say H is a subgraph of G if the edges and vertices of H are subsets of the edges and vertices of G (with the same weights), denoting $H = G_F$ if $E(H) = F$, and defining the same notion for directed graphs. For $U \subseteq V$, we let $G[U]$ denote the induced subgraph of G on U (i.e., keeping all of the edges within U). We let $\text{rev}(\vec{G})$ denote the directed graph with all edge orientations reversed from $\vec{G}$, and $\text{und}(\vec{G})$ denote the undirected graph which removes orientations (both keeping the same weights). When V is a set of vertices, we say $\{V_i\}_{i \in [I]}$ is a partition of V if $\bigcup_{i \in [I]} V_i = V$, and all V_i are disjoint. We say $\{G_j\}_{j \in [J]}$ are a family of edge-disjoint subgraphs of $G = (V, E, \mathbf{w})$ if all $E(G_j)$ are disjoint, and for all $j \in [J]$, $V(G_j) \subseteq V$, $E(G_j) \subseteq E$, and every edge weight in G_j is the same as its weight in G.

Graph matrices. For a graph with edges E and vertices V, we let $\mathbf{B} \in \{-1, 0, 1\}^{E \times V}$ be its edge-vertex incidence matrix, so that when $\vec{G}$ is directed and $e = (u, v)$, $\mathbf{B}_{e:}$ is 2-sparse with $\mathbf{B}_{eu} = 1$, $\mathbf{B}_{ev} = -1$ (for undirected graphs, we fix an arbitrary consistent orientation). For $u, v \in V$, we define $\mathbf{b}_{(u,v)} \stackrel{\text{def}}{=} \mathbf{e}_u - \mathbf{e}_v$. When $\mathbf{B}$ is the incidence matrix associated with graph $G = (V, E, \mathbf{w})$ (resp. $\vec{G}$), we say $\mathbf{x}$ is a circulation in G (resp. $\vec{G}$) if $\mathbf{B}^\top \mathbf{x} = \mathbf{0}_V$; when G (resp. $\vec{G}$) is clear we simply say $\mathbf{x}$ is a circulation. We let $\mathbf{H}, \mathbf{T} \in \{0, 1\}^{E \times V}$ indicate the heads and tails of each edge, i.e., have one nonzero entry per row indicating the relevant head or tail vertex for each edge, respectively, so that $\mathbf{B} = \mathbf{H} - \mathbf{T}$. When clear from context that $\mathbf{w}$ are edge weights, we let $\mathbf{W} \stackrel{\text{def}}{=} \mathbf{diag}\,(\mathbf{w})$. For undirected $G = (V, E, \mathbf{w})$ with incidence matrix $\mathbf{B}$, the Laplacian matrix of G is $\mathbf{L} \stackrel{\text{def}}{=} \mathbf{B}^\top \mathbf{W} \mathbf{B}$. For directed $\vec{G} = (V, E, \mathbf{w})$, the directed Laplacian matrix of $\vec{G}$ is $\vec{\mathbf{L}} \stackrel{\text{def}}{=} \mathbf{B}^\top \mathbf{W} \mathbf{H}$. To disambiguate, we use $\mathbf{L}_G$, $\mathbf{H}_G$, $\mathbf{T}_G$, $\mathbf{B}_G$, etc. to denote matrices associated with a graph G when convenient.

Note that $\vec{\mathbf{L}}^\top \mathbf{1}_V = \mathbf{0}_V$ for any directed Laplacian $\vec{\mathbf{L}}$. If $\vec{G}$ is Eulerian, then its directed Laplacian also satisfies $\vec{\mathbf{L}} \mathbf{1}_V = \mathbf{0}_V$ and $\mathbf{w}$ is a circulation in $\vec{G}$ (i.e., $\mathbf{B}^\top \mathbf{w} = \mathbf{0}_V$). Note that for a directed graph $\vec{G} = (V, E, \mathbf{w})$ and its corresponding undirected graph $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$, the undirected Laplacian is $\mathbf{L}_G = \mathbf{B}^\top \mathbf{W} \mathbf{B}$, and the reversed directed Laplacian is $\vec{\mathbf{L}}_{\text{rev}(\vec{G})} = -\mathbf{B}^\top \mathbf{W} \mathbf{T}$.

We let $\mathbf{\Pi}_V$ denote the Laplacian of the unweighted complete graph on V, i.e., $\mathbf{\Pi}_V \stackrel{\text{def}}{=} \mathbf{I}_V - \frac{1}{|V|} \mathbf{1}_V \mathbf{1}_V^\top$. Note that $\mathbf{\Pi}_V$ is the orthogonal projection on the the subspace spanned by the vector that is 1 in the coordinates of V and 0 elsewhere.

Effective resistance. For undirected $G = (V, E, \mathbf{w})$, the effective resistance (ER) of $u, v \in V$ is $\text{ER}_G(u, v) \stackrel{\text{def}}{=} \mathbf{b}_{(u,v)}^\top \mathbf{L}_G^\dagger \mathbf{b}_{(u,v)}$. We also define $\text{ER}_G(e)$ for $e = (u, v) \in E$ by $\text{ER}_G(e) \stackrel{\text{def}}{=} \text{ER}_G(u, v)$.

Graph linear algebra. In Appendix A we prove the following facts about graph matrices.

Fact 6. *Let $\mathbf{B} = \mathbf{H} - \mathbf{T}$ be the edge-vertex incidence matrix of a graph, let $\mathbf{x}$ be a circulation in the graph (i.e. $\mathbf{B}^\top \mathbf{x} = \mathbf{0}$), and let $\mathbf{X} \stackrel{\text{def}}{=} \mathbf{diag}\,(\mathbf{x})$. Then $\mathbf{H}^\top \mathbf{X} \mathbf{H} = \mathbf{T}^\top \mathbf{X} \mathbf{T}$ and $\mathbf{B}^\top \mathbf{X} \mathbf{H} = -\mathbf{T}^\top \mathbf{X} \mathbf{B}$.*

Fact 7. *Suppose $\vec{G} = (V, E, \mathbf{w}_{\vec{G}}), \vec{H} = (V, F, \mathbf{w}_{\vec{H}})$ share the same vertex set and $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$, $H \stackrel{\text{def}}{=} \text{und}(\vec{H})$. If $\mathbf{B}_{\vec{G}}^\top \mathbf{w}_{\vec{G}} = \mathbf{B}_{\vec{H}}^\top \mathbf{w}_{\vec{H}}$, then $\|\mathbf{L}_G^{\dagger \frac{1}{2}} (\mathbf{L}_G - \mathbf{L}_H) \mathbf{L}_G^{\dagger \frac{1}{2}}\|_{\text{op}} \leq 2 \|\mathbf{L}_G^{\dagger \frac{1}{2}} (\vec{\mathbf{L}}_{\vec{G}} - \vec{\mathbf{L}}_{\vec{H}}) \mathbf{L}_G^{\dagger \frac{1}{2}}\|_{\text{op}}$.*

Fact 8. *Suppose G, H are connected graphs on the same vertex set V, and $\|\mathbf{L}_G^{\dagger/2} (\mathbf{L}_G - \mathbf{L}_H) \mathbf{L}_G^{\dagger/2}\|_{\text{op}} \leq \varepsilon$. Then for any $\mathbf{M} \in \mathbb{R}^{V \times V}$, we have $\|\mathbf{L}_G^{\dagger/2} \mathbf{M} \mathbf{L}_G^{\dagger/2}\|_{\text{op}} \leq (1 + \varepsilon) \|\mathbf{L}_H^{\dagger/2} \mathbf{M} \mathbf{L}_H^{\dagger/2}\|_{\text{op}}$.*

3 Technical overview

In this section, we overview our strategy for preserving degree balance in efficient directed sparsification primitives, in greater detail than in Section 1.2. We first review a motivating construction for undirected sparsifiers via randomly signed edge weight updates. Then we introduce our extension of this construction to the Eulerian setting, based on electric projections of edge Laplacians.

To bound the spectral error incurred by random reweightings in the Eulerian setting, we then describe a new asymmetric matrix variance bound under certain bounds on the effective resistance diameter and weight ratio of the edges under consideration (Lemma 19). This Lemma 19 is the key technical tool enabling our results, proven in Section 5.

We then describe an *effective resistance decomposition* (Definition 9) subroutine we introduce in Section 4, used to guarantee the aforementioned weight and effective resistance bounds hold in our

sparsification procedures. Finally, we explain how each of our algorithms (in proving Theorems 2 and 4) and their applications in Sections 6, 7, 8, and 9 build upon these common primitives.

Sparsification from random signings. To motivate our approach, consider the following conceptual framework in the simpler setting of undirected sparsification. (Variants of this framework have appeared in the recent literature [CGPSSW18; RR20; JRT23].) Starting from undirected graph $G = (V, E, \mathbf{w})$ with n vertices and m edges, we initialize $\mathbf{w}_0 \leftarrow \mathbf{w}$ and in each iteration t, let

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t \circ (\mathbf{1}_E + \eta \mathbf{s}_t), \tag{3}$$

where $\mathbf{s}_t \in \{\pm 1\}^E$ has independent Rademacher entries and $\eta \in (0, 1]$. Intuitively, the update (3) drives edge weights rapidly to zero, as it induces an exponential negative drift on each weight:

$$\mathbb{E} \log \left(\frac{[\mathbf{w}_{t+1}]_e}{[\mathbf{w}_t]_e} \right) = \mathbb{E} \log(1 + \eta [\mathbf{s}_t]_e) \approx -\eta^2. \tag{4}$$

This phenomenon is most obvious when $\eta = 1$ (which suffices for undirected sparsification), as a constant fraction of edges are immediately zeroed out in each iteration, but (4) quantifies this for general η. Next, consider the spectral approximation error induced by the first step ($t = 0$), where we denote $G_0 \stackrel{\text{def}}{=} G = (V, E, \mathbf{w}_0)$ and $G_1 \stackrel{\text{def}}{=} (V, E, \mathbf{w}_1)$, and let $\eta = 1$. By standard matrix concentration inequalities on Rademacher signings (see, e.g., Lemma 30), w.h.p.,

$$\begin{aligned} \left\| \mathbf{L}_{G_0}^{\dagger\frac{1}{2}} \left(\mathbf{L}_{G_1} - \mathbf{L}_{G_0} \right) \mathbf{L}_{G_0}^{\dagger\frac{1}{2}} \right\|_{\text{op}} &= \left\| \mathbf{L}_{G_0}^{\dagger\frac{1}{2}} \mathbf{B}_G^\top \left(\mathbf{W}_1 - \mathbf{W}_0 \right) \mathbf{B}_G \mathbf{L}_{G_0}^{\dagger\frac{1}{2}} \right\|_{\text{op}} \\ &= \left\| \sum_{e \in E} \mathbf{s}_e \mathbf{A}_e \right\|_{\text{op}} \lesssim \sqrt{\left\| \sum_{e \in E} \mathbf{A}_e^2 \right\|_{\text{op}}}, \text{ where } \mathbf{A}_e \stackrel{\text{def}}{=} \mathbf{w}_e \mathbf{L}_G^{\dagger\frac{1}{2}} \mathbf{b}_e \mathbf{b}_e^\top \mathbf{L}_G^{\dagger\frac{1}{2}}. \end{aligned} \tag{5}$$

This argument suggests that it is crucial to control the following matrix variance statistic, $\sigma^2 \stackrel{\text{def}}{=} \| \sum_{e \in E} \mathbf{A}_e^2 \|_{\text{op}}$, as we incur spectral approximation error $\approx \sigma$. It is straightforward to see that, letting $\rho_{\max} \stackrel{\text{def}}{=} \max_{e \in E} \mathbf{w}_e \mathbf{b}_e^\top \mathbf{L}_G^\dagger \mathbf{b}_e = \max_{e \in E} \mathbf{w}_e \text{ER}_G(e)$ be the maximum weighted effective resistance of any edge in G, we have

$$\sum_{e \in E} \mathbf{A}_e^2 = \sum_{e \in E} \mathbf{w}_e \mathbf{L}_G^{\dagger\frac{1}{2}} \mathbf{b}_e \left(\mathbf{w}_e \mathbf{b}_e^\top \mathbf{L}_G^\dagger \mathbf{b}_e \right) \mathbf{b}_e^\top \mathbf{L}_G^{\dagger\frac{1}{2}} \preceq \rho_{\max} \mathbf{L}_G^{\dagger\frac{1}{2}} \left(\sum_{e \in E} \mathbf{w}_e \mathbf{b}_e \mathbf{b}_e^\top \right) \mathbf{L}_G^{\dagger\frac{1}{2}} \preceq \rho_{\max} \mathbf{I}_V. \tag{6}$$

By zeroing entries of $\mathbf{s}$ corresponding to the largest half of $\mathbf{w}_e \text{ER}_G(e)$ values, we can ensure $\rho_{\max} = O(\frac{n}{m})$, since $\sum_{e \in E} \mathbf{w}_e \text{ER}_G(e) \leq n$. Hence, (5) shows the spectral approximation error is $\lesssim \sqrt{n/m}$. Since the edge sparsity m decreases by a constant factor in each iteration t when $\eta = 1$, this induces a geometric sequence in the spectral approximation quality terminating at $\approx \varepsilon$ when $\text{nnz}(\mathbf{w}_t) \approx n\varepsilon^{-2}$, as desired. We remark that Rademacher signings are not the only way to instantiate this scheme; indeed, [RR20; JRT23] show how to use discrepancy-theoretic tools to choose the update (3) in a way which does not lose logarithmic factors in the spectral error bound.

Asymmetric variance statistics and ER decomposition. The aforementioned framework for undirected sparsification runs into immediate difficulties in the context of Eulerian sparsification (Definition 1), as it does not preserve degree balances. Previous Eulerian sparsification methods sidestepped this obstacle by either fixing degrees after sampling and incurring errors (e.g.,

via expander decomposition) or coordinating the sampling in a degree-preserving way (e.g., via short cycle decomposition). We propose an arguably more direct approach to preserving degrees, departing from prior work. Consider Eulerian $\vec{G}_0 \stackrel{\text{def}}{=} \vec{G} = (V, E, \mathbf{w}_0 \stackrel{\text{def}}{=} \mathbf{w})$. On iteration $t \geq 0$, let

$$\mathbf{P}_t \stackrel{\text{def}}{=} \mathbf{I}_E - \mathbf{W}_t \mathbf{B}_{\vec{G}} \mathbf{L}^{\dagger}_{G_t^2} \mathbf{B}_{\vec{G}}^{\top} \mathbf{W}_t,$$

where $\mathbf{L}_{G_t^2}$ is the undirected Laplacian of $G_t^2 \stackrel{\text{def}}{=} (V, E, \mathbf{w}_t^2)$, $\mathbf{w}_t^2$ is entrywise, and $\mathbf{W}_t \stackrel{\text{def}}{=} \mathbf{diag}\,(\mathbf{w}_t)$. Observe that $\mathbf{P}_t$ is the orthogonal projection matrix onto the space of degree-preserving reweightings on the graph G_t with weights $\mathbf{w}_t$, i.e., for all $\mathbf{x} \in \mathrm{Im}(\mathbf{P}_t)$, we have $\mathbf{B}_{\vec{G}_t}^{\top}(\mathbf{w}_t \circ \mathbf{x}) = \mathbf{0}_V$. Our starting point is thus a modification of the reweighting scheme (3), where the Rademacher vector $\mathbf{s}_t$ is replaced by $\mathbf{x}_t \stackrel{\text{def}}{=} \mathbf{P}_t \mathbf{s}_t$, and we choose an appropriate step size $\eta \approx \log^{-1/2}(n)$ to ensure no edge weight falls below 0. In other words, we simply let

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t \circ (\mathbf{1}_E + \eta \mathbf{x}_t), \text{ where } \mathbf{x}_t \leftarrow \mathbf{P}_t \mathbf{s}_t. \tag{7}$$

Because this reweighting scheme preserves degree imbalance by construction, it remains to analyze two properties of the reweighting. First, how much does the spectral approximation factor in (2) grow in each iteration? Second, does the reweighting significantly decrease the graph sparsity (ideally, after few iterations)? We postpone discussion of the second point until the end of this overview, when we discuss implementations of our framework. Our analysis of weight decay will ultimately carefully quantify the intuition in (4) with an appropriate step size schedule.

Regarding the first point, matrix Rademacher inequalities (extending (5) to the asymmetric setting) show that the spectral error in the first step $t = 0$ is controlled by

$$\begin{gathered} \sigma^2 \stackrel{\text{def}}{=} \max\left(\left\| \sum_{e \in E} \widetilde{\mathbf{A}}_e \widetilde{\mathbf{A}}_e^{\top} \right\|_{\text{op}}, \left\| \sum_{e \in E} \widetilde{\mathbf{A}}_e \widetilde{\mathbf{A}}_e^{\top} \right\|_{\text{op}} \right), \\ \text{where } \widetilde{\mathbf{A}}_e \stackrel{\text{def}}{=} \sum_{f \in E} \mathbf{P}_{fe} \mathbf{A}_f \text{ and } \mathbf{A}_e \stackrel{\text{def}}{=} \mathbf{w}_e \mathbf{L}_G^{\frac{\dagger}{2}} \mathbf{b}_e \mathbf{e}_{h(e)}^{\top} \mathbf{L}_G^{\frac{\dagger}{2}}, \end{gathered} \tag{8}$$

and we abbreviate $G = \mathrm{und}(\vec{G})$ and $\mathbf{P}_0 = \mathbf{P}$ for short. To briefly explain the formula (8), note that analogously to (5), the matrix $\mathbf{A}_e$ is defined so that the one-step spectral error when reweighting by Rademacher $\mathbf{s}$ (in the sense of (7)) is precisely $\|\sum_{e \in E} \mathbf{s}_e \mathbf{A}_e\|_{\text{op}}$. Correspondingly, the matrix $\widetilde{\mathbf{A}}_e$ is defined to capture the correct error statistic after first applying $\mathbf{P}$ to $\mathbf{s}$.

A primary technical contribution of our work is quantifying a sufficient condition under which the asymmetric variance statistic (8) is bounded, stated formally as Lemma 19. Recall that in the undirected setting, (6) bounds σ^2 in terms of the maximum weighted ER of the edges we choose to reweight. Similar logic suggests that the Eulerian variance statistic in (8) is small if $\mathbf{e}_v^{\top} \mathbf{L}_G^{\dagger} \mathbf{e}_v$ is bounded for each vertex $v \in V$, i.e., the diagonal entries of $\mathbf{L}_G^{\dagger}$ are small. In the undirected, unweighted case, $\mathbf{e}_v^{\top} \mathbf{L}_G^{\dagger} \mathbf{e}_v$ is bounded for all $v \in V$ if G has small *effective resistance diameter*, i.e., $\mathrm{ER}_G(u, v) \stackrel{\text{def}}{=} \mathbf{b}_{(u,v)}^{\top} \mathbf{L}_G^{\dagger} \mathbf{b}_{(u,v)}$ is small for all $(u, v) \in V \times V$ (Lemma 18).

This intuition neglects at least three factors: it only captures the variance matrix $\sum_{e \in E} \mathbf{A}_e \mathbf{A}_e^{\top}$ (rather than $\sum_{e \in E} \mathbf{A}_e^{\top} \mathbf{A}_e$), it is based on the matrices $\mathbf{A}_e$ (rather than $\widetilde{\mathbf{A}}_e$), and it ignores the effects of weights. Our bound in Lemma 19 tackles all three of these factors by using graph-theoretic construction we introduce, called an ER decomposition (Definition 9). Again considering only the first step for simplicity, we prove that if $\vec{H} = (V, F, \mathbf{w}_F)$ is a subgraph of $\vec{G}$ whose vertices all lie in

U, the quantities $\sum_{f\in F}\widetilde{\mathbf{A}}_f^\top\widetilde{\mathbf{A}}_f$ and $\sum_{f\in F}\widetilde{\mathbf{A}}_f\widetilde{\mathbf{A}}_f^\top$ are both bounded (in the Loewner ordering) by

$$\rho_{\max}(F)\cdot\mathbf{L}_G^{\dagger\frac{1}{2}}\mathbf{L}_H\mathbf{L}_G^{\dagger\frac{1}{2}},\text{ where }\rho_{\max}(F)\stackrel{\text{def}}{=}\left(\max_{f\in F}\mathbf{w}_f\right)\cdot\left(\max_{u,v\in U}\mathrm{ER}_G(u,v)\right),\text{ and }H\stackrel{\text{def}}{=}\mathrm{und}(\vec{H}).$$

This suggests that if we can isolate a cluster of edges F on a vertex set U, such that all edges in F have roughly even edge weight, and such that U has bounded effective resistance diameter through G (inversely proportional to the weights in F), we can pay for the contribution of all $\widetilde{\mathbf{A}}_f$ for $f\in F$ to the variance statistic in (8). We accordingly define ER decompositions to decompose E into such clusters $\{F_k\}_{k\in[K]}$, each with bounded $\rho_{\max}(F_k)\approx\frac{n}{m}$.

Our ER decomposition scheme. We take a brief digression to answer: how do we find such an edge-disjoint decomposition $\{F_k\}_{k\in[K]}$, each with bounded $\rho_{\max}(F_k)$? In fact, such a decomposition is immediately implied by the related ER decomposition of [AALG18], save two issues. The ER decomposition of [AALG18] only guarantees that a constant fraction of edges *by total weight* are cut, as opposed to by edge count (which our recursion can tolerate). The more pressing issue is that the [AALG18] algorithm uses $\Omega(mn)$ time, necessitating design of a faster decomposition scheme.

In Section 4, we provide a simple near-linear time decomposition scheme which makes use of the well-known fact that effective resistances in a graph form a metric. We first partition the undirected graph G in question into subgraphs $\{G^j\}_{j_{\min}\le j\le j_{\max}}$ for appropriate $j_{\max}-j_{\min}+1=O(\log U)$, where G^j consists of edges with weight between 2^j and 2^{j+1}, and U is the multiplicative range of edge weights. In each G^j, it suffices to partition the vertices to induce subgraphs $\{G_i^j\}_{i\in[K_j]}$, each with ER diameter $\approx\frac{n}{m}\cdot 2^{-j}$, and such that few edges are cut. We accomplish this by first providing constant-factor estimates to all edge effective resistances using standard sketching tools (Lemma 13). Within each subgraph G^j, we induce a shortest path metric based on our ER overestimates, and then apply classic region-growing techniques [GVY96] to partition the subgraphs into pieces of bounded shortest path diameter without cutting too many edges.

Implementations of our framework. Finally, we briefly outline how Theorems 2, 4, and 5 follow from the frameworks we outlined. Our Eulerian sparsification algorithms (for establishing Theorems 2 and 4) simply interleave computation of an ER decomposition on the current graph, with a small number of reweightings roughly of the form (7). For our nearly-linear time algorithm in Theorem 2, in each reweighting (7), we zero out the half of entries of $\mathbf{s}_t$ which are cut by the ER decomposition, and additionally enforce a linear constraint that the total weight is preserved. We show that by making the intuition (7) rigorous, after polylogarithmically many reweightings, a constant fraction of edge weights have decreased by a polynomial factor, which is enough to explicitly delete them from the graph and (after fixing degrees by routing through a spanning tree) incur small spectral error. This lets us recurse and obtain the same geometric sequence behavior on our accumulated spectral error bound as in the undirected setting.

Our proof of Theorem 4 applies carefully-coordinated reweighting vectors $\mathbf{x}_t$ which yield smaller spectral error than naïve random signing. We choose these vectors $\mathbf{x}_t$ based on recent progress towards the matrix Spencer conjecture (a well-known open problem in discrepancy theory) due to [BJM23]. Specifically, [BJM23] (along with earlier works, e.g., [Rot17; RR23]) provide tools which construct "partial colorings" $\mathbf{x}_t$ such that $[\mathbf{x}_t]_e=-1$ for a constant fraction of $e\in E$, and

$$\left\|\sum_{e\in E}[\mathbf{x}_t]_e\mathbf{A}_e\right\|_{\text{op}}$$

is smaller than what matrix Rademacher inequalities would predict for random $\mathbf{x}_t$ (based on the matrix variance statistic). Applying these higher-quality reweightings $\mathbf{x}_t$ in each iteration through (7) (with $\eta = 1$) then directly decreases the edge sparsity by a constant factor in each iteration, allowing for simple control of the spectral error in (2). This strategy immediately yields Theorem 4 upon recursing. As mentioned previously, in Appendix C, we examine natural routes which could further improve upon the sparsity bounds of Theorem 4.

Finally, we show that our subroutines designed for Eulerian sparsification compose well with a framework for obtaining graphical spectral sketches by [CGPSSW18], based on expander decomposition. Specifically, [CGPSSW18] (based on similar ideas in [ACKQWZ16; JS18]) showed that spectral bounds between degree matrices and Laplacians which hold in expander graphs yield improved per-vector quadratic form guarantees. We make the simple observation that expander subgraphs with large minimum degree also have bounded effective resistance diameter (Lemma 52). Hence, directly using our algorithms for sparsifying pieces of an effective resistance decomposition using electric projections in place of short cycle decompositions (as used in [CGPSSW18]) improves state-of-the-art runtimes by $m^{o(1)}$ factors. Our spectral sketch algorithm is flexible enough to extend straightforwardly to the Eulerian setting (following Definition 45), as described in Theorem 56.

4 Effective resistance decomposition

In this section, we show how to efficiently decompose a weighted, undirected graph into subgraphs with bounded weight ratio, small effective resistance diameter (relative to the edge weights it contains), a limited number of edges cut, and each vertex appearing in a limited number of subgraphs. This procedure will be a key subroutine in our sparsification algorithms, as captured by the variance bound in Lemma 19. Below in Definition 9 we formally define this type of decomposition guarantee and then in Proposition 10 we provide our main result on computing said decompositions.

Definition 9 (ER decomposition)**.** *We call $\{G_i\}_{i\in[I]}$ a (ρ, r, J)-*effective resistance (ER) decomposition *if $\{G_i\}_{i\in[I]}$ are edge-disjoint subgraphs of $G = (V, E, \mathbf{w})$, and the following hold.*

1. Bounded weight ratio*: For all $i \in [I]$, $\frac{\max_{e\in E(G_i)} \mathbf{w}_e}{\min_{e\in E(G_i)} \mathbf{w}_e} \leq r$.*
2. Effective resistance diameter*: For all $i \in [I]$, $(\max_{e\in E(G_i)} \mathbf{w}_e) \cdot (\max_{u,v\in V(G_i)} \mathrm{ER}_G(u, v)) \leq \rho$.*
3. Edges cut*: $|E(G) \setminus (\bigcup_{i\in[I]} E(G_i))| \leq \frac{m}{2}$.*
4. Vertex coverage*: Every vertex $v \in V(G)$ appears in at most J of the subgraphs.*

Proposition 10. *There is an algorithm,* ERDecomp(G, r, δ)*, which given any $G = (V, E, \mathbf{w})$ with $n = |V|$, $m = |E|$, $\frac{\max_{e\in E} \mathbf{w}_e}{\min_{e\in E} \mathbf{w}_e} \leq W$ and $r \geq 1$, $\delta \in (0, 1)$, computes a*

$$\left(\frac{8rn\log(n+1)}{m}, r, \log_r(W) + 3\right)\text{-ER decomposition of } G,$$

with probability $\geq 1 - \delta$ in time[4]

$$\breve{O}\left(m \log\left(\frac{n}{\delta}\right) + n\log(n)\log_r(W)\right).$$

[4]The $O(n \log n)$ term arises from the use of Fibonacci heaps to compute shortest paths in undirected graphs in Proposition 15. There are results that have since obtained faster algorithms for computing shortest paths in undirected graphs [Tho99; DMSY23]. Moreover, the shortest paths do not necessarily need to be computed exactly, so it is possible that this factor could be improved as it has been in other region growing settings [MPX13; AN19]. However, since this is not a bottleneck in the runtimes of our main results, we make no attempt to improve it here.

In the remainder of this section, we prove Proposition 10. The algorithm consists of two components. First, we use standard randomized algorithms (Lemma 14) to efficiently compute an ER overestimate for the graph edges (Definition 11). Then, we apply a standard result on region growing (Proposition 15) from [GVY96] to efficiently partition the edges within one weight range (Lemma 16). Applying this decomposition scheme at every weight scale to the graph with edge lengths given by the effective resistance overestimates then yields the result. Interestingly, the only use of randomization in this algorithm is in computing overestimates of effective resistances and if a sufficiently efficient deterministic subroutine for this was developed, substituting this subroutine into our algorithm would would obtain a deterministic counterpart of Proposition 10.

Definition 11 (Effective resistance overestimate)**.** *Given $G = (V, E, \mathbf{w})$ with $n = |V|$, we call $\tilde{\mathbf{r}} \in \mathbb{R}^E$ an* α-approximate effective resistance (ER) overestimate *if*

$$\mathbf{w}^\top \tilde{\mathbf{r}} \leq \alpha n \text{ and } \tilde{\mathbf{r}}_e \geq \mathrm{ER}_G(e) \text{ for all } e \in E.$$

To efficiently compute ER overestimates for use in our decomposition algorithms, we rely on near-linear time undirected Laplacian linear system solvers. To begin, we first provide a statement of the current fastest Laplacian linear system solver in the literature.

Proposition 12 (Theorem 1.6, [JS21])**.** *Let $\mathbf{L}_G$ be the Laplacian of $G = (V, E, \mathbf{w})$. There is an algorithm which takes $\mathbf{L}_G$, $\mathbf{b} \in \mathbb{R}^V$, and $\delta, \xi \in (0, 1)$, and outputs $\mathbf{x}$ such that with probability $\geq 1 - \delta$, $\mathbf{x}$ is an ξ-approximate solution to $\mathbf{L}_G \mathbf{x} = \mathbf{b}$, i.e.,*

$$\left\| \mathbf{x} - \mathbf{L}_G^\dagger \mathbf{b} \right\|_{\mathbf{L}_G} \leq \xi \left\| \mathbf{L}_G^\dagger \mathbf{b} \right\|_{\mathbf{L}_G},$$

in time $\breve{O}(|E| \cdot \log \frac{1}{\delta\xi})$. Moreover, the algorithm returns $\mathbf{x} = \mathbf{M}\mathbf{b}$ where $\mathbf{M}$ is a random linear operator constructed independently of $\mathbf{b}$, such that the above guarantee holds with $1 - \delta$ for all $\mathbf{b}$.

The runtime guarantee of the above proposition follows from Theorem 1.6 of [JS21]. We now briefly justify the second clause in Proposition 12, i.e. that the Laplacian solver is a randomized linear function of $\mathbf{b}$, as it is not explicitly stated in [JS21]. Theorem 1.6 follows by combining an algorithm which constructs low-stretch subgraphs with a recursive preconditioning framework (Algorithm 12). Algorithm 12 returns the result of an error-robust accelerated gradient descent procedure PreconNoisyAGD, which only applies linear transformations and a procedure RichardsonSolver, to $\mathbf{b}$. In turn, RichardsonSolver performs only linear transformations and another procedure PreconRichardson to its input. Finally, PreconRichardson applies linear transformations and Algorithm 12 to its input: in addition, these calls to Algorithm 12 operate on strictly smaller problems. Thus, if we assume that these inner calls to Algorithm 12 perform a linear transformation of $\mathbf{b}$, the outer call is also a linear transformation: the last claim in Proposition 12 follows.

Proposition 12 combined with a Johnson-Lindenstrauss based sketching approach from [SS11] shows we can efficiently approximate a set of effective resistances to constant multiplicative error, which we summarize in the following. We remark that the runtime in [SS11] is larger than in Lemma 13; our improvement stems from replacing the solver used there with Proposition 12.

Lemma 13 (Theorem 2, [SS11])**.** *Let $\delta \in (0, 1)$, let $\mathbf{L}_G$ be the Laplacian of $G = (V, E, \mathbf{w})$, and let $S \subseteq V \times V$. There is an algorithm, $\textsc{ApproxER}(G, S, \delta)$, which runs in time $\breve{O}((|E| + |S|) \log(\frac{|S|}{\delta}))$ and outputs $\mathbf{r} = \{\mathbf{r}_{(u,v)}\}_{(u,v)\in S}$ satisfying with probability $\geq 1 - \delta$,*

$$\frac{2}{3}\mathrm{ER}_G(u, v) \leq \mathbf{r}_{(u,v)} \leq \frac{4}{3}\mathrm{ER}_G(u, v), \text{ for all } (u, v) \in S.$$

Proof. Consider the following algorithm for approximating $\mathrm{ER}_G(u,v)$ for some $(u,v) \in S$. We output the median of $K = \Theta(\log \frac{|S|}{\delta})$ independent evaluations of

$$\left\| \mathbf{Q}\mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{M}(\mathbf{e}_v - \mathbf{e}_u) \right\|_2^2, \tag{9}$$

for $\mathbf{Q} \in \mathbf{R}^{\Theta(1)\times|E|}$ filled with random scaled Gaussian entries, and where $\mathbf{M}$ is the random linear operator given by the approximate solver in Proposition 12 with a sufficiently small constant ξ. We claim that (9) lies in the range $[\frac{2}{3}\mathrm{ER}_G(u,v), \frac{4}{3}\mathrm{ER}_G(u,v)]$ with probability $\frac{2}{3}$. By standard Johnson-Lindenstrauss guarantees (see, e.g., the proof of Theorem 2 in [SS11]), it suffices to prove that with probability $\frac{5}{6}$, letting $\mathbf{M}$ be the resulting linear operator from Proposition 12,

$$\left| \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{M}\mathbf{b} \right\|_2^2 - \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{L}_G^{\dagger}\mathbf{b} \right\|_2^2 \right| \leq \frac{1}{4} \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{L}_G^{\dagger}\mathbf{b} \right\|_2^2.$$

To this end, using $0.9\|\mathbf{u}\|_2^2 - 11\|\mathbf{v}\|_2^2 \leq \|\mathbf{u}+\mathbf{v}\|_2^2 \leq 1.1\|\mathbf{u}\|_2^2 + 11\|\mathbf{v}\|_2^2$, we have

$$\left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{M}\mathbf{b} \right\|_2^2 \leq 1.1 \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{L}_G^{\dagger}\mathbf{b} \right\|_2^2 + 11 \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G \left(\mathbf{L}_G^{\dagger}\mathbf{b} - \mathbf{M}\mathbf{b} \right) \right\|_2^2,$$
$$\left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{M}\mathbf{b} \right\|_2^2 \geq 0.9 \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G\mathbf{L}_G^{\dagger}\mathbf{b} \right\|_2^2 - 11 \left\| \mathbf{W}_G^{\frac{1}{2}}\mathbf{B}_G \left(\mathbf{L}_G^{\dagger}\mathbf{b} - \mathbf{M}\mathbf{b} \right) \right\|_2^2,$$

so choosing $\delta = \frac{1}{6}$ and $\xi = \frac{1}{100}$ in Proposition 12 yields the desired claim on each individual evaluation of (9). Thus, by Chernoff bounds the median estimate will lie in the specified range with probability $\geq 1 - \frac{\delta}{|S|}$, yielding correctness after a union bound over all of S.

We now discuss how to implement the above algorithm within the stated runtime. For each independent run $k \in [K]$, we first precompute $\mathbf{Q}\mathbf{W}_G^{1/2}\mathbf{B}_G$ in the given time, and apply $\mathbf{M}$ from Proposition 12 to each of the $\Theta(1)$ rows of this matrix. Notably, we can reuse the same random seed in the solver of [JS21] so that the random linear operator $\mathbf{M}$ provided by Proposition 12 is the same for all rows of $\mathbf{Q}\mathbf{W}_G^{1/2}\mathbf{B}_G$. The random linear function $\mathbf{M}$ is constructed obliviously to the choice of $\mathbf{Q}$, so $\mathbf{Q}$ is independent of these calls and Johnson-Lindenstrauss applies. Each evaluation of (9) takes constant time, which we need to repeat $|S|K$ times in total. □

Our ER overestimate computations then follow from an immediate application of Lemma 13.

Lemma 14. *There is a randomized algorithm, that given any $G = (V, E, \mathbf{w})$ with $n = |V|$, $m = |E|$, computes a 2-approximate ER overestimate with probability $\geq 1 - \delta$ in $\breve{O}(m \log \frac{n}{\delta})$ time.*

Proof. Consider applying Lemma 13 with $S = E$ and the specified δ. In $\breve{O}(m \log \frac{n}{\delta})$ time this procedure computes $\mathbf{r} \in \mathbb{R}^E$ such that with probability $\geq 1 - \delta$,

$$\frac{2}{3}\mathrm{ER}_G(u,v) \leq \mathbf{r}_{(u,v)} \leq \frac{4}{3}\mathrm{ER}_G(e), \text{ for all } e \in E.$$

Our algorithm simply computes this $\mathbf{r}$ and then outputs $\tilde{\mathbf{r}} = \frac{3}{2}\mathbf{r}$. The output $\tilde{\mathbf{r}}$ has the desired properties as $\tilde{\mathbf{r}}_e \geq \mathrm{ER}_G(e)$ for all $e \in E$ and

$$\sum_{e\in E} \mathbf{w}_e\tilde{\mathbf{r}}_e \leq \left(\frac{4}{3}\cdot\frac{3}{2}\right) \sum_{e\in E} \mathbf{w}_e \cdot \mathrm{ER}_G(e) \leq 2n,$$

as $\sum_{e\in E} \mathbf{w}_e\mathrm{ER}_G(e)$ is $n - c$ where c is the number of connected components in G. □

Next, we provide a key subroutine from prior work used in our decomposition.

Proposition 15 (Region growing, [GVY96], Section 4)**.** *There is a deterministic algorithm that given $G = (V, E, \mathbf{w})$ with $n = |V|$, $m = |E|$, edge lengths $\boldsymbol{\ell} \in \mathbb{R}^E_{>0}$ and $d > 0$, in $O(m+n\log n)$-time outputs a partition $\{S_k\}_{k\in[K]}$ of V, each with diameter $\leq 2d\log(n+1)$ with respect to $\boldsymbol{\ell}$, and with*

$$d \cdot \sum_{e\in\partial(\{S_k\}_{k\in[K]})} \mathbf{w}_e \leq 2\mathbf{w}^\top \boldsymbol{\ell},$$

where $\partial(\{S_k\}_{k\in[K]})$ is the set of edges $(u, v) \in E$ with $u \in S_i$, $v \in S_j$ and $i \neq j$.

By applying Proposition 15 instantiated with appropriate edge lengths, we have the following.

Lemma 16. *There is a deterministic algorithm that given $G = (V, E, \mathbf{w})$ with $n = |V|$, $m = |E|$, edge lengths $\boldsymbol{\ell} \in \mathbb{R}^E_{>0}$, and parameters $v, \alpha > 0$ and $r > 1$, in $O(m + n\log n)$-time outputs vertex-disjoint subgraphs $\{G_k\}_{k\in[K]}$ such that the following hold.*

1. *$\bigcup_{k\in[K]} E(G_k) \subseteq F$ for $F \stackrel{\text{def}}{=} \{e \in E \mid \mathbf{w}_e \in (\frac{v}{r}, v]\}$.*
2. *For all $k \in [K]$, the diameter of G_k with respect to $\boldsymbol{\ell}$ is at most $\frac{\alpha}{\max_{e\in E(G_k)} \mathbf{w}_e}$.*
3. *$|F \setminus \{\bigcup_{k\in[K]} E(G_k)\}| \leq \frac{4r\ln(n+1)}{\alpha} \cdot \sum_{e\in F} \mathbf{w}_e\boldsymbol{\ell}_e$.*

Proof. Let $\bar{\mathbf{w}}_e = \mathbf{w}_e$ for all $e \in F$ and $\bar{\mathbf{w}}_e = 0$ for all $e \in E \setminus F$. We apply Proposition 15 to G with $\mathbf{w} \leftarrow \bar{\mathbf{w}}$ and $d \leftarrow \frac{\alpha}{2v\log(n+1)}$ to obtain $\{S_k\}_{k\in[K]}$. Define $\{G_k\}_{k\in[K]}$ so that $V(G_k) = S_k$ and $E(G_k)$ are the edges of F with both endpoints in S_k, with the same weight as in G.

We prove that the $\{G_k\}_{k\in[K]}$ satisfy Items 1, 2, and 3. Item 1 follows directly by construction. Next, Proposition 15 implies that the diameter of each G_k with respect to ℓ is at most $\frac{\alpha}{v}$. Item 2 then follows as $\max_{e\in E(G_k)} \mathbf{w}_e \leq v$. For Item 3, note that Proposition 15 implies that

$$\left(\frac{\alpha}{2v\ln(n+1)}\right) \sum_{e\in E\setminus(\bigcup_{k\in[K]} E(G_k))} \bar{\mathbf{w}}_e \cdot \leq 2\bar{\mathbf{w}}^\top \boldsymbol{\ell}.$$

Item 3 then follows from combining the above, $\bar{\mathbf{w}}^\top \boldsymbol{\ell} = \sum_{e\in F} \mathbf{w}_e\boldsymbol{\ell}_e$, and

$$\left|F \setminus \left\{\bigcup_{k\in[K]} E(G_k)\right\}\right| = \sum_{\substack{e\in E\setminus(\bigcup_{k\in[K]} E(G_i)) \\ \bar{\mathbf{w}}_e>0}} \frac{v}{v} < \sum_{e\in E\setminus(\bigcup_{k\in[K]} E(G_k))} \frac{r\cdot\bar{\mathbf{w}}_e}{v}.$$

□

Proof of Proposition 10. Consider the following algorithm. First, apply Lemma 14 to compute a 2-approximate effective resistance overestimate with probability $\geq 1-\delta$, and save these as $\boldsymbol{\ell} \in \mathbb{R}^E_{>0}$. We then apply Lemma 16 for all integers $j \in [j_{\min}, j_{\max}]$ where $j_{\min} = \lfloor \log_r(\min_{e\in E} \mathbf{w}_e)\rfloor$ and $j_{\max} = \lceil \log_r(\max_{e\in E} \mathbf{w}_e)\rceil$ with

$$v \leftarrow v_j \stackrel{\text{def}}{=} r^j,\ \alpha \leftarrow \frac{16rn\log(n+1)}{m},\ \text{and } r \leftarrow r.$$

For all $j \in [j_{\min}, j_{\max}]$ we let $\{G^j_i\}_{i\in[K_j]}$ be the vertex-disjoint subgraphs output by Lemma 16 and we let F_j be the value of F for this application of Lemma 16. This algorithm has the desired

runtime as applying Lemma 14 takes time $\breve{O}(m \log \frac{n}{\delta})$ and each application of Lemma 16 takes time $O(|E(G_k)| + n \log n)$. Note that the sum of all the $O(|E(G_k)|)$ terms only contributes a single $O(m)$ to the runtime. Additionally, the number of distinct j is

$$j_{\max} - j_{\min} + 1 \leq \log_r \left(\max_{e \in E(G_i)} \mathbf{w}_e \right) + 1 - \left(\log_r \left(\min_{e \in E(G_i)} \mathbf{w}_e \right) - 1 \right) + 1 = \log_r(W) + 3 \,. \tag{10}$$

The runtime follows and it remains only to show that the output $\{G_i^j\}_{j_{\min} \leq j \leq j_{\max}, i \in [K_j]}$ have the desired properties provided that the $\boldsymbol{\ell}$ were indeed a 2-approximate ER overestimate.

Bounded weight ratio (Item 1). This follows directly by construction from Lemma 16.

Effective resistance diameter (Item 2). By Lemma 16, Item 2 we know that for any G_i^j it is the case that the diameter of G_i^j with respect to $\boldsymbol{\ell}$ is at most $\alpha(\max_{e \in E(G_i^j)} \mathbf{w}_e)^{-1}$. Consequently, for each $u, v \in V(G_i^j)$ it is the case that there is a path of edges whose sum of lengths is at most $\alpha(\max_{e \in E(G_i^j)} \mathbf{w}_e)^{-1}$. Each of these lengths is at least the effective resistance of the associated edge. Since effective resistances form a metric, by triangle inequality this means

$$\max_{u,v \in V(G_i^j)} \mathrm{ER}_G(u, v) \leq \frac{\alpha}{\max_{e \in E(G_i^j)} \mathbf{w}_e}$$

and Item 2 follows by the setting of α.

Edges cut (Item 3). Note that by our choice of v_j and Lemma 16, the $\{F_j\}$ partition E. Since $E(G_i^j) \subseteq F_j$ for all $i \in [K_j]$ and $j \in [j_{\min}, j_{\max}]$ we have that

$$\begin{aligned} \left| E(G) \setminus \left\{ \bigcup_{j_{\min} \leq j \leq j_{\max}, i \in [K_j]} E(G_i^j) \right\} \right| &= \sum_{j_{\min} \leq j \leq j_{\max}} \left| E(G) \setminus \left\{ \bigcup_{i \in [K_j]} E(G_i^j) \right\} \right| \\ &\leq \sum_{j_{\min} \leq j \leq j_{\max}} \left(\frac{4r \ln(n+1)}{\alpha} \sum_{e \in F_j} \mathbf{w}_e \boldsymbol{\ell}_e \right) = \frac{m}{4n} \sum_{e \in E} \mathbf{w}_e \boldsymbol{\ell}_e \end{aligned}$$

where we applied Lemma 16, Item 3 in the inequality. Since $\sum_{e \in E} \mathbf{w}_e \boldsymbol{\ell}_e \leq 2n$ by the definition of a 2-approximate effective resistance overestimate, the result follows.

Vertex coverage (Item 4). Each collection of $\{G_i^j\}_{i \in [K_j]}$ for fixed $j \in [j_{\min}, j_{\max}]$ is vertex-disjoint by Lemma 16. Consequently, each vertex $v \in V(G)$ is in at most $j_{\max} - j_{\min} + 1$ subgraphs and the result follows by our earlier bound (10). □

5 Variance bounds from effective resistance diameter

In this section, we provide an operator norm bound on a matrix variance quantity, used to bound the Gaussian measure of convex bodies induced by operator norm bounds encountered in our sparsification procedures. This variance bound (Lemma 19) is a key new structural insight which enables our applications in the remainder of the paper. In particular, it shows bounded ER diameter of decomposition pieces can be used to control the spectral error incurred by our reweightings.

We first provide a helpful result which upper bounds matrix variances after a projection operation, by the corresponding variance before the projection.

Lemma 17. *Let $\{\mathbf{A}_i\}_{i\in[m]} \in \mathbb{R}^{n\times n}$ and let $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{m\times m}$ be orthogonal projection matrices such that $\ker(\mathbf{Q}) \subseteq \ker(\mathbf{P})$. For each $i \in [m]$, let $\widetilde{\mathbf{A}}_i \stackrel{\text{def}}{=} \sum_{j\in[m]} \mathbf{P}_{ji}\mathbf{A}_j$ and $\widehat{\mathbf{A}}_i \stackrel{\text{def}}{=} \sum_{j\in[m]} \mathbf{Q}_{ji}\mathbf{A}_j$. Then,*

$$\sum_{i\in[m]} \widetilde{\mathbf{A}}_i\widetilde{\mathbf{A}}_i^\top \preceq \sum_{i\in[m]} \widehat{\mathbf{A}}_i\widehat{\mathbf{A}}_i^\top.$$

Proof. Throughout this proof, we denote the Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$ by $\mathbf{A}\otimes\mathbf{B}$. By $\ker(\mathbf{Q}) \subseteq \ker(\mathbf{P})$, we have $\mathbf{P} \preceq \mathbf{Q}$. Define the $n \times mn$ block-partitioned matrices

$$\mathcal{A} \stackrel{\text{def}}{=} \begin{pmatrix}\mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_m\end{pmatrix}, \widetilde{\mathcal{A}} \stackrel{\text{def}}{=} \begin{pmatrix}\widetilde{\mathbf{A}}_1 & \widetilde{\mathbf{A}}_2 & \cdots & \widetilde{\mathbf{A}}_m\end{pmatrix}, \widehat{\mathcal{A}} \stackrel{\text{def}}{=} \begin{pmatrix}\widehat{\mathbf{A}}_1 & \widehat{\mathbf{A}}_2 & \cdots & \widehat{\mathbf{A}}_m\end{pmatrix}.$$

Since $\widetilde{\mathcal{A}} = \mathcal{A}(\mathbf{P}\otimes\mathbf{I}_m)$ and $\widehat{\mathcal{A}} = \mathcal{A}(\mathbf{Q}\otimes\mathbf{I}_m)$ it now suffices to prove $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}^\top \preceq \widehat{\mathcal{A}}\widehat{\mathcal{A}}^\top$. Note that

$$(\mathbf{P}\otimes\mathbf{I}_m)^2 = (\mathbf{P}^2)\otimes(\mathbf{I}_m)^2 = \mathbf{P}\otimes\mathbf{I}_m \preceq \mathbf{Q}\otimes\mathbf{I}_m = (\mathbf{Q})^2\otimes(\mathbf{I}_m)^2 = (\mathbf{Q}\otimes\mathbf{I}_m)^2,$$

where the equality utilizes $\mathbf{P}, \mathbf{Q}$ are orthogonal projection matrices and the inequality holds since since $\mathbf{P} \preceq \mathbf{Q}$. Now utilizing the fact that if $\mathbf{A} \preceq \mathbf{B}$ and $\mathbf{C}$ is any matrix of compatible dimension, then $\mathbf{CAC}^\top \preceq \mathbf{CBC}^\top$ and we get the desired bound that

$$\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}^\top = \mathcal{A}(\mathbf{P}\otimes\mathbf{I}_m)^2\mathcal{A}^\top \preceq \mathcal{A}(\mathbf{Q}\otimes\mathbf{I}_m)^2\mathcal{A}^\top = \widehat{\mathcal{A}}\widehat{\mathcal{A}}^\top.$$

□

We also show that effective resistance decomposition pieces have bounded diagonal entries in an appropriate subgraph inverse Laplacian.

Lemma 18. *For any $G = (V, E, \mathbf{w})$, $U \subseteq V$, and $u \in U$, $\mathbf{e}_u^\top \mathbf{\Pi}_U \mathbf{L}_G^\dagger \mathbf{\Pi}_U \mathbf{e}_u \le \max_{a,b\in U} \mathrm{ER}_G(a,b)$.*

Proof. First, observe that $\mathbf{\Pi}_U\mathbf{e}_u = \mathbf{e}_u - \frac{1}{|U|}\mathbf{1}_U = \frac{1}{|U|}\sum_{v\in U, v\neq u} \mathbf{b}_{(u,v)}$. The conclusion follows from

$$\begin{aligned}\mathbf{e}_u^\top \mathbf{\Pi}_U \mathbf{L}_G^\dagger \mathbf{\Pi}_U \mathbf{e}_u &= \frac{1}{|U|^2}\left(\sum_{v\in U, v\neq u} \mathbf{b}_{(u,v)}\right)^\top \mathbf{L}_G^\dagger \left(\sum_{v\in U, v\neq u} \mathbf{b}_{(u,v)}\right) \\ &\le \frac{|U|-1}{|U|^2}\sum_{v\in U, v\neq u} \mathbf{b}_{(u,v)}^\top \mathbf{L}_G^\dagger \mathbf{b}_{(u,v)} \le \frac{(|U|-1)^2}{|U|^2}\max_{a,b\in U} \mathrm{ER}_G(a,b),\end{aligned}$$

where the first inequality was the Cauchy-Schwarz inequality. □

We now combine Fact 6, Lemma 17, and Lemma 18 to obtain the main result of this section.

Lemma 19. *Let $\vec{G} = (V, E, \mathbf{w})$ and let $\vec{H}$ be a subgraph on vertex set $U \subseteq V$. Suppose that for $\rho > 0$, $(\max_{e\in E(\vec{H})} \mathbf{w}_e)\cdot(\max_{u,v\in U} \mathrm{ER}_G(u,v)) \le \rho$. Define*

$$\begin{aligned}\mathbf{L}_{H^2} &\stackrel{\text{def}}{=} \mathbf{B}_{\vec{H}}^\top \mathbf{W}_{E(\vec{H})}^2 \mathbf{B}_{\vec{H}}, \\ \mathbf{P}_{\vec{H}} &\stackrel{\text{def}}{=} \mathbf{I}_{E(\vec{H})} - \mathbf{W}_{E(\vec{H})}\mathbf{B}_{\vec{H}}\mathbf{L}_{H^2}^\dagger \mathbf{B}_{\vec{H}}^\top \mathbf{W}_{E(\vec{H})}, \\ \widetilde{\mathbf{A}}_e &\stackrel{\text{def}}{=} \mathbf{L}_G^{\frac{\dagger}{2}}\left(\sum_{f\in E(\vec{H})} [\mathbf{P}_{\vec{H}}]_{fe}\mathbf{w}_f\mathbf{b}_f\mathbf{e}_{h(f)}^\top\right)\mathbf{L}_G^{\frac{\dagger}{2}} \text{ for all } e \in E(\vec{H}),\end{aligned} \tag{11}$$

where $\mathbf{W}_{E(\vec{H})}$, $\mathbf{I}_{E(\vec{H})}$ *zero out entries of* $\mathbf{W}_{\vec{G}}$, $\mathbf{I}_{E(\vec{G})}$ *not corresponding to edges in* $E(\vec{H})$. *Then,*

$$\sum_{e\in E(\vec{H})} \widetilde{\mathbf{A}}_e\widetilde{\mathbf{A}}_e^\top \preccurlyeq \rho\cdot \mathbf{L}_G^{\dagger\frac{1}{2}}\mathbf{L}_H\mathbf{L}_G^{\dagger\frac{1}{2}},\quad \sum_{e\in E(\vec{H})} \widetilde{\mathbf{A}}_e^\top\widetilde{\mathbf{A}}_e \preccurlyeq \rho\cdot \mathbf{L}_G^{\dagger\frac{1}{2}}\mathbf{L}_H\mathbf{L}_G^{\dagger\frac{1}{2}},$$

$$\text{where } G \stackrel{\text{def}}{=} \text{und}(\vec{G}),\ H\stackrel{\text{def}}{=}\text{und}(\vec{H}).$$

Proof. For simplicity, we write $\mathbf{W}_{\vec{H}} = \mathbf{W}_{E(\vec{H})}$ and $\mathbf{B}_{\vec{H}} = \mathbf{B}_{\vec{G}}\mathbf{I}_{E(\vec{H})}$. We first note that $\mathbf{P}_{\vec{H}}$ is a orthogonal projection matrix, since $\mathbf{W}_{\vec{H}}\mathbf{B}_{\vec{H}}\mathbf{L}_{H^2}^\dagger\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}$ is an orthogonal projection on the restriction to $\vec{H}$. This justifies our notation: the $\widetilde{\mathbf{A}}_e$ are as in Lemma 17, where $\mathbf{A}_e \stackrel{\text{def}}{=} \mathbf{L}_G^{\dagger/2}\mathbf{w}_e\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{L}_G^{\dagger/2}$. Next, let $\mathbf{x}_e \stackrel{\text{def}}{=} [\mathbf{P}_{\vec{H}}]_{e:}$ and $\mathbf{X}_e \stackrel{\text{def}}{=} \mathbf{diag}\,(\mathbf{x}_e)$, so $\widetilde{\mathbf{A}}_e = \mathbf{L}_G^{\dagger/2}\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{X}_e\mathbf{H}_{\vec{H}}\mathbf{L}_G^{\dagger/2}$. Since $\mathbf{P}_{\vec{H}}$ is an orthogonal projection matrix,

$$\mathbf{P}_{\vec{H}}\mathbf{x}_e = \mathbf{x}_e \implies \mathbf{W}_{\vec{H}}\mathbf{B}_{\vec{H}}\mathbf{L}_{H^2}^\dagger\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{x}_e = \mathbf{0}_V \implies \mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{x}_e = \mathbf{0}_V.$$

To see the last implication, note that $\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{x}_e$ is always orthogonal to the kernel of $\mathbf{L}_{H^2} = \mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{B}_{\vec{H}}$. The last equality then follows by noticing that $\ker(\mathbf{L}_{H^2}) = \ker(\mathbf{L}_{H^2}^\dagger)$. In other words, $\mathbf{W}_{\vec{H}}\mathbf{x}_e$ is a circulation on $\vec{H}$. Since $\mathbf{\Pi}_U$ is the projection onto the coordinates of U orthogonal to $\mathbf{1}_U$, by $\ker(\mathbf{H}_{\vec{H}}) \supseteq \text{span}(\mathbf{1}_U)\cup\mathbb{R}^{V\setminus U}$, we further have

$$\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{X}_e\mathbf{H}_{\vec{H}} = \mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{X}_e\mathbf{H}_{\vec{H}}\mathbf{\Pi}_U \implies \widetilde{\mathbf{A}}_e = \mathbf{L}_G^{\dagger\frac{1}{2}}\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{X}_e\mathbf{H}_{\vec{H}}\mathbf{\Pi}_U\mathbf{L}_G^{\dagger\frac{1}{2}}.$$

Applying Lemma 17 to $\{\widetilde{\mathbf{A}}_e\}_{e\in E(\vec{H})}$ using the characterization in the above display then gives

$$\begin{aligned}\sum_{e\in E(\vec{H})} \widetilde{\mathbf{A}}_e\widetilde{\mathbf{A}}_e^\top &\preccurlyeq \mathbf{L}_G^{\dagger\frac{1}{2}}\left(\sum_{e\in E(\vec{H})} w_e^2\cdot \mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{\Pi}_U\mathbf{L}_G^\dagger\mathbf{\Pi}_U\mathbf{e}_{h(e)}\mathbf{b}_e^\top\right)\mathbf{L}_G^{\dagger\frac{1}{2}}\\ &\preccurlyeq \mathbf{L}_G^{\dagger\frac{1}{2}}\left(\sum_{e\in E(\vec{H})}\rho w_e\cdot\mathbf{b}_e\mathbf{b}_e^\top\right)\mathbf{L}_G^{\dagger\frac{1}{2}} = \rho\cdot\mathbf{L}_G^{\dagger\frac{1}{2}}\mathbf{L}_H\mathbf{L}_G^{\dagger\frac{1}{2}}.\end{aligned}$$

The second inequality follows from Lemma 18 and the $w_e \le \max_{e\in E(\vec{H})} w_e$. This yields the first claim. To see the second, since $\mathbf{W}_{\vec{H}}\mathbf{x}_e$ is a circulation, by Fact 6, $\widetilde{\mathbf{A}}_e = -\mathbf{L}_G^{\dagger/2}\mathbf{T}_{\vec{H}}^\top\mathbf{W}_{\vec{H}}\mathbf{X}_e\mathbf{B}_{\vec{H}}\mathbf{L}_G^{\dagger/2}$. By instead applying Lemma 17 to the matrices $\{-\widetilde{\mathbf{A}}_e^\top\}_{e\in E(\vec{H})}$ (as $\mathbf{X}_e\mathbf{T}_{\vec{H}}\mathbf{1}_U = \mathbf{X}_e\mathbf{H}_{\vec{H}}\mathbf{1}_U = \mathbf{x}_e$) and following an analogous derivation, we obtain the desired bound. □

6 Sparser Eulerian sparsifiers

In this section, we give the first application of our framework by proving our Eulerian sparsification result obtaining the best-known sparsity bound in Theorem 4. This application serves as a warmup for our nearly-linear time sparsification result in Section 7.

Our approach is to recursively apply Lemma 19 on each subgraph component in a ER decomposition (Proposition 10), with known results from the literature on discrepancy theory, to sparsify an Eulerian graph. Specifically, our main tools are a powerful matrix discrepancy Gaussian measure lower bound recently developed by [BJM23] (motivated by the matrix Spencer conjecture), and a corresponding partial coloring framework from [Rot17; RR23].

Proposition 20 (Proof of Lemma 3.1, [BJM23]). *For every constant $c \in (0, \frac{1}{2})$, there is a constant C_{color} such that for any $\{\widetilde{\mathbf{A}}_i\}_{i\in[m]} \subset \mathbb{S}^n$ with $m > 2n$ that satisfy $\|\sum_{i\in[m]} \widetilde{\mathbf{A}}_i^2\|_{\text{op}} \le \sigma^2$, $\sum_{i\in[m]} \|\widetilde{\mathbf{A}}_i\|_{\text{F}}^2 \le mf^2$, and letting*

$$\widetilde{\mathcal{K}} \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^m \middle| \left\| \sum_{i\in[m]} \mathbf{x}_i \widetilde{\mathbf{A}}_i \right\|_{\text{op}} \le C_{\text{color}} \min\left\{ \sigma + \sqrt{\sigma f} \log^{\frac{3}{4}} n,\ \sigma \log^{\frac{1}{4}} n + \sqrt{\sigma f} \log^{\frac{1}{2}} n \right\} \right\},$$

there is a subspace $T \subseteq \mathbb{R}^m$ with $\dim(T) \ge (1-c)m$, $\gamma_T(\mathcal{K}) \ge \exp(-cm)$.

We note that the proof of Lemma 3.1 in [BJM23] only showed how to obtain the first of the two operator norm upper bounds within the min expression in Proposition 20, but the second follows straightforwardly by substituting an alternative matrix concentration inequality from [Tro18] into the same proof of [BJM23]. We formally show how to obtain the second bound in Appendix D.

Proposition 21 (Theorem 6, [RR23]). *Let $c_{\text{tight}} \in (0,1)$ be a constant, let $S \subseteq \mathbb{R}^m$ be a subspace with $\dim(S) \ge 2c_{\text{tight}}m$, and let $\mathcal{K} \subseteq \mathbb{R}^m$ be symmetric and convex. Suppose $\gamma_m(\mathcal{K}) \ge \exp(-Cm)$ for a constant C. There is $C_{\text{set}} > 0$ depending only on c_{tight}, C such that if $\mathbf{g} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$, and*

$$\mathbf{x} \stackrel{\text{def}}{=} \arg \min_{\mathbf{x} \in C_{\text{set}}\mathcal{K} \cap [-1,1]^m \cap S} \|\mathbf{x} - \mathbf{g}\|_2,$$

then $|\{i \in [m] \mid |\mathbf{x}_i| = 1\}| \ge c_{\text{tight}}m$ with probability $1 - \exp(-\Omega(m))$.

Roughly speaking, Proposition 20 shows that a convex body over $\mathbf{x} \in \mathbb{R}^m$, corresponding to a sublevel set of $\|\sum_{i\in[m]} \mathbf{x}_i \widetilde{\mathbf{A}}_i\|_{\text{op}}$, has large Gaussian measure restricted to a subspace. Proposition 21 then produces a "partially colored" point $[-1,1]^m$ with many tight constraints, i.e., coordinates $i \in [m]$ with $|\mathbf{x}_i| = 1$, which also lies in the convex body from Proposition 20. We summarize a useful consequence of Proposition 21 that is more compatible with Proposition 20. The difference is that the variant in Corollary 22 only requires a Gaussian measure lower bound on the convex body restricted to a subspace, the type of guarantee that Proposition 20 gives.

Corollary 22. *In the setting of Proposition 21, assume that $\gamma_S(\mathcal{K}) \ge \exp(-Cm)$ for a constant C, instead of $\gamma_m(\mathcal{K}) \ge \exp(-Cm)$. There is $C_{\text{set}} > 0$ depending only on c_{tight}, C such that if $\mathbf{g} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$, and*

$$\mathbf{x} \stackrel{\text{def}}{=} \arg \min_{\mathbf{x} \in C_{\text{set}}\mathcal{K} \cap [-1,1]^m \cap S} \|\mathbf{x} - \mathbf{g}\|_2,$$

then $|\{i \in [m] \mid |\mathbf{x}_i| = 1\}| \ge c_{\text{tight}}m$ with probability $1 - \exp(-\Omega(m))$.

Proof. Define $\mathcal{K}' \subseteq \mathbb{R}^m$ to be $\mathcal{K} \cap S$ expanded by a hypercube (centered at the origin and with side length 2) in the subspace orthogonal to S, denoted $S_\perp$; concretely, let $\mathcal{K}' \stackrel{\text{def}}{=} (\mathcal{K} \cap S) \oplus (\mathbf{P}_{S_\perp})[-1,1]^{\dim(S_\perp)}$, where $\oplus$ denotes the direct sum of two sets. Note that $\mathcal{K}'$ is symmetric and convex, and $\gamma_m(\mathcal{K}') \ge \exp(-C'm)$ for a constant C' depending only on C and the universal constant $\gamma_1([-1,1])$, since the probability $\mathbf{g} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$ falls in $\mathcal{K}'$ is the product of the probabilities of the independent events $\mathbf{g} \in \mathcal{K}' \cap S$ and $\mathbf{g} \in \mathcal{K}' \cap S_\perp$. Therefore, applying Proposition 21 to the subspace S and the set $\mathcal{K}'$ yields the conclusion, as $C_{\text{set}}\mathcal{K}' \cap S = C_{\text{set}}\mathcal{K} \cap S$. □

Finally, we give an equivalence we will later use.

Lemma 23. *For $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}^n$, a subspace $S \subseteq \mathbb{R}^m$ and a parameter $R \geq 0$, define*

$$\widetilde{\mathbf{A}}_i \stackrel{\text{def}}{=} \sum_{j\in[m]} [\mathbf{P}_S]_{ji} \mathbf{A}_j \text{ for all } i \in [m],$$

and their induced operator norm bodies

$$\mathcal{K} \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^m \middle| \left\| \sum_{i\in[m]} \mathbf{x}_i \mathbf{A}_i \right\|_{\text{op}} \leq R \right\}, \; \widetilde{\mathcal{K}} \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^m \middle| \left\| \sum_{i\in[m]} \mathbf{x}_i \widetilde{\mathbf{A}}_i \right\|_{\text{op}} \leq R \right\}.$$

Then $\mathcal{K} \cap T = \widetilde{\mathcal{K}} \cap T$ for any subspace $T \subseteq S$.

Proof. It suffices to note that for $\mathbf{x} \in T$, $\mathbf{P}_S \mathbf{x} = \mathbf{x}$ and therefore

$$\sum_{i\in[m]} \mathbf{x}_i \widetilde{\mathbf{A}}_i = \sum_{i\in[m]} \sum_{j\in[m]} [\mathbf{P}_S]_{ji} \mathbf{x}_i \mathbf{A}_j = \sum_{j\in[m]} [\mathbf{P}_S \mathbf{x}]_j \mathbf{A}_j = \sum_{j\in[m]} \mathbf{x}_j \mathbf{A}_j.$$

This shows that for $\mathbf{x} \in T$, $\|\sum_{i\in[m]} \mathbf{x}_i \widetilde{\mathbf{A}}_i\|_{\text{op}} \leq R \iff \|\sum_{i\in[m]} \mathbf{x}_i \mathbf{A}_i\|_{\text{op}} \leq R$, so $\mathcal{K} \cap T = \widetilde{\mathcal{K}} \cap T$. □

Next, we state a guarantee on a degree-rounding algorithm, ROUNDING. This algorithm is used in all of our sparsification subroutines, to deal with small degree imbalances induced by approximation errors in projection operations. The algorithm (Algorithm 1) follows a standard approach of rerouting the vertex imbalances $\mathbf{B}_{\vec{G}}^\top \mathbf{z}$ through a spanning tree. We bound the incurred discrepancy in the directed Laplacian by the size of $\mathbf{z}$. This procedure is related to, and inspired by, other tree-based rounding schemes in the literature, see e.g., [KOSZ13].

Algorithm 1: ROUNDING$(\vec{G}, \mathbf{z}, T)$

Input: $\vec{G} = (V, E, \mathbf{w})$, $\mathbf{z} \in \mathbb{R}^E$, T a tree subgraph of $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$
$\mathbf{d} \leftarrow \mathbf{B}_{\vec{G}}^\top \mathbf{z}$
$\mathbf{y} \leftarrow$ unique vector in $\mathbb{R}^E$ with $\text{supp}(\mathbf{y}) \subseteq E(T)$ and $\mathbf{B}_{\vec{G}}^\top \mathbf{y} = \mathbf{d}$
return $\mathbf{y}$

Lemma 24. *Given $\vec{G} = (V, E, \mathbf{w})$, a tree subgraph T of $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$ with $\min_{e\in E(T)} \mathbf{w}_e \geq 1$,* ROUNDING *(Algorithm 1) returns in $O(n)$ time $\mathbf{y} \in \mathbb{R}^E$ with $\text{supp}(\mathbf{y}) \subseteq T$ satisfying:*

1. $\mathbf{B}_{\vec{G}}^\top \mathbf{y} = \mathbf{d}$.
2. $\|\mathbf{y}\|_\infty \leq \frac{1}{2} \|\mathbf{d}\|_1$.
3. *For any $\mathbf{z} \in \mathbb{R}^E$ satisfying $\mathbf{B}_{\vec{G}}^\top \mathbf{z} = \mathbf{d}$, we have $\|\mathbf{L}_G^{\dagger/2} \mathbf{B}_{\vec{G}}^\top (\mathbf{Y} - \mathbf{Z}) \mathbf{H}_{\vec{G}} \mathbf{L}_G^{\dagger/2}\|_{\text{op}} \leq n \|\mathbf{z}\|_1$.*
4. $\|\mathbf{L}_G^{\dagger/2} \mathbf{B}_{\vec{G}}^\top \mathbf{Y} \mathbf{H}_{\vec{G}} \mathbf{L}_G^{\dagger/2}\|_{\text{op}} \leq n \|\mathbf{y}\|_1$.

A proof of Lemma 24 is deferred to Appendix B.

We next show how to combine Corollary 22 with our variance bound in Lemma 19 to slightly sparsify an Eulerian graph, while incurring small operator norm discrepancy.

Algorithm 2: ExistentialDecompSparsify($\{\vec{G}_i\}_{i\in[I]}, \vec{G}, T, \varepsilon, W$)

Input: $\{\vec{G}^{(i)}\}_{i\in[I]}$, subgraphs of simple $\vec{G} = (V, E, \mathbf{w})$ with $\max_{e\in\mathrm{supp}(\mathbf{w})} \mathbf{w}_e \le W$, and such that $\{G^{(i)} \stackrel{\text{def}}{=} \mathrm{und}(\vec{G}^{(i)})\}_{i\in[I]}$ are a $(\rho, 2, J)$-ER decomposition of $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$, T a tree subgraph of G with $\min_{e\in E(T)} \mathbf{w}_e \ge 1$ and $E(T) \cap \bigcup_{i\in I} E(\vec{G}^{(i)}) = \emptyset$, $\delta, \varepsilon \in (0, \frac{1}{100})$

$\hat{m} \leftarrow \mathrm{nnz}(\mathbf{w})$, $\hat{E} \leftarrow \mathrm{supp}(\mathbf{w})$, $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$, $n \leftarrow |V|$

if $\hat{m} \ge 8nJ$ **then**

  $S_i \leftarrow \{\mathbf{x} \in \mathbb{R}^{\hat{E}} \mid \mathrm{supp}(\mathbf{x}) \subseteq E(\vec{G}^{(i)}), \mathbf{B}_{\vec{G}^{(i)}}^\top \mathbf{W}_{\vec{G}_i} \mathbf{x} = \mathbf{0}_V\}$ for all $i \in [I]$

  $S \leftarrow \bigcup_{i\in[I]} S_i$

  $\mathbf{x} \leftarrow$ point in $[-1, 1]^{\hat{E}} \cap S$ such that for universal constants C_{ESO}, c_{tight},

$$\begin{aligned} &\left\| \mathbf{L}_G^{\dagger\frac{1}{2}} \left(\sum_{i\in I} \sum_{e\in E(\vec{G}_i)} \mathbf{x}_e \mathbf{w}_e \mathbf{b}_e \mathbf{e}_{h(e)}^\top \right) \mathbf{L}_G^{\dagger\frac{1}{2}} \right\|_{\mathrm{op}} \\ &\le C_{\mathrm{ESO}} \min\left\{ \rho^{\frac{1}{2}} \log^{\frac{1}{2}} + \rho^{\frac{3}{4}} \log^{\frac{5}{4}}(n),\ \rho^{\frac{1}{2}} \log^{\frac{3}{4}}(n) + \rho^{\frac{3}{4}} \log(n) \right\}, \\ &\left| \left\{ e \in \hat{E} \mid \mathbf{x}_e = -1 \right\} \right| \ge c_{\mathrm{tight}} \hat{m} \end{aligned} \tag{12}$$

  ▷ Existence of $\mathbf{x}$, C_{ESO}, c_{tight} follow from Lemma 19, Proposition 20, and Corollary 22, see Lemma 25.

  $\mathbf{x}' \leftarrow$ extension of $\mathbf{x}$ to $\mathbb{R}^E$ with $\mathbf{x}'_e = \mathbf{x}_e$ if $e \in \bigcup_{i\in I} E(G_i)$ and $\mathbf{x}'_e = 0$ otherwise

  $\mathbf{w} \leftarrow \mathbf{w} \circ (\mathbf{1}_E + \mathbf{x}')$

$D \stackrel{\text{def}}{=} \{e \in \hat{E} \mid \mathbf{w}_e \le \ell\}$

return $\vec{G}' \leftarrow (V, E, [\mathbf{w}]_{\hat{E}\setminus D} + \mathrm{Rounding}(\vec{G}, [\mathbf{w}]_D, T))$

Lemma 25. *Suppose that* ExistentialDecompSparsify *(Algorithm 2) is run on inputs as specified in Line 1. Then, it returns $\vec{G}' = (V, E, \mathbf{w}')$ satisfying the following properties, with probability $\ge 1 - \delta$.*

1. $\max_{e\in\hat{E}} \mathbf{w}'_e \le 2W$, $\min_{e\in\hat{E}} \mathbf{w}_e > \ell$ *and* $\min_{e\in E(T)} \mathbf{w}'_e \ge \min_{e\in E(T)} \mathbf{w}_e - n\hat{m}\ell$.
2. $\mathbf{B}_{\vec{G}}^\top \mathbf{w}' = \mathbf{B}_{\vec{G}}^\top \mathbf{w}$.
3. $\mathrm{nnz}([\mathbf{w}']_{\hat{E}}) \le (1 - c_{\mathrm{tight}})\hat{m} + C_{\mathrm{ESO}} \cdot nJ$.
4. $$\begin{aligned} \|\mathbf{L}_G^{\dagger/2} \mathbf{B}_{\vec{G}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{G}} \mathbf{L}_G^{\dagger/2}\|_{\mathrm{op}} \le\ & C_{\mathrm{ESO}} \min\{\rho^{\frac{1}{2}} \log^{\frac{1}{2}} n + \rho^{\frac{3}{4}} \log^{\frac{5}{4}} n, \\ & \rho^{\frac{1}{2}} \log^{\frac{3}{4}} n + \rho^{\frac{3}{4}} \log n\} + n\hat{m}\ell. \end{aligned}$$

Moreover, ExistentialDecompSparsify *is implementable in* $\mathrm{poly}(n, \log U, \log \frac{1}{\delta})$ *time.*

Proof. If Line 3 does not pass, then Items 1 to 3 trivially hold and it only incurs the second term in the spectral error (Item 4) due to Lemma 24. We then assume it does pass for the remainder of the proof. We defer proving existence of $\mathbf{x}, C_{\mathrm{ESO}}, c_{\mathrm{tight}}$ in (12) until the end. Since $\mathbf{x} \in [-1, 1]^E$ and $\mathrm{supp}(\mathbf{x}) \subseteq \hat{E}$, no edge weight in $\hat{E}$ more than doubles, giving the first claim of Item 1. Our

definition of D on Line 9 and ROUNDING ensures the second claim of Item 1. Next, since $\mathbf{w} \circ \mathbf{x}$ is only supported on $E' \stackrel{\text{def}}{=} \bigcup_{i \in [I]} E(G^{(i)})$ and $[\mathbf{w} \circ \mathbf{x}]_{E'}$ is the sum of disjoint circulations on each G_i by the definition of each S_i, $\mathbf{w} \circ \mathbf{x}$ is itself a circulation on G. Combining with the first guarantee of Lemma 24, this implies Item 2. Since any $e \in \hat{E}$ where $\mathbf{x}_e = -1$ necessary has $\mathbf{w}_e(1 + \mathbf{x}_e) = 0$ and that ROUNDING only introduces new non-zero entries on $E(T)$, Item 3 holds. Item 4 is follows from the definitions of $\mathbf{w}'$ and D, (12) and the third guarantee of Lemma 24.

It remains to prove $\mathbf{x}, C_{\text{ESO}}, c_{\text{tight}}$ exist when Line 3 passes. For each $e \in E'$, define $\mathbf{A}_e$ and $\widetilde{\mathbf{A}}_e$ as in the proof of Lemma 19 where $\vec{H}$ is set to the partition piece $\vec{G}^{(i)}$ with $E(\vec{G}^{(i)}) \ni e$. Summing the bound in Lemma 19 over all pieces gives $\sigma^2 = \rho$ in Proposition 20, where we overload

$$\widetilde{\mathbf{A}}_e \leftarrow \begin{pmatrix} & \widetilde{\mathbf{A}}_e \\ \widetilde{\mathbf{A}}_e^\top & \end{pmatrix}$$

in its use (padding with zeroes as necessary). Correctness follows from the observations

$$\begin{pmatrix} & \mathbf{A} \\ \mathbf{A}^\top & \end{pmatrix}^2 = \begin{pmatrix} \mathbf{A}\mathbf{A}^\top & \\ & \mathbf{A}^\top\mathbf{A} \end{pmatrix}, \left\| \begin{pmatrix} & \mathbf{A} \\ \mathbf{A}^\top & \end{pmatrix} \right\|_{\text{op}} = \|\mathbf{A}\|_{\text{op}}.$$

Further, we always have $f^2 \leq \frac{n\sigma^2}{m}$ by linearity of trace and $\|\widetilde{\mathbf{A}}\|_{\text{F}}^2 = \text{Tr}(\widetilde{\mathbf{A}}^2)$ for $\widetilde{\mathbf{A}} \in \mathbb{S}^n$. This gives a Gaussian measure lower bound on $\widetilde{\mathcal{K}}$ restricted to a subspace S' of S. By the characterization in Lemma 23, this also implies a Gaussian measure lower bound on $\mathcal{K}$ restricted to S'. We next observe that S is a subspace of $\mathbb{R}^{E'}$ where each S_i enforces $|V(G^i)| - 1$ linear constraints (corresponding to weighted degrees in the subgraph). By Definition 9, the total number of such linear constraints is $\leq nJ$ and $|E'| \geq \frac{1}{2}\hat{m}$. The condition on Line 3 then guarantees our final subspace has sufficiently large dimension to apply Corollary 22. Finally, Corollary 22 guarantees existence of $\mathbf{x}, c_{\text{tight}}, C_{\text{ESO}}$ satisfying the guarantees in (12) (we may negate $\mathbf{x}$ if it has more 1s than -1s, and halve c_{tight}).

Lastly, we observe that Algorithm 2 is implementable in polynomial time. This is clear for ROUNDING and Line 7 to 9. The most computationally intensive step is Line 6, which consists of finding a subspace of large Gaussian measure and solving a convex program. The latter is polynomial time [GLS88]; the former is due to intersecting the explicit subspace from Line 5 and the subspace from Proposition 20. The subspace from Proposition 20 is explicitly described in the proof of Lemma 3.1 of [BJM23]; it is an eigenspace of a flattened second moment matrix.

All steps are deterministic except for the use of Corollary 22 in Line 6 (note that we can bypass Lemma 13 via exact linear algebra computations). This line succeeds with probability $\geq \frac{1}{2}$ for a random draw. Finally, we can boost this line to have failure probability δ by running $\log(\frac{1}{\delta})$ independent trials, as we can verify whether a run succeeds in $\text{poly}(n, \log U)$ time. □

We are now ready to state and analyze our overall sparsification algorithm, EXISTENTIALSPARSIFY (Algorithm 3). The following is a refined version of Theorem 4.

Theorem 26. *Given Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, U]^E$ and $\varepsilon \in (0, 1)$,* EXISTENTIALSPARSIFY *(Algorithm 3) returns Eulerian $\vec{H}$ such that $\vec{H}$ is an ε-approximate Eulerian sparsifier of $\vec{G}$, and*

$$|\vec{H}| = O\left(n \log U + \frac{n \log n}{\varepsilon^2} \min\left\{1 + (\varepsilon \log n)^{\frac{2}{3}}, \log^{\frac{1}{2}} n\right\}\right),$$

$$\log\left(\frac{\max_{e \in \text{supp}(\mathbf{w}')} \mathbf{w}'_e}{\min_{e \in \text{supp}(\mathbf{w}')} \mathbf{w}'_e}\right) = O\left(\log\left(nU\right)\right).$$

EXISTENTIALSPARSIFY *succeeds with probability $\geq 1 - \delta$ and runs in time* $\text{poly}(n, \log U, \log \frac{1}{\delta})$.

Algorithm 3: ExistentialSparsify($\vec{G}, \varepsilon, \delta$)

Input: Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $\mathbf{w}_e \in [1, U]$ for all $e \in E$, $\varepsilon \in (0, 1)$
$n \leftarrow |V|$, $m \leftarrow |E|$
$T \leftarrow$ arbitrary spanning tree of $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$, $\hat{E} \leftarrow E \setminus E(T)$
$R \leftarrow \lfloor \log_{1-\frac{1}{2}c_{\text{tight}}} \frac{1}{n} \rfloor + 1$, $C_1 \leftarrow (\frac{256 C_{\text{ESO}}}{c_{\text{tight}}})^2, C_2 \leftarrow (\frac{256 C_{\text{ESO}}}{c_{\text{tight}}})^{4/3}, C_3 \leftarrow (\frac{256 C_{\text{ESO}}}{c_{\text{tight}}})^{3/2}$ for $c_{\text{tight}}, C_{\text{ESO}}$ in (12)
$U_{\max} \leftarrow U \cdot 2^R$, $J_{\max} \leftarrow \log_2 \left(\frac{64 mnRU_{\max}}{\varepsilon}\right)$
$t \leftarrow 0$, $\vec{G}_0 \leftarrow \vec{G}$
while $\text{nnz}([\mathbf{w}_t]_{\hat{E}}) > \max\{2C_{\text{ESO}} c_{\text{tight}} \cdot nJ_{\max}, \min\{C_1 \cdot \frac{n \log n}{\varepsilon^2} + C_2 \cdot \frac{n \log^{5/3} n}{\varepsilon^{4/3}}, 2C_3 \cdot \frac{n \log^{3/2} n}{\varepsilon^2}\}\}$ **do**
  $G_t \leftarrow \text{und}(\vec{G}_t)$
  $S \leftarrow \text{ERDecomp}([G_t]_{\hat{E}}, 2, \frac{\delta}{R})$ ▷ See Proposition 10.
  $\vec{G}_{t+1} \stackrel{\text{def}}{=} (V, E, \mathbf{w}_{t+1}) \leftarrow \text{ExistentialDecompSparsify}(S, G_t, T, \frac{\varepsilon}{8mnR}, U_{\max})$
  $t \leftarrow t + 1$
return $\vec{H} \leftarrow (V, \text{supp}(\mathbf{w}_t), \mathbf{w}_t)$

Proof. Recall from Section 2 that we assume without loss of generality that G is connected. Throughout, condition on the event that all of the at most R calls to ERDecomp succeed, which happens with probability $\geq 1 - \delta$. Because ExistentialDecompSparsify guarantees that no weight grows by more than a 2 factor in each call, $U_{\max}$ is a valid upper bound for the maximum weight of any edge throughout the algorithm's execution. Moreover, since no weight falls below $\frac{\varepsilon}{8mnR}$ throughout by ExistentialDecompSparsify, $J_{\max} \stackrel{\text{def}}{=} \log_2(\frac{64 mnRU_{\max}}{\varepsilon})$ is an upper bound on the number of decomposition pieces ever returned by ERDecomp, by Proposition 10.

Next, note that under the given lower bound on $\text{nnz}([\mathbf{w}_t]_{\hat{E}})$ in a given iteration (which is larger than $2C_{\text{ESO}} c_{\text{tight}} \cdot nJ_{\max}$), the sparsity progress guarantee in Item 3 of Lemma 25 shows that the number of edges in each iteration is decreasing by at least a $(1 - c_{\text{tight}}) + \frac{1}{2}c_{\text{tight}} = (1 - \frac{1}{2}c_{\text{tight}})$ factor until termination. Since $m \leq n^2$ and the algorithm terminates before reaching n edges, R is a valid upper bound on the number of iterations before the second condition in Line 7 fails to hold, which gives the sparsity claim.

Let $\hat{m}_i \stackrel{\text{def}}{=} \text{nnz}([\mathbf{w}]_{\hat{E}})$. To prove the spectral error bound, we show by induction that until the algorithm terminates, the following conditions hold, where we use t to denote the number of times the while loop runs in total:

1. $\mathbf{B}^\top \mathbf{w}_i = \mathbf{B}^\top \mathbf{w}_0$
2. $\hat{m}_i \leq (1 - \frac{1}{2}c_{\text{tight}})^i \hat{m}_0$.
3. $\|\mathbf{L}^{\dagger/2} \mathbf{B}^\top (\mathbf{W}_i - \mathbf{W}_0) \mathbf{H} \mathbf{L}^{\dagger/2}\|_{\text{op}} \leq 2C_{\text{ESO}} \sum_{j=0}^{i-1} \min\{(\frac{n \log n}{\hat{m}_j})^{\frac{1}{2}} + (\frac{n}{\hat{m}_j})^{\frac{3}{4}} \log^{\frac{5}{4}} n, (\frac{n}{\hat{m}_j})^{\frac{1}{2}} \log^{\frac{3}{4}} n + (\frac{n}{\hat{m}_j})^{\frac{3}{4}} \log n\} + \frac{i}{4R}$.

Note that Items 1 to 3 all hold trivially for $i = 0$. Suppose inductively all conditions above hold for all iterations $k \leq i < t$. By our stopping condition, $n \leq \hat{m}_i \leq (1 - \frac{1}{2}c_{\text{tight}})^{i-1} \hat{m}_0$ and hence $i \leq \frac{\log n}{-\log(1 - \frac{1}{2}c_{\text{tight}})} < R$. Items 2 and 3 of Lemma 25 then implies Items 1 and 2 are satisfied for

iteration $i+1$. We also have by Item 4 of Lemma 25 that

$$\left\|\mathbf{L}_{G_i}^{\frac{\dagger}{2}}\mathbf{B}^\top(\mathbf{W}_{i+1}-\mathbf{W}_i)\mathbf{H}\mathbf{L}_{G_i}^{\frac{\dagger}{2}}\right\|_{\text{op}}$$

$$\leq C_{\text{ESO}}\min\left\{\sqrt{\frac{n\log n}{\hat{m}_i}}+\left(\frac{n\log^{\frac{5}{3}}n}{\hat{m}_i}\right)^{\frac{3}{4}},\ \sqrt{\frac{n}{\hat{m}_i}}\log^{\frac{3}{4}}(n)+\left(\frac{n}{\hat{m}_i}\right)^{\frac{3}{4}}\log(n)\right\}+\frac{\varepsilon}{8R},$$

where we define $\vec{G}_i \stackrel{\text{def}}{=} (V,E,\mathbf{w}_i)$ and $G_i \stackrel{\text{def}}{=} \text{und}(\vec{G}_i)$ for any $0\leq i\leq t$. Note that $\vec{G}_0=(V,E,\mathbf{w}_0)=\vec{G}$, the original input Eulerian graph. Moreover, $\mathbf{L}_{\vec{G}_i}-\mathbf{L}_{\vec{G}_0}=\mathbf{B}^\top(\mathbf{W}_i-\mathbf{W}_0)\mathbf{H}$. By our choice of C_1,C_2, the stopping condition, Item 2, and Lemma 25,

$$2C_{\text{ESO}}\sum_{j=0}^{i-1}\min\left\{\sqrt{\frac{n\log n}{\hat{m}_j}}+\left(\frac{n\log^{\frac{5}{3}}n}{\hat{m}_j}\right)^{\frac{3}{4}},\ \sqrt{\frac{n}{\hat{m}_j}}\log^{\frac{3}{4}}(n)+\left(\frac{n}{\hat{m}_j}\right)^{\frac{3}{4}}\log(n)\right\}$$

$$\leq 2C_{\text{ESO}}\sum_{j=0}^{i-1}\left(1-\frac{c_{\text{tight}}}{4}\right)^{i-1-j}$$

$$\cdot\min\left\{\sqrt{\frac{n\log n}{\hat{m}_{i-1}}}+\left(\frac{n\log^{\frac{5}{3}}n}{\hat{m}_{i-1}}\right)^{\frac{3}{4}},\ \sqrt{\frac{n}{\hat{m}_{i-1}}}\log^{\frac{3}{4}}(n)+\left(\frac{n}{\hat{m}_{i-1}}\right)^{\frac{3}{4}}\log(n)\right\}$$

$$\leq\frac{8C_{\text{ESO}}}{c_{\text{tight}}}\min\left\{\sqrt{\frac{n\log n}{\hat{m}_{i-1}}}+\left(\frac{n\log^{\frac{5}{3}}n}{\hat{m}_{i-1}}\right)^{\frac{3}{4}},\ \sqrt{\frac{n}{\hat{m}_{i-1}}}\log^{\frac{3}{4}}(n)+\left(\frac{n}{\hat{m}_{i-1}}\right)^{\frac{3}{4}}\log(n)\right\}\leq\frac{\varepsilon}{8}\leq\frac{1}{8}.$$

As we also have $\frac{i\varepsilon}{4R}\leq\frac{\varepsilon}{4}\leq\frac{1}{4}$, Fact 7 then gives $\frac{1}{2}\mathbf{L}\preceq\mathbf{L}_{G_i}\preceq\frac{3}{2}\mathbf{L}$. Consequently, G_i has the same connected components as the original graph G, i.e., since we assumed G is connected, so is G_i. Hence, Fact 8 implies that

$$\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top(\mathbf{W}_{i+1}-\mathbf{W}_i)\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\text{op}}\leq 2\cdot\left\|\mathbf{L}_{G_i}^{\frac{\dagger}{2}}\mathbf{B}^\top(\mathbf{W}_{i+1}-\mathbf{W}_i)\mathbf{H}\mathbf{L}_{G_i}^{\frac{\dagger}{2}}\right\|_{\text{op}}$$

$$\leq 2C_{\text{ESO}}\min\left\{\sqrt{\frac{n\log n}{\hat{m}_i}}+\left(\frac{n\log^{\frac{5}{3}}n}{\hat{m}_i}\right)^{\frac{3}{4}},\ \sqrt{\frac{n}{\hat{m}_i}}\log^{\frac{3}{4}}(n)+\left(\frac{n}{\hat{m}_i}\right)^{\frac{3}{4}}\log(n)\right\}+\frac{\varepsilon}{4R}.$$

This proves Item 3 in the inductive hypothesis, as desired, and also implies that after the t^{th} loop,

$$\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top\left(\mathbf{W}_t-\mathbf{W}_0\right)\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\text{op}}\leq\varepsilon. \tag{13}$$

The sparsity bound follows by explicitly removing any $e\in E$ where $[\mathbf{w}_t]_e=0$ from $\vec{H}$. In light of Lemma 25, we note that each of the $\text{poly}(n)$ calls to ExistentialSparsify can be implemented in $\text{poly}(n,\log U,\log\frac{1}{\delta})$ time, and all steps of Algorithm 3 other than ExistentialDecompSparsify run in linear time. We adjust the failure probability by a $\text{poly}(n)$ factor to account for the multiple uses of Corollary 22 via a union bound, giving the claim. □

Theorem 4 is one logarithmic factor in nU away from being optimal, up to low-order terms in ε. The extra logarithmic factor is due to the parameters of our ER decomposition in Proposition 10, and the low-order terms come from the additive terms with polylogarithmic overhead in Proposition 20. In Appendix C, we discuss routes towards removing this overhead, and relate them to known results and open problems in the literature on graph decomposition (i.e., the [AALG18] decomposition scheme) and matrix discrepancy (i.e., the matrix Spencer conjecture).

7 Eulerian sparsification in nearly-linear time

In this section, building upon our approach from Section 6, we provide a nearly-linear time algorithm for sparsifying Eulerian directed graphs. We develop our algorithm via several reductions.

- In Section 7.2, we develop BasicFastSparsify, a basic subroutine which takes as input an initial subgraph with bounded ER diameter (in the sense of Definition 9), and edge weights within a constant multiplicative range. It then returns a reweighting of the initial subgraph which decreases weights by a constant factor on average.
- In Section 7.3, we give a two-phase algorithm which builds upon BasicFastSparsify. In the first phase, the algorithm calls BasicFastSparsify $\approx \log\log n$ times, and we demonstrate that these applications decrease a constant fraction of the edge weights from the original subgraph by a polylog(n) factor. We separate out this small cluster of edges and pass it to the second phase, which applies BasicFastSparsify $\approx \log n$ times to decrease a constant fraction of edge weights by a polynomial factor. We then apply Rounding to fully sparsify these edge weights, incurring small spectral error. Our sparsity-spectral error tradeoff in the second phase loses a polylogarithmic factor over our final desired tradeoff; this is canceled out by the mild edge weight decrease from the first phase, and does not dominate.
- In Section 7.4, we recursively call our ER decomposition algorithm from Section 4, and the two-phase procedure described above. Each round of calls makes constant factor progress on the overall sparsity of our final graph, and hence terminates quickly.

As a preliminary, we provide tools in Section 7.1 to streamline handling of approximation error incurred by state-of-the-art undirected Laplacian solvers, when projecting into circulation space.

7.1 Approximating modified circulations

In this section, we give a self-contained solution to the key computational bottleneck in Section 7.2 when using approximate Laplacian system solvers. We begin by introducing some notation to simplify our presentation. Let $\vec{H}$ be a subgraph of $\vec{G} = (V, E, \mathbf{w})$ with edge set F. We define $H \stackrel{\text{def}}{=} \text{und}(\vec{H})$ and $H^2 \stackrel{\text{def}}{=} (V(H), F, \mathbf{w}_F^2)$, where $\mathbf{w}^2$ is $\mathbf{w}$ with its entries squared. We further define

$$\mathbf{P}_{\vec{H}} \stackrel{\text{def}}{=} \mathbf{I}_F - \mathbf{C}_{\vec{H}}, \text{ where } \mathbf{C}_{\vec{H}} \stackrel{\text{def}}{=} \mathbf{W}_F \mathbf{B}_{\vec{H}} \mathbf{L}_{H^2}^{\dagger} \mathbf{B}_{\vec{H}}^{\top} \mathbf{W}_F, \tag{14}$$

and where $\mathbf{I}_F, \mathbf{W}_F \in \mathbb{R}_{\geq 0}^{E \times E}$ zero out entries of $\mathbf{I}_E, \mathbf{W}$ which do not correspond to edges in F. In Section 7.2, we apply reweightings which are circulations on $\vec{H}$, but which also are orthogonal to a specified vector $\mathbf{v}$. We will eventually set $\mathbf{v}$ to be a current weight vector, to enforce that the total weight of the edges remains unchanged. We hence define the modified projection matrix

$$\mathbf{P}_{\vec{H},\mathbf{v}} \stackrel{\text{def}}{=} \mathbf{P}_{\vec{H}} - \mathbf{u}_{\vec{H},\mathbf{v}} \mathbf{u}_{\vec{H},\mathbf{v}}^{\top}, \text{ where } \mathbf{u}_{\vec{H},\mathbf{v}} \stackrel{\text{def}}{=} \frac{1}{\sqrt{\mathbf{v}^{\top} \mathbf{P}_{\vec{H}} \mathbf{v}}} \mathbf{P}_{\vec{H}} \mathbf{v}. \tag{15}$$

We prove a basic fact about $\mathbf{P}_{\vec{H},\mathbf{v}}$, motivated by the Sherman-Morrison formula.

Lemma 27. *For any $\mathbf{u} \in \mathbb{R}^E$, $\mathbf{P}_{\vec{H},\mathbf{v}}$ defined in (15) satisfies*

$$\mathbf{P}_{\vec{H},\mathbf{v}} \mathbf{v} = \mathbf{0}_E, \ \mathbf{P}_{\vec{H},\mathbf{v}}^2 = \mathbf{P}_{\vec{H},\mathbf{v}}, \text{ and } \mathbf{B}_{\vec{H}}^{\top} \mathbf{W}_{E(\vec{H})} \mathbf{P}_{\vec{H},\mathbf{v}} \mathbf{u} = \mathbf{0}_E.$$

Proof. The first claim follows from directly computing $\mathbf{u}_{\vec{H},\mathbf{v}}\mathbf{u}_{\vec{H},\mathbf{v}}^\top\mathbf{v} = \mathbf{P}_{\vec{H}}\mathbf{v}$. The second follows similarly: since $\mathbf{P}_{\vec{H}}$ is an orthogonal projection matrix, $\mathbf{u}_{\vec{H},\mathbf{v}}$ is a unit vector, and we observe

$$\mathbf{P}_{\vec{H}}\mathbf{u}_{\vec{H},\mathbf{v}}\mathbf{u}_{\vec{H},\mathbf{v}}^\top = \mathbf{u}_{\vec{H},\mathbf{v}}\mathbf{u}_{\vec{H},\mathbf{v}}^\top\mathbf{P}_{\vec{H}} = \mathbf{u}_{\vec{H},\mathbf{v}}\mathbf{u}_{\vec{H},\mathbf{v}}^\top.$$

Finally, the last follows from the fact that $\mathbf{B}_{\vec{H}}^\top\mathbf{W}_{E(\vec{H})}\mathbf{P}_{\vec{H}}$ is the zero operator on $\mathbb{R}^{E\times E}$. □

Thus, $\mathbf{P}_{\vec{H},\mathbf{v}}$ is the projection matrix into the subspace of $\mathbf{P}_{\vec{H}}$'s span that is orthogonal to $\mathbf{v}$.

Algorithm 4 solves the following problem: on input $\xi > 0$, $\mathbf{z} \in \mathbb{R}^E$ with $\mathrm{supp}(\mathbf{z}) \subseteq F$, $\|\mathbf{z}\|_\infty \le 1$, return $\mathbf{x} \in \mathbb{R}^E$ with

$$\mathrm{supp}(\mathbf{x}) \subseteq F,\ \left\|\mathbf{x} - \mathbf{P}_{\vec{H},\mathbf{v}}\mathbf{z}\right\|_\infty \le \xi,\ \left\|\mathbf{B}_{\vec{G}}^\top\mathbf{W}\mathbf{x}\right\|_\infty \le \xi,\ |\langle\mathbf{x},\mathbf{v}\rangle| \le \xi\|\mathbf{v}\|_2. \tag{16}$$

In other words, for an error parameter ξ, we wish to enforce that $\mathbf{w}\circ\mathbf{x}$ is an approximate circulation, and that $\mathbf{x}$ is approximately orthogonal to $\mathbf{v}$ and approximates the true $\mathbf{P}_{\vec{H},\mathbf{v}}\mathbf{z}$ we wish to compute. We remark that $\mathbf{x} = \mathbf{P}_{\vec{H},\mathbf{v}}\mathbf{z}$ satisfies (16) with $\xi = 0$. We will ultimately call Algorithm 4 with inverse-polynomially small ξ, and apply Rounding to incur small error when rounding the residual.

Algorithm 4: ProjMinusRankOne$(\vec{H}, \mathbf{v}, \mathbf{z}, \delta, \xi)$

Input: $\vec{H}$, a subgraph of $\vec{G} = (V, E, \mathbf{w})$ with $\|\mathbf{w}\|_\infty \le u$ and $F \stackrel{\text{def}}{=} E(\vec{H})$, $\mathbf{v}, \mathbf{z} \in \mathbb{R}^E$ with $\mathrm{supp}(\mathbf{v}), \mathrm{supp}(\mathbf{z}) \subseteq F$ and $\|\mathbf{z}\|_\infty \le 1$, $\delta, \xi \in (0,1)$
$n \leftarrow |V|$
$\xi' \leftarrow \frac{\xi}{9nu\sqrt{m}}$
$\mathbf{a} \leftarrow \xi'$-approximate solution to $\mathbf{L}_{H^2}\mathbf{a} = \mathbf{B}_{\vec{H}}^\top\mathbf{W}_F\mathbf{v}$, with probability $\ge 1 - \frac{\delta}{2}$
$\mathbf{b} \leftarrow \xi'$-approximate solution to $\mathbf{L}_{H^2}\mathbf{b} = \mathbf{B}_{\vec{H}}^\top\mathbf{W}_F\mathbf{z}$, with probability $\ge 1 - \frac{\delta}{2}$
$\mathbf{u} \leftarrow \frac{\mathbf{W}_F\mathbf{B}_{\vec{H}}^\top\mathbf{a}}{\|\mathbf{W}_F\mathbf{B}_{\vec{H}}^\top\mathbf{a}\|_2}$, $\mathbf{y} \leftarrow \mathbf{W}_F\mathbf{B}_{\vec{H}}^\top\mathbf{b}$
return $\mathbf{x} \leftarrow \mathbf{z} - \mathbf{y} - \langle\mathbf{y},\mathbf{u}\rangle\mathbf{u}$

Before giving our analysis in Lemma 29, we require one elementary helper calculation.

Lemma 28. *Let $\mathbf{a}, \mathbf{a}_\star \in \mathbb{R}^d$ satisfy $\|\mathbf{a} - \mathbf{a}_\star\|_2 \le \alpha\|\mathbf{a}\|_2$ for $\alpha \in (0, \frac{1}{5})$. Then, for $\mathbf{u} \stackrel{\text{def}}{=} \frac{\mathbf{a}}{\|\mathbf{a}\|_2}$ and $\mathbf{u}_\star \stackrel{\text{def}}{=} \frac{\mathbf{a}_\star}{\|\mathbf{a}_\star\|_2}$, we have $\|\mathbf{u} - \mathbf{u}_\star\|_2 \le 2\alpha$.*

Proof. The problem statement is invariant under scaling $\mathbf{a}$, so without loss of generality assume $\mathbf{u} = \mathbf{a}$, which implies $\|\mathbf{a}_\star\| \in [1-\alpha, 1+\alpha]$. The conclusion follows by triangle inequality:

$$\|\mathbf{u} - \mathbf{u}_\star\|_2 \le \|\mathbf{u} - \mathbf{a}_\star\|_2 + \|\mathbf{a}_\star - \mathbf{u}_\star\|_2 \le \alpha + |\|\mathbf{a}_\star\| - 1| \le 2\alpha.$$

□

Lemma 29. *Under the stated input assumptions,* ProjMinusRankOne *(Algorithm 4) using Proposition 12 in Lines 4-5 returns $\mathbf{x}$ satisfying (16) in time $\breve{O}(|F|\log\frac{nu}{\xi\delta})$ with probability $\ge 1-\delta$.*

Proof. The problem definition and error guarantee (16) are invariant under scaling $\mathbf{v}$, so we assume $\|\mathbf{v}\|_2 = 1$ without loss of generality. Further, the problem is identical if we eliminate all coordinates on $E \setminus F$ (as the input and output are supported in F), so we only handle the case $E = F$. Finally,

for simplicity in this proof, we let $\mathbf{L} \stackrel{\text{def}}{=} \mathbf{L}_{H^2}$, $\mathbf{B} \stackrel{\text{def}}{=} \mathbf{B}_{\vec{H}}$, $\mathbf{W} \stackrel{\text{def}}{=} \mathbf{W}_F$, $\mathbf{I} \stackrel{\text{def}}{=} \mathbf{I}_F$, and $n \stackrel{\text{def}}{=} |V|, m \stackrel{\text{def}}{=} |F|$, and define the ideal vectors (which would be computed in the algorithm if $\xi = 0$):

$$\mathbf{a}_\star \stackrel{\text{def}}{=} \mathbf{L}^\dagger \mathbf{B}^\top \mathbf{W}\mathbf{v},\ \mathbf{b}_\star \stackrel{\text{def}}{=} \mathbf{L}^\dagger \mathbf{B}^\top \mathbf{W}\mathbf{z},$$
$$\mathbf{u}_\star \stackrel{\text{def}}{=} \frac{\mathbf{W}\mathbf{B}\mathbf{a}_\star}{\|\mathbf{W}\mathbf{B}\mathbf{a}_\star\|_2} = \mathbf{u}_{\vec{H},\mathbf{v}},\ \mathbf{y}_\star \stackrel{\text{def}}{=} \mathbf{W}\mathbf{B}\mathbf{b}_\star,\ \mathbf{x}_\star \stackrel{\text{def}}{=} \mathbf{z} - \mathbf{y}_\star - \langle \mathbf{y}_\star, \mathbf{u}_\star \rangle \mathbf{u}_\star = \mathbf{P}_{\vec{H},\mathbf{v}}\mathbf{z}.$$

First, by the definition of approximate solutions (see Proposition 12), we have

$$\|\mathbf{W}\mathbf{B}(\mathbf{a} - \mathbf{a}_\star)\|_2 = \|\mathbf{a} - \mathbf{a}_\star\|_{\mathbf{L}} \le \xi' \|\mathbf{a}_\star\|_{\mathbf{L}} = \xi' \|\mathbf{W}\mathbf{B}\mathbf{a}_\star\|_2 .$$

Hence, by applying Lemma 28, we have $\|\mathbf{u} - \mathbf{u}_\star\|_2 \le 2\xi'$. Similarly,

$$\|\mathbf{y} - \mathbf{y}_\star\|_2 = \|\mathbf{W}\mathbf{B}(\mathbf{b} - \mathbf{b}_\star)\|_2 = \|\mathbf{b} - \mathbf{b}_\star\|_{\mathbf{L}} \le \xi' \|\mathbf{b}_\star\|_{\mathbf{L}} = \xi' \|\mathbf{W}\mathbf{B}\mathbf{b}_\star\|_2 = \xi' \|\mathbf{y}_\star\|_2 \le \xi' \|\mathbf{z}\|_2 ,$$

where the last equality follows by $\mathbf{y}_\star = \mathbf{C}_{\vec{H}}\mathbf{z}$ and the fact that $\mathbf{C}_{\vec{H}}$ is a orthogonal projection. Now,

$$\begin{aligned}\mathbf{x} - \mathbf{x}_\star &= (\mathbf{y}_\star - \mathbf{y}) + (\langle \mathbf{y}_\star, \mathbf{u}_\star \rangle \mathbf{u}_\star - \langle \mathbf{y}, \mathbf{u} \rangle \mathbf{u}) \\ &= (\mathbf{y}_\star - \mathbf{y}) + \langle \mathbf{y}_\star, \mathbf{u}_\star - \mathbf{u} \rangle \mathbf{u}_\star + \langle \mathbf{y}_\star - \mathbf{y}, \mathbf{u} \rangle \mathbf{u}_\star + \langle \mathbf{y}, \mathbf{u} \rangle (\mathbf{u}_\star - \mathbf{u}),\end{aligned}$$

so that by the triangle and Cauchy-Schwarz inequalities, the first conclusion in (16) holds:

$$\begin{aligned}\|\mathbf{x} - \mathbf{x}_\star\|_\infty &\le \|\mathbf{x} - \mathbf{x}_\star\|_2 \\ &\le \|\mathbf{y}_\star - \mathbf{y}\|_2 + \|\mathbf{u}_\star - \mathbf{u}\|_2 \|\mathbf{y}_\star\|_2 \|\mathbf{u}_\star\|_2 + \|\mathbf{y}_\star - \mathbf{y}\|_2 \|\mathbf{u}\|_2 \|\mathbf{u}_\star\|_2 + \|\mathbf{u}_\star - \mathbf{u}\|_2 \|\mathbf{y}\|_2 \|\mathbf{u}\|_2 \\ &\le \|\mathbf{y}_\star - \mathbf{y}\|_2 + 2\xi' \|\mathbf{y}_\star\|_2 + \|\mathbf{y}_\star - \mathbf{y}\|_2 + 2\xi'(\|\mathbf{y}_\star\|_2 + \|\mathbf{y}_\star - \mathbf{y}\|_2) \\ &\le 9\xi' \|\mathbf{z}\|_2 < 9\xi'\sqrt{m} \|\mathbf{z}\|_\infty \le 9\xi'\sqrt{m} < \xi,\end{aligned}$$

given that $\xi' < 1$. Moreover, letting $\|\mathbf{A}\|_{\infty\to\infty} \stackrel{\text{def}}{=} \sup_{\|\mathbf{x}\|_\infty = 1} \|\mathbf{A}\mathbf{x}\|_\infty$ be the largest ℓ_1 norm of a row of $\mathbf{A}$, and noting that $\|\mathbf{B}\|_{\infty\to\infty} \le n$ and $\|\mathbf{W}\|_{\infty\to\infty} \le u$, we have

$$\begin{aligned}\left\|\mathbf{B}^\top \mathbf{W}\mathbf{x}\right\|_\infty &\stackrel{(a)}{=} \left\|\mathbf{B}^\top \mathbf{W}(\mathbf{x} - \mathbf{x}_\star)\right\|_\infty \\ &\le \left\|\mathbf{B}^\top\right\|_{\infty\to\infty} \|\mathbf{W}\|_{\infty\to\infty} \|\mathbf{x} - \mathbf{x}_\star\|_\infty \le nu \|\mathbf{x} - \mathbf{x}_\star\|_\infty \le 9nu\xi'\sqrt{m}, \\ |\langle \mathbf{x}, \mathbf{v} \rangle| &\stackrel{(b)}{=} |\langle \mathbf{x} - \mathbf{x}_\star, \mathbf{v} \rangle| \le \|\mathbf{x} - \mathbf{x}_\star\|_2 \|\mathbf{v}\|_2 \le 9\xi'\sqrt{m} \|\mathbf{v}\|_2 .\end{aligned}$$

Here, both (a) and (b) followed from Lemma 27. By our choice of $\xi' = \frac{\xi}{9nu\sqrt{m}} < 1$, we can guarantee all the desired bounds in (16). Finally, the runtime bound follows directly from Proposition 12. □

7.2 Basic partial sparsification

In this section, we give the basic subroutine of our fast sparsification algorithms, which modifies the edge weights on a well-controlled subgraph (formally, see Definition 33). We first require stating several standard helper matrix concentration results from the literature.

Lemma 30 (Theorem 7.1, [Tro11]). *Let $\delta \in (0,1)$ and let $\{\mathbf{M}_k\}_{k\in[K]} \in \mathbb{R}^{d\times d}$ be a sequence of matrices, and let $\mathbf{s} \in \{\pm 1\}^K$ be a martingale sequence of Rademachers, i.e., $\mathbf{s}_k$ is a Rademacher random variable conditioned on $\{\mathbf{s}_j\}_{j\in[k-1]}$ for all $k \in [K]$. Further, suppose for $\sigma \ge 0$,*

$$\sum_{k\in[K]} \mathbf{M}_k \mathbf{M}_k^\top \preceq \sigma^2 \mathbf{I}_d,\ \sum_{k\in[K]} \mathbf{M}_k^\top \mathbf{M}_k \preceq \sigma^2 \mathbf{I}_d. \tag{17}$$

Then with probability $\geq 1-\delta$,

$$\left\|\sum_{k\in[K]} \mathbf{s}_k \mathbf{M}_k\right\|_{\mathrm{op}} \leq \sigma\sqrt{8\log\left(\frac{2d}{\delta}\right)}.$$

Lemma 31. *Let* $\delta \in (0,1)$*, let* $\mathbf{P} \in \mathbb{R}^{d\times d}$ *be an orthogonal projection matrix, and let* $\mathbf{s} \in \{\pm 1\}^d$ *have independent Rademacher entries. There is a universal constant* C_{sign} *such that*

$$\|\mathbf{P}\mathbf{s}\|_\infty \leq C_{\mathrm{sign}}\sqrt{\log\frac{d}{\delta}} \text{ with probability} \geq 1-\delta.$$

Proof. For any fixed $j \in [d]$, the random variable $X \stackrel{\text{def}}{=} \mathbf{e}_j^\top \mathbf{P}\mathbf{s}$ is sub-Gaussian with parameter $\sigma \stackrel{\text{def}}{=} \|\mathbf{P}_{j:}\|_2 \leq 1$. Standard sub-Gaussian concentration bounds (e.g., [Ver18], Proposition 2.5.2) now imply that with probability $\geq 1-\frac{\delta}{d}$, we have for a universal constant C_{sign}, $X \leq C_{\mathrm{sign}}\sqrt{\log\frac{d}{\delta}}$. Applying a union bound for all $j \in [d]$ concludes the proof. □

We also use the following helper scalar concentration inequality.

Lemma 32. *Let* X *be a* 1*-sub-Gaussian random variable with* $\mathbb{E}X = 0$*, and let* $\mathcal{E}$ *be an event on the outcome of* X *with* $\Pr[\mathcal{E}] \geq 1-\delta$ *where* $\delta \leq \frac{1}{10}$*. Then,* $\left|\mathbb{E}\left[X^2 - \mathbb{E}\left[X^2\right] \mid \mathcal{E}\right]\right| \leq 300\sqrt{\delta}$.

Proof. Let $\mathbb{1}_{\mathcal{E}}$ and $\mathbb{1}_{\mathcal{E}^c}$ denote the 0-1 indicator variables for $\mathcal{E}$ and its complement $\mathcal{E}^c$. Further, we will assume $\Pr[\mathcal{E}^c] = \delta$ as the stated bound is monotone in δ. The random variable $Z \stackrel{\text{def}}{=} X^2 - \mathbb{E}[X^2]$ is 16-sub-exponential (Lemma 1.12, [RH17]), so applying the Cauchy-Schwarz inequality and standard sub-exponential moment bounds (Lemma 1.10, [RH17]) yields

$$\left|\mathbb{E}\left[Z \mid \mathcal{E}\right]\right| = \frac{1}{\Pr[\mathcal{E}]}\left|\mathbb{E}[Z\cdot\mathbb{1}_{\mathcal{E}}]\right| = \frac{1}{1-\delta}\left|\mathbb{E}[Z\cdot\mathbb{1}_{\mathcal{E}^c}]\right| \leq \frac{1}{1-\delta}\mathbb{E}[Z^2]^{\frac{1}{2}}\mathbb{E}\left[\mathbb{1}_{\mathcal{E}^c}\right]^{\frac{1}{2}} \leq 300\sqrt{\delta}.$$

□

Finally, to simplify the statement of the input to our algorithm, we give a useful definition.

Definition 33 (Cluster)**.** *We say* $\vec{H}$ *is a* $(\bar{w},\rho)$*-cluster in* $\vec{G} = (V, E, \mathbf{w})$ *if* $\vec{H}$ *is a subgraph of* $\vec{G}$*,* $\mathbf{w}_e \in [\bar{w}, 2\bar{w}]$ *for all* $e \in E(\vec{H})$*, and letting* $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$*,*

$$\left(\max_{e\in E(\vec{H})} \mathbf{w}_e\right)\cdot\left(\max_{u,v\in V(\vec{H})} \mathrm{ER}_G(u,v)\right) \leq \rho.$$

By definition, any piece in a $(\rho, 2, J)$-ER decomposition of $G = \mathrm{und}(\vec{G})$ (Definition 9) is a $(\bar{w},\rho)$-cluster in $\vec{G}$, for some $\bar{w}$. We now state our main algorithm in this section, BasicFastSparsify.

Intuitively, BasicFastSparsify randomly reweights a current subset of edges in each of τ iterations, after removing any edge whose weight has significantly changed with respect to a reference vector $\mathbf{w}_\star$. In each loop of Lines 6 to 16, the algorithm terminates if either a constant fraction of edge weights in $E(\vec{H})$ have decreased by an ℓ factor compared to $\mathbf{w}_\star$, or a certain potential function bounding the change in weights has decreased significantly. Moreover, each reweighting adds a circulation (and hence preserves degrees), while maintaining that $\|\mathbf{w}_t\|_1$ is unchanged, up to an inverse-polynomial approximation error due to our subroutine ProjMinusRankOne. The algorithm simply iterates this loop until termination. We now analyze Algorithm 5, by bounding the spectral error and showing that each loop of Lines 6 to 16 is likely to terminate.

Algorithm 5: BasicFastSparsify$(\vec{H}, \vec{G}, \mathbf{w}_\star, \ell, \delta, \varepsilon, F, T)$

Input: $\ell \in (0,1)$, $\vec{H}$ a subgraph of $\vec{G} = (V, E, \mathbf{w})$ with $|E(\vec{H})| \geq 40|V(\vec{H})|$, $\mathbf{w}_\star \in \mathbb{R}^E$ with

$$\frac{\|\mathbf{w}\|_1}{\|\mathbf{w}_\star\|_1} \in [0.99, 1.01],\ [\mathbf{w}_\star]_{E \setminus E(\vec{H})} = \mathbf{w}_{E \setminus E(\vec{H})},\ \text{and}\ \frac{\ell}{2}[\mathbf{w}_\star]_e \leq \mathbf{w}_e \leq 60[\mathbf{w}_\star]_e \text{ for all } e \in E(\vec{H}), \tag{18}$$

and $\vec{H}_\star \stackrel{\text{def}}{=} (V(\vec{H}), E(\vec{H}), [\mathbf{w}_\star]_{E(\vec{H})})$ is a $(\bar{w}, \rho)$-cluster in $\vec{G}_\star \stackrel{\text{def}}{=} (V, E, \mathbf{w}_\star)$ and $0.9\mathbf{L}_G \preceq \mathbf{L}_{G_\star} \preceq 1.1\mathbf{L}_G$ for $G_\star \stackrel{\text{def}}{=} \text{und}(\vec{G}_\star)$, $\delta, \varepsilon \in (0, \frac{1}{100})$, $F \subseteq E(\vec{H})$ with $|F| \geq \frac{|E(\vec{H})|}{4}$, T a tree subgraph of $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$ with $\min_{e \in E(T)} \mathbf{w}_e \geq 1$

$m \leftarrow |E(\vec{G})|$, $n \leftarrow |V(\vec{G})|$

$\xi \leftarrow \min(\frac{\ell}{10}, \frac{1}{1000 C_{\text{sign}} \log(\frac{60m\tau}{\delta})}, \frac{\varepsilon}{200mn^2\tau})$, for C_{sign} from Lemma 31

$\eta \leftarrow \frac{1}{20 C_{\text{sign}} \sqrt{\log \frac{60m\tau}{\delta}}}$, $\tau \leftarrow \lceil \frac{720}{\eta^2} \rceil$

$t \leftarrow 0$, $L_t \leftarrow \{e \in F \mid [\mathbf{w}_t]_e \geq 50 \min([\mathbf{w}_\star]_e, \frac{\|\mathbf{w}_F\|_1}{|F|})\}$, $S_t \leftarrow \{e \in F \mid [\mathbf{w}_t]_e \leq \ell[\mathbf{w}_\star]_e\}$

while $|S_t| < \frac{1}{4}|F|$ **and** $\sum_{e \in E(\vec{H})} \log([\mathbf{w}_t]_e) - \log([\mathbf{w}_0]_e) > -|E(\vec{H})|$ **do**

  $\mathbf{w}_0 \leftarrow \mathbf{w}$

  for $0 \leq t \leq \tau$ **do**

    $L_t \leftarrow \{e \in F \mid [\mathbf{w}_t]_e \geq 50 \min([\mathbf{w}_\star]_e, \frac{\|\mathbf{w}_F\|_1}{|F|})\}$, $S_t \leftarrow \{e \in F \mid [\mathbf{w}_t]_e \leq \ell[\mathbf{w}_\star]_e\}$

    if $|S_t| < \frac{1}{4}|F|$ **and** $\sum_{e \in E(\vec{H})} \log([\mathbf{w}_t]_e) - \log([\mathbf{w}_0]_e) > -|E(\vec{H})|$ **then**

      $\vec{H}_t \leftarrow (V(\vec{H}_t), F \setminus (S_t \cup L_t), [\mathbf{w}_t]_{F \setminus (S_t \cup L_t)})$

      $\mathbf{s} \leftarrow$ random vector in $\{-1, 0, 1\}^E$, where $\mathbf{s}_e$ is an independent ± 1 random variable for all $e \in E(\vec{H}_t)$, and $\mathbf{s}_e = 0$ for all $e \in E \setminus E(\vec{H}_t)$

      $\mathbf{x}_t \leftarrow \text{ProjMinusRankOne}(\vec{H}_t, \mathbf{w}_t, \eta\mathbf{s}, \frac{\delta}{4\tau \log_2(\frac{4}{\delta})}, \xi)$

        ▷ That is, $\mathbf{x}_t \approx [\mathbf{x}_\star]_t \stackrel{\text{def}}{=} \eta \mathbf{P}_{\vec{H}_t, \mathbf{w}_t} \mathbf{s}$.

      $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t \circ (\mathbf{1}_E + \mathbf{x}_t)$

    else

      $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$

$\mathbf{d} \leftarrow \mathbf{B}_{\vec{G}}^\top (\mathbf{w} - \mathbf{w}_t)$

$\mathbf{y} \leftarrow$ unique vector in $\mathbf{R}^E$ with $\text{supp}(\mathbf{y}) \subseteq E(T)$ and $\mathbf{B}_{\vec{G}}^\top \mathbf{y} = \mathbf{d}$

$\mathbf{w}_t \leftarrow \mathbf{w}_t + \mathbf{y}$

return $\mathbf{w}' \leftarrow \mathbf{w}_t$

Lemma 34. *There is a universal constant C_{BFS} such that if $C_{\text{BFS}} \cdot \alpha\rho \log(\frac{m}{\delta}) \leq 1$, where*

$$\alpha \stackrel{\text{def}}{=} \frac{\|\mathbf{w}_F\|_1}{|F|\bar{w}},$$

BasicFastSparsify *(Algorithm 5) returns $\mathbf{w}'$ satisfying, with probability $\geq 1 - \delta$:*

1. $\mathbf{B}_{\vec{G}}^\top \mathbf{w}' = \mathbf{B}_{\vec{G}}^\top \mathbf{w}$ *and* $\frac{\|\mathbf{w}'\|_1}{\|\mathbf{w}\|_1} \in [1 - \varepsilon, 1 + \varepsilon]$.
2. $\mathbf{w}'_e \in [\frac{\ell}{2}[\mathbf{w}_\star]_e, 60[\mathbf{w}_\star]_e]$ *for all* $e \in E(\vec{H})$.
3. *Either* $|\{e \in E(\vec{H}) \mid \mathbf{w}'_e \leq \ell[\mathbf{w}_\star]_e\}| \geq \frac{1}{4}|F|$, *or* $\sum_{e \in E(\vec{H})} \log\left(\frac{\mathbf{w}'_e}{\mathbf{w}_e}\right) \leq -|E(\vec{H})|$.

4. $\|\mathbf{L}_G^{\dagger/2}\mathbf{B}_{\vec{G}}^\top(\mathbf{W}'-\mathbf{W})\mathbf{H}_{\vec{G}}\mathbf{L}_G^{\dagger/2}\|_{\mathrm{op}} \le C_{\mathrm{BFS}}\cdot\sqrt{\alpha\rho\log(\frac{m}{\delta})}+\varepsilon$, *where* $G \stackrel{\mathrm{def}}{=} \mathrm{und}(\vec{G})$.

The runtime of BasicFastSparsify *is, for* $Z\sim\mathrm{Geom}(p)$ *where* $p\in[\frac{1}{2},1]$,[5]

$$\breve{O}\left(|E(\vec{H})|\log\left(\frac{n}{\delta\varepsilon\ell}\right)\log\left(\frac{n}{\delta}\right)\cdot Z+|V|\right).$$

Proof. Let $\hat{m}\stackrel{\mathrm{def}}{=}|E(\vec{H})|$. Because the algorithm continues looping Lines 6 to 16 until the condition in Item 3 is met, the conclusion that Item 3 holds is immediate. The remainder of the proof proceeds as follows. We first prove the runtime claim by giving a constant lower bound on the probability a single run of Lines 6 to 16 ever fails to enter the else branch on Line 15, assuming for simplicity that all calls to ProjMinusRankOne are exact, i.e., that every time Line 13 is run,

$$\mathbf{x}_t=[\mathbf{x}_\star]_t=\eta\mathbf{P}_{\vec{H}_t,\mathbf{v}_t}\mathbf{s}. \tag{19}$$

We next prove that Items 1, 2, and 4 hold with the requisite failure probability. Finally, we modify the argument to handle approximation error due to inexactness in Line 13.

Runtime bound. Our goal in this part of the proof is to establish that each run of Lines 6 to 16 results in the else branch on Line 15 being entered with probability $\ge\frac{1}{2}$. We use this claim to obtain our runtime bound. In the following discussion, fix a single run of Lines 6 to 16. We let $\mathcal{E}_t$ denote the event that $\|\mathbf{x}_t\|_\infty\le\frac{1}{10}$ conditioned on the randomness of all iterations $0\le s<t$. We also let $\mathcal{F}_t$ denote the event that the algorithm enters the if branch on Line 10 on iteration t, and

$$p_t\stackrel{\mathrm{def}}{=}\Pr\left[\bigcup_{0\le s<t}\mathcal{F}_s\mid\bigcup_{0\le s<t}\mathcal{E}_s\right],\ \Phi_t\stackrel{\mathrm{def}}{=}\mathbb{E}\left[\sum_{e\in E(\vec{H})}\log\left(\frac{[\mathbf{w}_{t+1}]_e}{[\mathbf{w}_0]_e}\right)\mid\bigcup_{0\le s\le t}\mathcal{E}_s\right], \tag{20}$$

where both definitions in (20) are taken with respect to all randomness used in the current run of Lines 6 to 16. In other words, p_t is the probability the algorithm has not entered the else branch on Line 15 in any iteration $0\le s<t$, and Φ_t is an expected potential function tracking edge weights over iterations $0\le s\le t$, both conditioned on $\bigcup_{0\le s\le t}\mathcal{E}_s$ occurring. Also, note that by Lemma 31, $\Pr[\mathcal{E}_t]\ge 1-\frac{\delta}{4\tau}$, so $\Pr[\bigcup_{0\le t\le\tau}\mathcal{E}_t]\ge 1-\frac{\delta\tau}{4\tau}\ge\frac{3}{4}$. Thus, if we can show $p_\tau\ge\frac{2}{3}$, we have our goal:

$$\Pr\left[\bigcup_{0\le s\le\tau}\mathcal{F}_t\right]=\Pr\left[\bigcup_{0\le s\le\tau}\mathcal{F}_t\mid\bigcup_{0\le s\le\tau}\mathcal{E}_t\right]\Pr\left[\bigcup_{0\le s\le\tau}\mathcal{E}_t\right]\ge\frac{2}{3}\cdot\frac{3}{4}=\frac{1}{2}. \tag{21}$$

Suppose for contradiction that $p_\tau\le\frac{2}{3}$, so that $p_t\le\frac{2}{3}$ for all $0\le t\le\tau$. First, we compute,

[5]The polyloglog factors hidden by the $\breve{O}$ notation will be polyloglog(nU) factors where U is the edge weight ratio of the original graph we sparsify in Section 7.4, as discussed in that section.

following the convention that $[\mathbf{x}_t]_e = 0$ if $e \notin E(\vec{H}_t)$ or we run the else branch in iteration t,

$$\begin{aligned}
\Phi_t - \Phi_{t-1} &= \mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} \log\left(1+[\mathbf{x}_t]_e\right) \mid \bigcup_{0\le s\le t} \mathcal{E}_s\right] \\
&\le \mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e - \frac{1}{3}[\mathbf{x}_t]_e^2 \mid \bigcup_{0\le s\le t} \mathcal{E}_s\right] \\
&= (1-p_t)\,\mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e - \frac{1}{3}[\mathbf{x}_t]_e^2 \mid \mathcal{E}_t \cup \bigcup_{0\le s<t} (\mathcal{E}_s \cup \mathcal{F}_s)\right] \\
&\le \frac{1}{3}\max\left(\mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e - \frac{1}{3}[\mathbf{x}_t]_e^2 \mid \mathcal{E}_t \cup \bigcup_{0\le s<t} (\mathcal{E}_s \cup \mathcal{F}_s)\right], 0\right).
\end{aligned} \tag{22}$$

The second line used the approximation $\log(1+x) \le x - \frac{1}{3}x^2$ for $|x| \le \frac{1}{10}$, the third line used that no weight changes if we enter the else branch, and the last line used our assumption $p_t \le \frac{2}{3}$.

We next upper bound the right-hand side of (22). Observe that the definition of $\mathbf{x}_t$ (assuming (19)) ensures $\sum_{e\in E}[\mathbf{x}_t \circ \mathbf{w}_t]_e = 0$ using Lemma 27, so $\|\mathbf{w}_t\|_1 = \|\mathbf{w}_0\|_1$ in every iteration. Since any $e \in L_t$ due to $[\mathbf{w}_t]_e \ge 50[\mathbf{w}_\star]_e$ must have $[\mathbf{w}_t]_e \ge 50\bar{w}$, and $\|\mathbf{v}_t\|_1 \le 1.01\,\|\mathbf{w}_\star\|_1 \le 2.02\hat{m}\bar{w}$, there can be at most $\frac{\hat{m}}{24}$ such edges. Similarly, at most $\frac{\hat{m}}{50}$ edges $e \in F$ can have $[\mathbf{w}_t]_e \ge \frac{50\|\mathbf{w}\|_1}{|F|}$, so $|L_t| \le \frac{1}{4}|F|$ throughout the algorithm. Hence under $\bigcup_{0\le s<t}(\mathcal{E}_s \cup \mathcal{F}_s)$, which also implies $|S_t| \le \frac{1}{4}|F|$, we always have $|E(\vec{H}_t)| \ge \frac{1}{2}|F|$. Moreover, note that since $\mathbf{x}_t = \eta \mathbf{P}_{\vec{H}_t,\mathbf{v}_t}\mathbf{s}$ for Rademacher $\mathbf{s}$,

$$\mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e\right] = 0,\ \mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e^2\right] = \eta^2\mathbb{E}\left\|\mathbf{P}_{\vec{H}_t,\mathbf{v}_t}\right\|_2^2 = \eta^2 \mathrm{Tr}\left(\mathbf{P}_{\vec{H}_t,\mathbf{v}_t}\right). \tag{23}$$

However, note that the dimension of the subspace spanned by $\mathbf{P}_{\vec{H}_t,\mathbf{v}_t}$ is at least

$$|E(\vec{H}_t)| - (|V(\vec{H})|-1) - 1 \ge \frac{\hat{m}}{8} - \frac{\hat{m}}{40} = \frac{\hat{m}}{10},$$

under the assumption $|E(\vec{H}_t)| \ge \frac{1}{2}|F| \ge \frac{\hat{m}}{8}$, since it has $|V(\vec{H})|-1$ degree constraints and one orthogonality constraint to $\mathbf{w}_t$. We now handle conditioning on the event $\mathcal{E}_t$, which satisfies $1 - \Pr[\mathcal{E}_t] \le \frac{1}{6000^2}$. Combining (23) with the above, and using that each $[\mathbf{x}_t]_e$ is 1-sub-Gaussian (Lemma 31) and the set of $\mathbf{s}$ satisfying $\mathcal{E}_t$ is closed under negation, applying Lemma 32 shows

$$\mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e \mid \mathcal{E}_t\right] = 0,\ \mathbb{E}\left[\sum_{e\in E(\vec{H}_t)} [\mathbf{x}_t]_e^2 \mid \mathcal{E}_t\right] \ge \eta^2\left(\frac{\hat{m}}{10} - \hat{m}\cdot\left(300\cdot\frac{1}{6000}\right)\right) = \frac{\eta^2\hat{m}}{20}. \tag{24}$$

Therefore, combining with (22) shows that Φ_t decreases by at least $\frac{\eta^2\hat{m}}{180}$ for each of the first τ iterations. However, we also have that with probability 1,

$$\sum_{e\in E(\vec{H})} \log\left(\frac{[\mathbf{w}_\tau]_e}{[\mathbf{w}_0]_e}\right) \mid \bigcup_{0\le s\le\tau} \mathcal{E}_s \ge -2\hat{m}.$$

This is because the algorithm freezes the weights $\mathbf{w}_t$ as soon as $\sum_{e\in E(\vec{H})} \log\left(\frac{[\mathbf{w}_t]_e}{[\mathbf{w}_0]_e}\right) \le -\hat{m}$, and the potential can only change by $-\hat{m}$ in an iteration t assuming $\mathcal{E}_t$, since then $\log(1+[\mathbf{x}_t]_e) \ge -1$ entrywise for $e \in F$. This is a contradiction since $\tau \ge \frac{360}{\eta^2}$ (indeed, we choose τ larger by a constant factor to account for inexactness in ProjMinusRankOne later), so $p_\tau \ge \frac{2}{3}$ as claimed. The runtime follows from Lemma 29, as the number of runs of Lines 6 to 16 is $Z \sim \text{Geom}(p)$ for $p \ge \frac{1}{2}$.

Items 1, 2, and 4. We have shown that with probability $\ge 1 - \frac{\delta}{4}$, Lines 6 to 16 terminate after

$$k \stackrel{\text{def}}{=} \log_2\left(\frac{4}{\delta}\right)$$

loops. Conditional on this event and following our earlier notation, the probability of $\bigcup_{0\le t\le\tau} \mathcal{E}_t$ all occurring in each of the at most k loops is at least $1 - \frac{\delta}{4}$ by our choice of η and Lemma 31. Under these events (i.e. that there are at most k loops and all $\|\mathbf{x}_t\|_\infty$ are small), Item 2 is immediate, since edges e with $[\mathbf{w}_t]_e \notin [\ell[\mathbf{w}_\star]_e, 50[\mathbf{w}_\star]_e]$ are removed from consideration in a current iteration t, and no edge weight changes by more than a 1.1 factor multiplicatively. Also, assuming (19), Item 1 is also immediate (we will analyze the inexactness tolerance later).

We now prove Item 4. For all $0 \le t \le \tau$, let $\vec{G}_t \stackrel{\text{def}}{=} (V, E, \mathbf{w}_t)$ and let $G_t \stackrel{\text{def}}{=} \text{und}(\vec{G}_t)$. We assumed that $\vec{H}_\star$ was a $(\bar{w}, \rho)$-cluster in $\vec{G}_\star$, and no entry of $\mathbf{w}_t$ restricted to $E(\vec{H}_t) = F \setminus (S_t \cup L_t)$ is larger than $50\alpha\bar{w}$ by definition of L_t, so

$$\left(\max_{e\in E(\vec{H}_t)} [\mathbf{w}_t]_e\right) \cdot \left(\max_{u,v\in V(\vec{H})} \text{ER}_G(u,v)\right) \le 75\alpha\rho \text{ for all } 0 \le t \le \tau.$$

Here we used that $\text{ER}_G(u,v) \le 1.5\text{ER}_{G_\star}(u,v)$ for all u, v by assumption. By applying Lemma 19 for all iterations $0 \le t' \le t$ to the sequence of matrices $\widetilde{\mathbf{A}}_e$ in (11) for $e \in E(\vec{H}_t)$, we inductively apply Lemma 30 to show that with probability $1 - \frac{\delta t}{4\tau k}$, on any of the k runs of Lines 6 to 16,

$$\left\|\mathbf{L}_{\vec{G}}^{\frac{\dagger}{2}} \mathbf{B}_{\vec{G}}^\top (\mathbf{W}_t - \mathbf{W}) \mathbf{H}_{\vec{G}} \mathbf{L}_{\vec{G}}^{\frac{\dagger}{2}}\right\|_{\text{op}} \le \frac{1}{20C_{\text{sign}}\sqrt{\log\frac{60m\tau}{\delta}}} \cdot 4\sqrt{75\alpha\rho t \log\left(\frac{8m\tau k}{\delta t}\right)} \le \frac{4}{C_{\text{sign}}} \cdot \sqrt{\alpha\rho t}.$$

There are a few subtleties in the above calculation. First, observe that Lemma 19 implies that if the $\widetilde{\mathbf{A}}_e$ are defined with respect to $\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}$ rather than $\mathbf{P}_{\vec{H}_t}$ (as in Algorithm 5), the variance bound still holds, because Lemma 17 applies to $\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}$ as well. Second, inductively using the guarantee above with Fact 7 shows that $0.9\mathbf{L}_G \preceq \mathbf{L}_{G_t} \preceq 1.1\mathbf{L}_G$ for all iterations t, where we used the assumption on $\alpha\rho$ for a large enough choice of C_{BFS}, so we adjusted the right-hand side by a constant factor. Third, note that the above argument holds with probability $\ge 1 - \frac{\delta}{4k}$ for each of the $\le k$ runs of Lines 6 to 16, so it holds with probability $\ge 1 - \frac{\delta}{4}$ for all of them by a union bound.

Finally, we need to condition on all $\mathcal{E}_t$ holding in all loops. We give a simple argument which removes this conditioning. If any $\mathcal{E}_t$ fails, we set all future weight updates to zero. Therefore, regardless of whether the $\mathcal{E}_t$ occur, the matrix variance (17) in our application of Lemma 30 is bounded as we claimed. In particular, in an iteration t, as long as no $\mathcal{E}_s$ has occured for $0 \le s < t$, Lemma 19 holds, and if any have occured, the variance is trivially bounded by 0.

The overall failure probability of $\le \delta$ comes from union bounding on the three events we have conditioned on so far (finishing in k loops, all $\mathcal{E}_t$ holding in all loops, Item 4 holding), and the event that all of the $\le k\tau$ executions of Line 13 succeeed, which occurs with probability $\ge 1 - \frac{\delta}{4}$.

Inexactness of projection. It remains to discuss the effect of replacing our exact projections with our approximation through ProjMinusRankOne. Because we ensured $\xi \leq \frac{\ell}{10}$, the first bound in (16) shows that entrywise $\mathbf{x}_t$ is not affected by more than $\frac{\ell}{10}$ by approximation, so accounting for slack in our earlier argument Item 2 remains true. Next, using

$$-\frac{1}{3}[\mathbf{x}_t]_e^2 \leq -\frac{1}{3.3}[[\mathbf{x}_\star]_t]_e^2 + 4[\mathbf{x}_t - [\mathbf{x}_\star]_t]_e^2 \leq -\frac{1}{3.3}[\mathbf{x}_\star]_t]_e^2 + 4\xi^2,$$

we have by $\xi \leq \frac{1}{1000 C_{\text{sign}} \log(\frac{60m\tau}{\delta})}$ that the approximation negligibly affects the argument in (24), which we accommodated in the constant factors in τ, so it is still the case that Lines 6 to 16 terminate with probability $\geq \frac{1}{2}$ in each loop. Regarding Item 1, note that

$$\mathbf{B}_{\vec{G}}^\top \mathbf{w}_t + \mathbf{B}_{\vec{G}}^\top \mathbf{y} = \mathbf{B}_{\vec{G}}^\top \mathbf{w}$$

in each iteration after applying the degree fixing in Line 19, so the invariant on degrees holds as claimed. The bound $\|\mathbf{w}_t\|_2 \leq \sqrt{m}\,\|\mathbf{w}_t\|_\infty \leq 120\sqrt{m}\bar{w}$, combined with the last claim in (16) and $\xi \leq \frac{\varepsilon}{200\sqrt{m}\tau}$, shows the ℓ_1 norm of the weights cannot grow by more than $\varepsilon\bar{w}$ throughout. Moreover, the assumption $\xi \leq \frac{\varepsilon}{mn^3\tau}$ with the second guarantee in (16) shows that in each iteration, the total degree imbalance $\|\mathbf{d}\|_1 \leq \frac{\varepsilon}{3mn^2\tau}$, and the error vector $\mathbf{z}$ (in the context of Lemma 24) satisfies $\|\mathbf{z}\|_1 \leq m\xi \leq \frac{\varepsilon}{3n^2\tau}$. Lemma 24 then shows that $\|\mathbf{y}\|_1 \leq m\,\|\mathbf{y}\|_\infty \leq m\,\|\mathbf{d}\|_1 \leq \frac{\varepsilon}{3n\tau}$. The last two guarantees in Lemma 24 combined with the triangle inequality show that in each iteration, the additional spectral error due to approximate solves is $\frac{2\varepsilon}{3\tau}$, and the additional error due to rounding is $\frac{\varepsilon}{3\tau}$ giving the additional spectral error term in Item 4 after accumulating over all iterations. Finally, the runtime follows directly from Lemma 24 (for computing $\mathbf{y}$), and Lemma 29. □

We provide one additional result which helps in disjoint applications of BasicFastSparsify.

Corollary 35. *Consider calling* BasicFastSparsify *I times, with shared parameters $\vec{G}, \mathbf{w}_\star, \ell, \delta, \varepsilon$, but on edge-disjoint subgraphs $\{\vec{H}_i\}_{i\in[I]}$ through $\vec{G}$, so that the corresponding $[\vec{H}_\star]_i$ are all $(\bar{w}_i, \rho)$-clusters in $\vec{G}_\star$ for some value of $\bar{w}_i$. Then with probability $\geq 1 - \delta I$, the total operator norm error (i.e., Item 4) incurred by all calls is bounded by*

$$C_{\text{BFS}} \cdot \sqrt{\rho \log\left(\frac{m}{\delta}\right)} + \varepsilon I.$$

Proof. The claim is that we do not incur an I factor overhead in the operator norm error on the first term in the spectral error, and also do not incur an I factor overhead on the $|V|$ term in the runtime. Note that the bound came from combining the variance bound in Lemma 19 with the high-probability guarantee in Lemma 30. By treating each of the at most τ reweightings applied by Algorithm 5 in parallel across the edge-disjoint clusters, the combined variance in the sense of Lemma 19, where $\vec{H}$ is set to the union of all clusters, is still bounded. The failure probability is by a union bound over I calls. For the runtime, note that we can compute the degree imbalances in Line 17 for all clusters simultaneously, and route them on T in time $O(|V|)$ per iteration. □

7.3 Sparsifying an ER decomposition

In this section, we state and analyze DecompSparsify, which is a two-phase application (with different parameters) of BasicFastSparsify to components of an ER decomposition.

We use the following scalar concentration inequality to bound the runtime with high probability.

Algorithm 6: DecompSparsify($\{\vec{G}_i\}_{i\in[I]}, \vec{G}, T, \delta, \varepsilon, W$)

Input: $\{\vec{G}^{(i)}\}_{i\in[I]}$, subgraphs of simple $\vec{G} = (V, E, \mathbf{w})$ with $\max_{e\in\mathrm{supp}(\mathbf{w})} \mathbf{w}_e \le W$, and such that $\{G^{(i)} \stackrel{\text{def}}{=} \mathrm{und}(\vec{G}^{(i)})\}_{i\in[I]}$ are a $(\rho, 2, J)$-ER decomposition of $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$, T a tree subgraph of G with $\min_{e\in E(T)} \mathbf{w}_e \ge 1$, $\delta, \varepsilon \in (0, \frac{1}{100})$

$m \leftarrow E(\vec{G})$, $n \leftarrow V(\vec{G})$, $R \leftarrow \emptyset$

for $i \in [I]$ **do**

  $\vec{H} \leftarrow \vec{G}^{(i)}$, $\hat{m} \leftarrow |E(\vec{H})|$, $\hat{n} \leftarrow |V(\vec{H})|$, $\mathbf{w}_\star \leftarrow \mathbf{w}$

  if $\hat{m} \ge 40\hat{n}$ **then**

    $\mathbf{w}_0 \leftarrow \mathbf{w}$, $\vec{G}_0 \leftarrow \vec{G}$, $\vec{H}_0 \leftarrow \vec{H}$, $\ell_1 \leftarrow \frac{1}{2\log^2(\frac{nW}{\varepsilon})}$, $\tau_1 \leftarrow \log(\frac{2}{\ell_1})$

    for $0 \le t < \tau_1$ **do**

      $\mathbf{w}_{t+1} \leftarrow$ BasicFastSparsify$(\vec{H}_t, \vec{G}_t, \mathbf{w}_\star, \ell_1, \frac{\delta}{4I\tau_1}, \frac{\varepsilon}{4I\tau_1}, E(\vec{H}), T)$

      $\vec{G}_{t+1} \leftarrow (V, E, \mathbf{w}_{t+1})$, $\vec{H}_{t+1} \leftarrow (V(\vec{H}), E(\vec{H}), [\mathbf{w}_{t+1}]_{E(\vec{H})})$

    $F \leftarrow \{e \in E(\vec{H}) \mid [\mathbf{w}_t]_e \le \ell_1[\mathbf{w}_\star]_e\}$

    $\mathbf{w}_0 \leftarrow \mathbf{w}_t$, $\vec{G}_0 \leftarrow \vec{G}_t$, $\vec{H}_0 \leftarrow \vec{H}$, $\ell_2 \leftarrow \frac{\varepsilon}{4nmW}$, $\tau_2 \leftarrow \log(\frac{2}{\ell_2})$

    for $0 \le t < \tau_2$ **do**

      $\mathbf{w}_{t+1} \leftarrow$ BasicFastSparsify$(\vec{H}_t, \vec{G}_t, \mathbf{w}_\star, \ell_2, \frac{\delta}{4I\tau_2}, \frac{\varepsilon}{4I\tau_2}, F, T)$

      $\vec{G}_{t+1} \leftarrow (V, E, \mathbf{w}_{t+1})$, $\vec{H}_{t+1} \leftarrow (V(\vec{H}), E(\vec{H}), [\mathbf{w}_{t+1}]_{E(\vec{H})})$

    $R \leftarrow R \cup \{e \in E(\vec{H}) \mid [\mathbf{w}_t]_e \le \frac{\varepsilon}{4nm}\}$, $\mathbf{w} \leftarrow \mathbf{w}_t$

return $\vec{G}' \leftarrow (V, E, \mathbf{w}_{E\setminus R} +$ Rounding$(\vec{G}, \mathbf{w}_R, T))$

Lemma 36. *Let $\delta \in (0,1)$, and let $\{Z_i\}_{i\in[I]} \subset \mathbb{N}$ be distributed as $Z_i \mid \{Z_j\}_{j<i} \sim \mathrm{Geom}(p_i)$ where $p_i \in [\frac{1}{2}, 1]$ for all $i \in [I]$. Then for $S \stackrel{\text{def}}{=} \sum_{i\in[I]} Z_i$,*

$$\Pr\left[S > 5\left(I + \log\left(\frac{1}{\delta}\right)\right)\right] \le \delta.$$

Proof. It suffices to handle the case where $p_i = \frac{1}{2}$ for all $i \in [I]$, since otherwise we can couple Z_i to an instance of $\mathrm{Geom}(\frac{1}{2})$ which never exceeds Z_i. Then we compute the moment generating function of S: for $\lambda < \log(2)$, $\mathbb{E}\exp(\lambda S) = (\frac{\exp(\lambda)}{2-\exp(\lambda)})^I$, so by Markov's inequality, for $t \stackrel{\text{def}}{=} 5(I + \log_2(\frac{1}{\delta}))$,

$$\Pr[S > t] < \exp(-\lambda t)\left(\frac{\exp(\lambda)}{2-\exp(\lambda)}\right)^I = \left(\frac{2}{3}\right)^t 3^I < \delta,$$

where we use the choice $\lambda = \log(\frac{3}{2})$ and substituted our choice of t. □

We now state our guarantee on Algorithm 6 and provide its analysis.

Lemma 37. *There is a universal constant C_{PS} such that if $C_{\mathrm{PS}} \cdot \rho\log(\frac{nW}{\delta\varepsilon})\log^2\log(\frac{nW}{\varepsilon}) \le 1$,* DecompSparsify *(Algorithm 6) returns $\vec{G}' = (V, E, \mathbf{w}')$ satisfying, with probability $\ge 1 - \delta$,*

$$\begin{aligned} &\mathbf{B}_{\vec{G}}^\top \mathbf{w}' = \mathbf{B}_{\vec{G}}^\top \mathbf{w}, \quad \mathrm{nnz}(\mathbf{w}') \le \frac{31}{32}\mathrm{nnz}(\mathbf{w}) + C_{\mathrm{PS}} \cdot nJ, \\ &\text{and } \left\|\mathbf{L}_G^{\frac{\dagger}{2}} \mathbf{B}_{\vec{G}}^\top \left(\mathbf{W}' - \mathbf{W}\right) \mathbf{H}_{\vec{G}} \mathbf{L}_G^{\frac{\dagger}{2}}\right\|_{\mathrm{op}} \le C_{\mathrm{PS}} \sqrt{\rho \log\left(\frac{nW}{\delta\varepsilon}\right)} \log\log\left(\frac{nW}{\varepsilon}\right) + \varepsilon. \end{aligned} \tag{25}$$

Moreover, $\max_{e\in E}\frac{\mathbf{w}'_e}{\mathbf{w}_e}\le C_{\text{PS}}$. *The runtime of* DecompSparsify *is*

$$\breve{O}\left(|E|\log^2\left(\frac{nW}{\delta\varepsilon}\right)\log\left(\frac{nW}{\varepsilon}\right)\right).$$

Proof. Throughout the proof, condition on all calls to BasicFastSparsify succeeding assuming their input conditions are met (i.e., the guarantees in Lemma 34 hold, with total spectral error controlled by Corollary 35), which gives a failure probability of $\frac{\delta}{2}$. We claim that every $\vec{G}_t$ used in calls to BasicFastSparsify satisfies $0.9\mathbf{L}_{G_t}\preceq\mathbf{L}_G\preceq 1.1\mathbf{L}_{G_t}$, where $G\stackrel{\text{def}}{=}\text{und}(\vec{G})$ for $\vec{G}$ the original input to the algorithm, and $G_t\stackrel{\text{def}}{=}\text{und}(\vec{G}_t)$. We defer the proof of this claim to the end.

Next, fix $i\in[I]$ and consider the τ_1 loops of Lines 7 to 9. In all calls to BasicFastSparsify, the conditions on $\mathbf{w}_\star$ are met by assumption (i.e., each $\vec{G}^{(i)}$ is an ER decomposition piece with parameters $(1.2\rho,2)$ in $\vec{G}_t$, since we claimed $0.9\mathbf{L}_{G_t}\preceq\mathbf{L}_G\preceq 1.1\mathbf{L}_{G_t}$). Moreover, BasicFastSparsify is only called if $\hat{m}\ge 40\hat{n}$, and the conditions in (18) are preserved inductively by Lemma 34, since the ℓ_1 norm of the weights does not change by more than a $\frac{\varepsilon}{4\tau_1}$ factor in each iteration. This shows that the τ_1 loops of Lines 7 to 9 all have their input conditions met, so we may assume they succeed. We claim that in this case, F on Line 10 must have $|F|\ge\frac{\hat{m}}{4}$. To see this, suppose $|F|<\frac{\hat{m}}{4}$, which means the second part of Item 3 in Lemma 34 holds for all iterations $0\le t<\tau_1$. However, since Lemma 34 also guarantees

$$\sum_{e\in E(\vec{H})}\log\left(\frac{[\mathbf{w}_\tau]_e}{\mathbf{w}_e}\right)>-\hat{m}\log\left(\frac{2}{\ell}\right)=-\hat{m}\tau_1,$$

we arrive at a contradiction after τ_1 iterations, so the first part of Item 3 must have held at some point. With this size bound (showing F is a valid input), an analogous argument shows that after the τ_2 loops in Lines 12 to 14 have finished, at least $\frac{\hat{m}}{16}$ edges are added to R. Observe that each component $\vec{G}^{(i)}$ with $\hat{m}_i$ edges and $\hat{n}_i$ vertices either has $\frac{1}{16}$ of its edges added to R or $\hat{m}_i\le 40\hat{n}_i$, and further $\sum_{i\in[I]}\hat{n}_i\le nJ$. Since all edges from R are zeroed out in the final weighting $\mathbf{w}'$, and at most half the edges do not belong to any $\vec{G}^{(i)}$, this gives the bound on $\text{nnz}(\mathbf{w}')$. Similarly, if all calls to BasicFastSparsify succeed, since applying Rounding at the end of the algorithm preserves degrees, recursively applying Item 1 in Lemma 34 shows that $\mathbf{B}_{\vec{G}}^\top\mathbf{w}'=\mathbf{B}_{\vec{G}}^\top\mathbf{w}$.

It remains to show the spectral error bound. Observe that we have $\alpha=2$ in the first τ_1 calls to BasicFastSparsify for each cluster (in Lines 9 to 9), and $\alpha=\frac{1}{\log^2(\frac{nW}{\varepsilon})}$ in the last τ_2 calls (in Lines 12 to 14). Therefore, taking note of Corollary 35 and since $I\le m$, the spectral error in all intermediate iterations across all decomposition pieces is bounded by

$$O\left(\sqrt{\rho\log\left(\frac{m\tau_1}{\delta}\right)}\cdot\tau_1+\sqrt{\frac{\rho}{\log^2\left(\frac{nW}{\varepsilon}\right)}\log\left(\frac{m\tau_2}{\delta}\right)}\cdot\tau_2\right)=O\left(\sqrt{\rho\log\left(\frac{mW}{\delta\varepsilon}\right)}\log\log\left(\frac{nW}{\varepsilon}\right)\right).$$

Additionally, there is an $\frac{\varepsilon}{4\tau_1 I}\cdot\tau_1 I+\frac{\varepsilon}{4\tau_2 I}\cdot\tau_2 I$ additive error term which comes from Corollary 35, which is bounded by $\frac{2\varepsilon}{3}$ after accounting for the change in the graph Laplacian (i.e., by Fact 8). For appropriate C_{PS}, this both proves the desired spectral error bound by the triangle inequality, as well as the claimed $0.9\mathbf{L}_{G_t}\preceq\mathbf{L}_G\preceq 1.1\mathbf{L}_{G_t}$ throughout the algorithm by Fact 7, which again implies that G_t is connected under our assumption that G is connected (see discussion in Section 2). Finally, applying Rounding incurs at most $\frac{\varepsilon}{3}$ spectral error through the final graph by Lemma 24, which is at most ε spectral error through the original graph by Fact 8. The guarantee on the weight increase is clear as we only modify weights within clusters, and Item 2 of Lemma 34 shows no edge weight grows by more than a factor of 60. This concludes the correctness proof.

For the runtime, the total number of times we call BasicFastSparsify on each piece of the ER decomposition is $\tau_1 + \tau_2 = O(\log \frac{nW}{\varepsilon})$. Thus, Lemma 36 shows that with probability $\leq \frac{\delta}{2}$, the number of times Lines 6 to 16 runs is $O(\log \frac{nW}{\delta\varepsilon})$, for all decomposition pieces simultaneously. This gives the first term in the runtime via Lemma 34, as all decomposition pieces have disjoint edges. For the second term in the runtime, it suffices to note that Lines 17 to 19 can be applied in parallel (after summing the degree imbalances $\mathbf{d}$ in Line 17) for all decomposition pieces which terminate in a given run of Lines 6 to 16, so we do not pay a multiplicative overhead of $|I|$ on the runtime of Lemma 24. The total failure probability is via a union bound over Lemmas 34 and 36. □

7.4 Complete sparsification algorithm

We now provide our complete near-linear time Eulerian sparsification algorithm. Our algorithm iteratively applies the ER decomposition from Proposition 10, sparsifies the decomposition using Algorithm 6, and calls Algorithm 1 on small-weight edges to maintain a bounded weight ratio. The following theorem gives a refined version of Theorem 2.

Algorithm 7: FastSparsify$(\vec{G}, \varepsilon, \delta)$

1 **Input:** Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $\mathbf{w}_e \in [1, U]$ for all $e \in E$, $\varepsilon, \delta \in (0, 1)$
2 $n \leftarrow |V|$, $m \leftarrow |E|$
3 $T \leftarrow$ arbitrary spanning tree of $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$, $\hat{E} \leftarrow E \setminus E(T)$
4 $R \leftarrow 6 \log n$, $U_{\max} \leftarrow U \cdot C_{\text{PS}}^R$ for C_{PS} in Lemma 37
5 $t \leftarrow 0$, $\mathbf{w}_0 \leftarrow \mathbf{w}$
6 **while** $t < R$ **and** $\text{nnz}([\mathbf{w}_t]_{\hat{E}}) > n \log(n) \log(\frac{32R^2 mnU_{\max}}{\delta\varepsilon}) \log^2 \log(\frac{32RmnU_{\max}}{\varepsilon}) \cdot \frac{2^{22} C_{\text{PS}}^2}{\varepsilon^2}$ **do**
7 $\quad \vec{G}_t \leftarrow (V, E, \mathbf{w}_t)$, $G_t \leftarrow \text{und}(\vec{G}_t)$
8 $\quad S \leftarrow \text{ERDecomp}([G_t]_{\hat{E}}, 2, \frac{\delta}{2R})$ ▷ See Proposition 10.
9 $\quad \vec{G}'_t \stackrel{\text{def}}{=} (V, E, \mathbf{w}'_t) \leftarrow \text{DecompSparsify}(S, \vec{G}_t, T, \frac{\delta}{2R}, \frac{\varepsilon}{4R}, U_{\max})$
10 $\quad D \leftarrow \{e \in \hat{E} \mid [\mathbf{w}'_t]_e \leq \frac{\varepsilon}{4mn}\}$
11 $\quad \mathbf{w}_{t+1} \leftarrow [\mathbf{w}'_t]_{E \setminus D} + \text{Rounding}(\vec{G}'_t, [\mathbf{w}'_t]_D, T)$
12 $\quad t \leftarrow t + 1$
13 **return** $\vec{H} \leftarrow (V, E, \mathbf{w}_t)$

Theorem 38. *Given Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, U]^E$ and $\varepsilon, \delta \in (0, 1)$, FastSparsify (Algorithm 7) returns Eulerian $\vec{H}$ such that with probability $\geq 1 - \delta$, $\vec{H}$ is an ε-approximate Eulerian sparsifier of $\vec{G}$, and*

$$|E(\vec{H})| = O\left(\frac{n}{\varepsilon^2} \log(n) \log\left(\frac{nU}{\delta}\right) \log^2 \log(nU)\right), \ \log\left(\frac{\max_{e \in \text{supp}(\mathbf{w}')} \mathbf{w}'_e}{\min_{e \in \text{supp}(\mathbf{w}')} \mathbf{w}'_e}\right) = O(\log(nU)).$$

The runtime of FastSparsify *is $\breve{O}\left(m \log^2\left(\frac{nU}{\delta}\right) \log(nU)\right)$.*

Proof. Throughout, condition on the event that all of the at most R calls to ERDecomp and DecompSparsify succeed, which happens with probability $\geq 1 - \delta$. Because DecompSparsify guarantees that no weight grows by more than a C_{PS} factor in each call, $U_{\max}$ is a valid upper bound for the maximum weight of any edge throughout the algorithm's execution. Moreover, we explicitly delete any edge whose weight falls below $\frac{\varepsilon}{4mn}$ throughout the algorithm in Line 10, and these edges

never appear in a call to ERDECOMP again. Hence, $J_{\max} \stackrel{\text{def}}{=} \log_2(\frac{32mnU_{\max}}{\varepsilon})$ is a valid upper bound on the number of decomposition pieces ever returned by ERDECOMP, by Proposition 10.

Next, note that under the given lower bound on $[\mathbf{w}_t]_{\hat{E}}$ in a given iteration (which is larger than $2C_{\text{PS}} \cdot nJ_{\max}$), the sparsity progress guarantee in (25) shows that the number of edges in each iteration is decreasing by at least a $\frac{1}{64}$ factor until termination. Since $m \leq n^2$ and the algorithm terminates before reaching n edges, R is a valid upper bound on the number of iterations before the second condition in Line 6 fails to hold, which gives the sparsity claim. Moreover, because the first term in the spectral error bound in (25) decreases by a geometric factor of $1 - \frac{1}{256}$ in each round (as ρ scales inversely in the current support size of $\mathbf{w}_t$), the sum of all such terms contributes at most 256 times the final contribution before termination. By plugging in the bound $\rho \leq \frac{33n\log(n)}{m}$ from Proposition 10 with the lower bound on m throughout the algorithm, the total contribution of these terms is at most $\frac{\varepsilon}{4}$. Similarly, the second additive term in (25) contributes at most $\frac{\varepsilon}{4}$ throughout the R rounds, and the rounding on Line 11 also contributes at most $\frac{\varepsilon}{4}$ by Lemma 24. Here we remark that once an edge is rounded on Line 11, it is removed from the support of $\mathbf{w}_t$ for the rest of the algorithm. Adjusting these error terms by a $\frac{4}{3}$ factor (i.e., because of Fact 7 which shows $\mathbf{L}_{G_t}$ for $G_t \stackrel{\text{def}}{=} \text{und}(\vec{G}_t)$ is stable throughout the algorithm, and Fact 8 which shows how this affects the error terms), we have the claimed spectral error guarantee. The sparsity bound follows again by explicitly removing any $e \in E$ where $[\mathbf{w}_t]_e = 0$ from $\vec{H}$.

Finally, the runtime follows from combining Proposition 10 (which does not dominate), and Lemma 37. Here we note that we do not incur an extra logarithmic factor over Lemma 37 because the edge count is a geometrically decreasing sequence (with constant ratio). □

8 Applications

A direct consequence of our improved nearly-linear time Eulerian sparsifier in Theorem 38 is a significant improvement in the runtime of solving Eulerian Laplacian linear systems due to Peng and Song [PS22]. In turn, combined with reductions in [CKPPSV16], our improved Eulerian system solver implies faster algorithms for a host of problems in directed graphs. We summarize these applications in this section. As a starting point, we state the reduction of [PS22] from solving Eulerian Laplacian linear systems to sparsifying Eulerian graphs.

Proposition 39 (Theorem 1.1, [PS22]). *Suppose there is an algorithm which takes in Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $n = |V|$, $m = |E|$, $\mathbf{w} \in [1, U]^E$, and returns an ε'-approximate Eulerian sparsifier with $\mathcal{S}(n, U, \varepsilon')$ edges with probability $\geq 1 - \delta$, in time $\mathcal{T}(m, n, U, \varepsilon', \delta)$. Then given Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $n = |V|$, $m = |E|$, $\mathbf{w} \in [1, U]^E$, $\mathbf{b} \in \mathbb{R}^V$, and error parameter $\varepsilon \in (0, 1)$, there is an algorithm running in time*

$$O\left(m \log\left(\frac{nU}{\varepsilon}\right) + \mathcal{T}\left(m, n, U, 1, \frac{\delta}{\log nU}\right)\right)$$
$$+\breve{O}\left(\mathcal{T}\left(\mathcal{S}\left(n, U, 1\right), n, U, 1, \frac{\delta}{\log nU}\right)\log(nU) + \mathcal{S}(n, U, 1)\log(nU)\log\left(\frac{nU}{\varepsilon}\right)\right)$$

which returns $\mathbf{x} \in \mathbb{R}^V$ satisfying, with probability $\geq 1 - \delta$,

$$\left\|\mathbf{x} - \vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\right\|_{\mathbf{L}_G} \leq \varepsilon \left\|\vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\right\|_{\mathbf{L}_G}, \text{ where } G \stackrel{\text{def}}{=} \text{und}(\vec{G}). \tag{26}$$

Plugging Theorem 38 into Proposition 39, we obtain our faster solver for Eulerian Laplacians. The following corollary is a refined version of Corollary 3.

Corollary 40 (Eulerian Laplacian solver)**.** *Given Eulerian* $\vec{G} = (V, E, \mathbf{w})$ *with* $|V| = n, |E| = m, \mathbf{w} \in [1, U]^E$*,* $\mathbf{b} \in \mathbb{R}^V$*, and error parameter* $\varepsilon \in (0, 1)$*, there is an algorithm running in time*

$$\breve{O}\left(m \log^2\left(\frac{nU}{\delta}\right) \log\left(\frac{nU}{\varepsilon}\right) + n \log^2(nU) \log^3\left(\frac{nU}{\delta}\right) \log\left(\frac{nU}{\varepsilon}\right)\right)$$

which returns $\mathbf{x} \in \mathbb{R}^V$ *satisfying, with probability* $\geq 1 - \delta$*,*

$$\left\|\mathbf{x} - \vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\right\|_{\mathbf{L}_G} \leq \varepsilon \left\|\vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\right\|_{\mathbf{L}_G}, \text{ where } G \stackrel{\text{def}}{=} \text{und}(\vec{G}).$$

We remark that there is a more precise runtime improving upon Corollary 40 in the logarithmic terms when δ, ε are sufficiently small or U is sufficiently large, but we state the simpler variant for the following applications and for readability purposes. Plugging our primitive in Corollary 40 into black-box reductions from [CKPPSV16] then gives algorithms to solve linear systems in row-or-column diagonally dominant matrices, which we now define.

Definition 41. *We say* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *is row-column diagonally dominant (RCDD) if* $\mathbf{M}_{ii} \geq \sum_{j \neq i} |\mathbf{M}_{ij}|$ *and* $\mathbf{M}_{ii} \geq \sum_{j \neq i} |\mathbf{M}_{ji}|$ *for all* $i \in [n]$*. We say* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *is row-or-column diagonally dominant (ROCDD) if either* $\mathbf{M}_{ii} \geq \sum_{j \neq i} |\mathbf{M}_{ij}|$ *for all* $i \in [n]$*, or* $\mathbf{M}_{ii} \geq \sum_{j \neq i} |\mathbf{M}_{ji}|$ *for all* $i \in [n]$*.*

Most notably, Eulerian Laplacians are RCDD, and all directed Laplacians are ROCDD. In [CKPPSV16] (see also [AJSS19] for an alternative exposition), the following reduction was provided.

Proposition 42 (Theorem 42, [CKPPSV16])**.** *Let* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *be ROCDD, and suppose both* $\mathbf{M}$ *and its diagonal have multiplicative range at most* κ *on their nonzero singular values. There is an algorithm which, given* $\mathbf{M}$*,* $\mathbf{b} \in \text{Im}(\mathbf{M})$*, and error parameter* $\varepsilon \in (0, 1)$*, solves* $\log^2(\frac{n\kappa}{\varepsilon})$ *Eulerian linear systems to relative accuracy* $\text{poly}(\frac{\varepsilon}{n\kappa})$ *(in the sense of (26)) and returns* $\mathbf{x} \in \mathbb{R}^n$ *satisfying*

$$\|\mathbf{M}\mathbf{x} - \mathbf{b}\|_2 \leq \varepsilon \|\mathbf{b}\|_2. \tag{27}$$

Moreover, if $\mathbf{M}$ *is RCDD, a single such Eulerian linear system solve suffices.*

Combining Corollary 40, Proposition 42, and a union bound then yields the following.

Corollary 43 (Directed Laplacian solver)**.** *Given* $\vec{G} = (V, E, \mathbf{w})$ *with* $|V| = n, |E| = m, \mathbf{w} \in [1, U]^E$*,* $\mathbf{b} \in \mathbb{R}^V$*, and error parameter* $\varepsilon \in (0, 1)$*, there is an algorithm running in time*

$$\breve{O}\left(m \log^2\left(\frac{nU}{\delta\varepsilon}\right) \log^3\left(\frac{nU}{\varepsilon}\right) + n \log^2(nU) \log^3\left(\frac{nU}{\delta\varepsilon}\right) \log^3\left(\frac{nU}{\varepsilon}\right)\right)$$

which returns $\mathbf{x} \in \mathbb{R}^V$ *satisfying, with probability* $\geq 1 - \delta$*,*

$$\left\|\mathbf{x} - \vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\right\|_{\mathbf{L}_G} \leq \varepsilon \left\|\vec{\mathbf{L}}_{\vec{G}}^{\dagger}\mathbf{b}\right\|_{\mathbf{L}_G}, \text{ where } G \stackrel{\text{def}}{=} \text{und}(\vec{G}).$$

Finally, we mention a number of results from [CKPPSV16; CKPPRSV17; AJSS19] which leverage RCDD solvers as a black box. Plugging Corollary 40, Proposition 42, and Corollary 43 into these results, we obtain the following runtimes. For simplicity, we only consider problems with $\text{poly}(n)$-bounded conditioning and $\text{poly}(\frac{1}{n})$-bounded failure probability, and let $\mathcal{T}_{\text{solve}}(m, n, \varepsilon) \stackrel{\text{def}}{=} \breve{O}(m \log^2(n) \log(\frac{n}{\varepsilon}) + n \log^5(n) \log(\frac{n}{\varepsilon}))$ be the runtime of our Eulerian Laplacian solver.

- **Stationary distributions.** We can compute a vector within ℓ_2 distance ε of the stationary distribution of a random walk on a directed graph in time $\mathcal{T}_{\text{solve}}(m, n, 1) \cdot O(\log^2(\frac{n}{\varepsilon}))$.

- **Random walks.** We can compute the escape probability, hitting times and commute times for a random walk on a directed graph to ε additive error in time $\mathcal{T}_{\text{solve}}(m,n,1)\cdot O(\log^2(\frac{n}{\varepsilon}))$.
- **Mixing time.** We can compute an ε-multiplicative approximation of the mixing time of a random walk on a directed graph in time $\mathcal{T}_{\text{solve}}(m,n,1)\cdot O(\log^2(\frac{n}{\varepsilon}))$.
- **PageRank.** We can compute a vector within ℓ_2 distance ε of the Personalized PageRank vector with restart probability β on a directed graph in time $\mathcal{T}_{\text{solve}}(m,n,1)\cdot O(\log^2(\frac{n}{\beta})+\log(\frac{1}{\varepsilon}))$.
- **M-matrix linear systems.** We can compute a vector achieving relative accuracy ε (in the sense of (27)) to a linear system in an M-matrix $\mathbf{M}$ in time

$$\mathcal{T}_{\text{solve}}(m,n,\varepsilon)\cdot O\left(\log^2(n)\log\left(\frac{\|\mathbf{M}^{-1}\|_{1\to 1}+\|\mathbf{M}^{-1}\|_{\infty\to\infty}}{\varepsilon}\right)\right).$$

- **Perron-Frobenius theory.** Given a nonnegative matrix $\mathbf{A}\in\mathbb{R}^{n\times n}$ with m nonzero entries, we can find $s\in\mathbb{R}$ and $\mathbf{v}_l,\mathbf{v}_r\in\mathbb{R}^n$ such that $\frac{s}{\rho(\mathbf{A})}\in[1,1+\varepsilon]$,[6] $\|\mathbf{A}\mathbf{v}_r-s\mathbf{v}_r\|_\infty\le\varepsilon\|\mathbf{v}_r\|_\infty$, and $\|\mathbf{A}^\top\mathbf{v}_l-s\mathbf{v}_l\|_\infty\le\varepsilon\|\mathbf{v}_l\|_\infty$, in time

$$\mathcal{T}_{\text{solve}}(m,n,\varepsilon)\cdot O\left(\log^3\left(\frac{\|\mathbf{A}\|_{1\to 1}+\|\mathbf{A}\|_{\infty\to\infty}}{\varepsilon\rho(\mathbf{A})}\right)\right).$$

9 Graphical spectral sketches

In this section, we give an additional application of the techniques we developed for efficiently constructing Eulerian sparsifiers in Sections 6 and 7. Specifically, we show that they yield improved constructions of the following graph-theoretic object, originally introduced in [ACKQWZ16; JS18; CGPSSW18] in the undirected graph setting.

Definition 44 (Graphical spectral sketch)**.** *Given a undirected graph $G=(V,E,\mathbf{w})$, a distribution $\mathcal{H}$ over random undirected graphs $H=(V,E',\mathbf{w}')$ with $E'\subseteq E$ is said to be a* (ε,δ)-graphical spectral sketch for G *if for any fixed vector $\mathbf{x}\in\mathbb{R}^V$, with probability $\ge 1-\delta$ over the sample $H\sim\mathcal{H}$, we have*

$$\left|\mathbf{x}^\top(\mathbf{L}_H-\mathbf{L}_G)\mathbf{x}\right|\le\varepsilon\cdot\mathbf{x}^\top\mathbf{L}_G\mathbf{x}.$$

We generalize Definition 44 to the Eulerian graph setting (which to our knowledge has not been studied before), and show that our primitives extend to capture this generalization.

Definition 45 (Eulerian graphical spectral sketch)**.** *Given an Eulerian graph $\vec{G}=(V,E,\mathbf{w})$, a distribution $\mathcal{H}$ over random Eulerian graphs $\vec{H}=(V,E',\mathbf{w}')$ with $E'\subseteq E$ is said to be a* (ε,δ)-Eulerian graphical spectral sketch for $\vec{G}$ *if for any fixed vectors $\mathbf{a},\mathbf{z}\in\mathbb{R}^V$, with probability $\ge 1-\delta$ over the sample $\vec{H}\sim\mathcal{H}$, we have for $G\stackrel{\text{def}}{=}\text{und}(\vec{G})$,*

$$\left|\mathbf{a}^\top(\vec{\mathbf{L}}_{\vec{H}}-\vec{\mathbf{L}}_{\vec{G}})\mathbf{z}\right|\le\varepsilon\cdot\|\mathbf{a}\|_{\mathbf{L}_G}\|\mathbf{z}\|_{\mathbf{L}_G}. \tag{28}$$

Our algorithm closely follows the framework of [JS18; CGPSSW18]. We aim to recursively reduce a constant fraction of the edges while keeping a small additive error for a bilinear form

[6] $\rho(\mathbf{A})$ is the spectral radius of $\mathbf{A}$: $\rho(\mathbf{A})\stackrel{\text{def}}{=}\lim_{k\to\infty}\|\mathbf{A}^k\|_{\text{op}}^{1/k}$.

applied to fixed vectors $\mathbf{a}, \mathbf{z}$, as in (28). Similar to our spectral sparsification algorithm in Section 7, we repeat this process for $O(\log n)$ phases. Within each phase, we accomplish our goal by first using an expander decomposition from prior work [ADK23], and then within each piece, we restrict to a subgraph on vertices with sufficiently large combinatorial (unweighted) degrees. At this point, Cheeger's inequality (Lemma 50) gives us an effective resistance diameter bound on the decomposition piece as well, so we can use most of the guarantees from Section 7 directly. We are able to obtain the tighter per-vector pair parameter tradeoff required by spectral sketches by exploiting a tighter connection between the Laplacian and degree matrices within expanders, as in [JS18] which used this for undirected spectral sketches. This is used alongside a key degree-based spectral inequality from [CGPSSW18] (see Lemma 51).

9.1 Degree-preserving primitives

In this section, we give several basic helper results which we use to ensure degree-preserving properties of our algorithms by working with bipartite lifts. Given a directed graph $\vec{G} = (V, E, \mathbf{w})$, we let the directed graph $\vec{G}^{\uparrow} \stackrel{\text{def}}{=} \text{blift}(\vec{G})$ be its bipartite lift, which is defined so that $V_{\vec{G}^{\uparrow}} = V \cup \bar{V}$ where $\bar{V}$ is a copy of V, and $E_{\vec{G}^{\uparrow}} = \{f = (u, \bar{v}) | (u, v) \in E\}$ with $\mathbf{w}_{u,\bar{v}} = \mathbf{w}_{(u,v)}$. Notice that our definition gives a canonical bijection between $E_{\vec{G}^{\uparrow}}$ and $E_{\vec{G}}$.

Lemma 46. *Let $\vec{G} = (V, E, \mathbf{w})$ be a directed graph and let its bipartite lift be $\vec{G}^{\uparrow} \stackrel{\text{def}}{=} (V \cup V', E^{\uparrow}, \mathbf{w}) = \text{blift}(\vec{G})$, with $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$ and $G^{\uparrow} \stackrel{\text{def}}{=} \text{und}(\vec{G}^{\uparrow})$. Suppose that for some $\varepsilon > 0$, $\mathbf{w}' \in \mathbb{R}^{E}_{>0}$ satisfies*

$$\mathbf{B}_{\vec{G}^{\uparrow}}^{\top} \mathbf{w}' = \mathbf{B}_{\vec{G}^{\uparrow}}^{\top} \mathbf{w}, \quad \left\| \mathbf{L}_{G^{\uparrow}}^{\dagger \frac{1}{2}} \mathbf{B}_{\vec{G}^{\uparrow}}^{\top} (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{G}^{\uparrow}} \mathbf{L}_{G^{\uparrow}}^{\dagger \frac{1}{2}} \right\|_{\text{op}} \leq \varepsilon.$$

Then, letting $|\mathbf{B}_{\vec{G}}|$ apply the absolute value entrywise,

$$\mathbf{B}_{\vec{G}}^{\top} \mathbf{w}' = \mathbf{B}_{\vec{G}}^{\top} \mathbf{w}, \quad |\mathbf{B}_{\vec{G}}|^{\top} \mathbf{w}' = |\mathbf{B}_{\vec{G}}|^{\top} \mathbf{w}, \quad \left\| \mathbf{L}_{G}^{\dagger \frac{1}{2}} \mathbf{B}_{\vec{G}}^{\top} (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{G}} \mathbf{L}_{G}^{\dagger \frac{1}{2}} \right\|_{\text{op}} \leq \varepsilon.$$

Proof. Consider the edge-vertex incidence matrix of $\vec{G}$, $\mathbf{B}_{\vec{G}} = \mathbf{H}_{\vec{G}} - \mathbf{T}_{\vec{G}}$. The edge-vertex incidence matrix of $\vec{G}^{\uparrow}$ is then $\mathbf{B}_{\vec{G}^{\uparrow}} = \begin{pmatrix} \mathbf{H}_{\vec{G}} & -\mathbf{T}_{\vec{G}} \end{pmatrix}$. Hence, any vector $\mathbf{x}$ satisfying $\mathbf{B}_{\vec{G}^{\uparrow}}^{\top} \mathbf{x} = \mathbf{0}_V$ must have $\mathbf{H}_{\vec{G}}^{\top} \mathbf{x} = \mathbf{0}_V, \mathbf{T}_{\vec{G}}^{\top} \mathbf{x} = \mathbf{0}_V$, giving us preservation of both the difference between in and out degrees and the sum of in and out degrees $\vec{G}$, i.e.,

$$\mathbf{B}_{\vec{G}}^{\top} \mathbf{x} = \mathbf{0}_V, \ |\mathbf{B}_{\vec{G}}|^{\top} \mathbf{x} = \mathbf{0}_V,$$

where we used that $\mathbf{B}_{\vec{G}} = \mathbf{H}_{\vec{G}} - \mathbf{T}_{\vec{G}}$ and $|\mathbf{B}_{\vec{G}}| = \mathbf{H}_{\vec{G}} + \mathbf{T}_{\vec{G}}$. Taking $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{w}' - \mathbf{w}$ then gives the the first two claims. We remark that the directed graph $\vec{G}$ need not be Eulerian.

We proceed to prove the third claim. For ease of notation, we omit $\vec{G}$ in the subscripts of the matrices and denote $\mathbf{A} \stackrel{\text{def}}{=} \mathbf{A}_{\vec{G}}$, $\mathbf{A}_{\uparrow} \stackrel{\text{def}}{=} \mathbf{A}_{\vec{G}^{\uparrow}}$ for all matrices $\mathbf{A}$. We also use the following equivalent definition of operator norms with the convention that the fraction is 0 if the numerator is 0:

$$\|\mathbf{A}\|_{\text{op}} = \max_{\mathbf{x}, \mathbf{y}} \frac{|\mathbf{x}^{\top} \mathbf{A} \mathbf{y}|}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2},$$

where the max is over $\mathbf{x}, \mathbf{y}$ of compatible dimensions. Also, we let $\mathbf{Q} \in \mathbb{R}^{(V \cup \bar{V}) \times V}$ be defined by

$$\mathbf{Q} \mathbf{e}_v = \begin{pmatrix} \mathbf{e}_v \\ \mathbf{e}_{\bar{v}} \end{pmatrix}$$

for all $v \in V$, where $\bar{v}$ is identified with v. Notice that $\mathbf{H} = \mathbf{H}_\uparrow \mathbf{Q}$, $\mathbf{T} = \mathbf{T}_\uparrow \mathbf{Q}$ and $\mathbf{B} = \mathbf{B}_\uparrow \mathbf{Q}$. Then,

$$\mathbf{Q}^\top \mathbf{L}_\uparrow \mathbf{Q} = \mathbf{Q}^\top \mathbf{B}_\uparrow^\top \mathbf{W} \mathbf{B}_\uparrow \mathbf{Q} = \mathbf{B}^\top \mathbf{W} \mathbf{B} = \mathbf{L}.$$

Finally, for any non-trivial vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^V$ satisfying $\mathbf{x}, \mathbf{y} \perp \mathbf{1}_V$, and defining

$$\mathbf{a} \stackrel{\text{def}}{=} \mathbf{L}^{\dagger\frac{1}{2}} \mathbf{x},\ \mathbf{b} \stackrel{\text{def}}{=} \mathbf{L}^{\dagger\frac{1}{2}} \mathbf{y},\ \hat{\mathbf{x}} \stackrel{\text{def}}{=} \mathbf{L}_\uparrow^{\dagger\frac{1}{2}} \mathbf{Q}\mathbf{a},\ \hat{\mathbf{y}} \stackrel{\text{def}}{=} \mathbf{L}_\uparrow^{\dagger\frac{1}{2}} \mathbf{Q}\mathbf{b},$$

we have

$$\begin{aligned}
\frac{|\mathbf{x}^\top \mathbf{L}^{\dagger\frac{1}{2}} \mathbf{B}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H} \mathbf{L}^{\dagger\frac{1}{2}} \mathbf{y}|}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} &= \frac{|\mathbf{a}^\top \mathbf{B}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H} \mathbf{b}|}{\|\mathbf{a}\|_{\mathbf{L}} \|\mathbf{b}\|_{\mathbf{L}}} \\
&= \frac{|\mathbf{a}^\top \mathbf{Q}^\top \mathbf{B}_\uparrow^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_\uparrow \mathbf{Q} \mathbf{b}|}{\|\mathbf{Q}\mathbf{a}\|_{\mathbf{L}_\uparrow} \|\mathbf{Q}\mathbf{b}\|_{\mathbf{L}_\uparrow}} \\
&= \frac{|\hat{\mathbf{x}}^\top \mathbf{L}_\uparrow^{\dagger\frac{1}{2}} \mathbf{B}_\uparrow^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_\uparrow \mathbf{L}_\uparrow^{\dagger\frac{1}{2}} \hat{\mathbf{y}}|}{\|\hat{\mathbf{x}}\|_2 \|\hat{\mathbf{y}}\|_2},
\end{aligned}$$

giving us the desired operator norm bound. We note that $\mathbf{Q}\mathbf{a}, \mathbf{Q}\mathbf{b} \perp \ker(\mathbf{L}_\uparrow)$.

Finally, we record a consequence of this proof we will later use. Recall that we have shown

$$\mathbf{H} = \mathbf{H}_\uparrow \mathbf{Q}, \quad \mathbf{T} = \mathbf{T}_\uparrow \mathbf{Q}, \quad \mathbf{B} = \mathbf{B}_\uparrow \mathbf{Q}, \quad \mathbf{Q}^\top \mathbf{L}_\uparrow \mathbf{Q} = \mathbf{L}.$$

Therefore, suppose that for some $\varepsilon > 0$ and fixed vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^V$, $\mathbf{w}' \in \mathbb{R}^E_{>0}$ satisfies

$$\left|\mathbf{x}^\top \mathbf{Q}^\top \mathbf{B}_\uparrow^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_\uparrow \mathbf{Q} \mathbf{y}\right| \leq \varepsilon \cdot \|\mathbf{Q}\mathbf{x}\|_{\mathbf{L}_\uparrow} \|\mathbf{Q}\mathbf{y}\|_{\mathbf{L}_\uparrow}. \tag{29}$$

Then, we also have the bound in the unlifted graph G,

$$\left|\mathbf{x}^\top \mathbf{B}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H} \mathbf{y}\right| \leq \varepsilon \cdot \|\mathbf{x}\|_{\mathbf{L}} \|\mathbf{y}\|_{\mathbf{L}}. \tag{30}$$

□

9.2 Expander decomposition and sketching by degrees

In this section, we provide guarantees on our earlier BasicFastSparsify (Algorithm 5) which hold when the algorithm is passed a expander graph, that is also a bipartite lift, as input. We first recall the definition of an expander graph, parameterized by a minimum conductance threshold ϕ.

Definition 47 (Expander graphs)**.** *Let $G = (V, E, \mathbf{w})$ be an undirected graph, and let $\mathbf{d} \in \mathbb{R}^V_{\geq 0}$ be its weighted degrees. For a set $S \subseteq V$, let $\mathrm{Vol}(S) = \sum_{v \in S} \mathbf{d}_v$ be the sum of weighted degrees in S, and let $\partial S = \{e = (u, v) \in E \mid u \in S \text{ and } v \notin S\}$ be the edge boundary of S. We define the cut value of S by $\mathbf{w}(\partial S) \stackrel{\text{def}}{=} \sum_{e \in \partial S} \mathbf{w}_e$, and the conductance of S by*

$$\Phi(S) = \frac{\mathbf{w}(\partial S)}{\min\{\mathrm{Vol}(S), \mathrm{Vol}(V \backslash S)\}}.$$

*Finally, we say G is a ϕ-*expander *if $\Phi(S) \geq \phi$ for all $S \subseteq V$.*

An important algorithmic primitive related to Definition 47 is an expander decomposition.

Definition 48 (Expander decomposition)**.** *We call* $\{G_i\}_{i\in[I]}$ *a* (ϕ, r, J)-expander decomposition *if* $\{G_i\}_{i\in[I]}$ *are edge-disjoint subgraphs of* $G = (V, E, \mathbf{w})$*, and the following hold.*

1. Bounded weight ratio*: For all* $i \in [I]$, $\frac{\max_{e\in E(G_i)} \mathbf{w}_e}{\min_{e\in E(G_i)} \mathbf{w}_e} \leq r$.
2. Conductance*: For all* $i \in [I]$, $\Phi(G_i) \geq \phi$.
3. Edges cut*:* $|E(G) \setminus (\bigcup_{i\in[I]} E(G_i))| \leq \frac{m}{2}$.
4. Vertex coverage*: Every vertex* $v \in V(G)$ *appears in at most* J *of the subgraphs.*

We recall the state-of-the-art expander decomposition algorithm in the literature, which will later be used in conjunction with the subroutines developed in this section.

Proposition 49 (Theorem 4.4, [ADK23])**.** *There is an algorithm* ExpanderDecompADK(G, r, δ) *that, given as input undirected* $G = (V, E, \mathbf{w})$ *with* $\frac{\max_{e\in\mathrm{supp}(\mathbf{w})} \mathbf{w}_e}{\min_{e\in\mathrm{supp}(\mathbf{w})} \mathbf{w}_e} \leq W$ *and* $r \geq 1$*, computes in time*

$$O\left(m \log^6(n) \log\left(\frac{n}{\delta}\right)\right)$$

a $(C_{\mathrm{ADK}} \log^{-2}(n), r, \log_r W+3)$*-expander decomposition of* G *with probability* $\geq 1-\delta$*, for a universal constant* C_{ADK}*.*[7]

We further require two spectral inequalities based on the expansion.

Lemma 50 (Cheeger's inequality)**.** *If* G *is a* ϕ*-expander with Laplacian* $\mathbf{L}_G$ *and degrees* $\mathbf{D}_G$*, then*

$$\lambda_2\left(\mathbf{D}_G^{\dagger\frac{1}{2}} \mathbf{L}_G \mathbf{D}_G^{\dagger\frac{1}{2}}\right) \geq \frac{\phi^2}{2}.$$

Lemma 51 (Lemma 6.6, [CGPSSW18])**.** *If* $G = (V, E, \mathbf{w})$ *is a* ϕ*-expander that satisfies, for some* $\bar{w} > 0$, $\mathbf{w}_e \in [\bar{w}, 2\bar{w}]$ *for all* $e \in E$*, then for any* $\mathbf{x} \in \mathbb{R}^V$*,*

$$\|\mathbf{x}\|_{\mathbf{L}_G}^2 \geq \frac{\phi^2 \bar{w}}{2} \cdot \sum_{v\in V} [\mathbf{deg}_G]_v (\mathbf{x}_v - \hat{x})^2,$$

where $\mathbf{deg}_G \in \mathbb{N}_{\geq 0}^V$ *is the combinatorial (unweighted) degrees of* G*, and* $\hat{x} \stackrel{\mathrm{def}}{=} \frac{\mathbf{d}_G^\top \mathbf{x}}{\|\mathbf{d}_G\|_1}$.

Importantly, Lemma 51 allows us to obtain improved tradeoffs on how well spectral sketch guarantees are preserved in Lemma 55, by first lower bounding degrees of vertices under consideration. Next, we show that expander graphs with small weight ratio form clusters (Definition 33), which make them compatible with our algorithm BasicFastSparsify.

Lemma 52. *Let* $\vec{G} = (V, E, \mathbf{w})$ *and let* $G = \mathrm{und}(\vec{G})$*. Suppose* G *is a* ϕ*-expander and that for all* $e \in E$, $\mathbf{w}_e \in [\bar{w}, 2\bar{w}]$ *for some* $\bar{w} > 0$*. Given any* $\beta > 0$*, let* $U \subseteq V$ *be the set of vertices with* $[\mathbf{deg}_G]_u \geq \beta$ *for every* $u \in U$*. Then, the subgraph* $\vec{G}[U]$ *is a* $(\bar{w}, 8\beta^{-1}\phi^{-2})$*-cluster in* $\vec{G}$*.*

[7] The algorithm of [ADK23] is stated for $n^{-O(1)}$ failure probabilities, but examining Section 5.3, the only place where randomness is used in the argument, shows that we can obtain failure probability δ at the stated overhead. The vertex coverage parameter $\log_r(W) + 3$ is due to bucketing the edges by weight, analogously to the proof of Proposition 10. We note that there is no $\log_r(W)$ overhead in the runtime, as the edges in each piece are disjoint.

Proof. Let $\mathbf{D}_G \in \mathbb{R}^{V\times V}$ be the diagonal matrix whose diagonal is the weighted degrees of G. By Cheeger's inequality (Lemma 50), for any pair of distinct vertices $a, b \in U$, we have the desired

$$\begin{aligned}\mathbf{w}_e \mathrm{ER}_G(a,b) &\le 2\bar{w}\mathbf{b}_{a,b}^\top \mathbf{L}_G^\dagger \mathbf{b}_{a,b} \le 4\bar{w}\phi^{-2}\mathbf{b}_{(a,b)}^\top \mathbf{D}_G^{-1}\mathbf{b}_{(a,b)} \\ &\le 4\phi^{-2}\left(\frac{1}{[\mathbf{deg}_G]_a} + \frac{1}{[\mathbf{deg}_G]_b}\right) \le 8\phi^{-2}\beta^{-1}.\end{aligned}$$

□

Finally, we state one additional sketching property enjoyed by BasicFastSparsify in Lemma 53. We mention that this is the key step in our proof where we require that our input graph is a bipartite lift of a directed graph. We exploit this property by employing machinery from Section 9.1.

Lemma 53. *Suppose* BasicFastSparsify *is given input $\vec{C}$ instead of $\vec{G}$ where $\vec{C}$ is a subgraph of $\vec{G} = (V, E, \mathbf{w})$, a bipartite lift of a directed graph, satisfying $V = A \cup B$ and $E \subseteq A \times B$, $\vec{H}_\star$ is a $(\bar{w}, \rho)$-cluster of $\vec{C}_\star = (V(\vec{C}), E(\vec{C}), [\mathbf{w}_\star]_{E(\vec{C})})$, and T is a subgraph of $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$ and $E(T) \cap E(C) = \emptyset$, where $C \stackrel{\text{def}}{=} \mathrm{und}(\vec{C})$. Under the same assumptions as Lemma 34, with probability $\ge 1-\delta$, Items 1 to 4 and the runtime of Lemma 34 still hold.*

In addition, $|\mathbf{B}_{\vec{G}}|\mathbf{w} = |\mathbf{B}_{\vec{G}}|\mathbf{w}'$, and for fixed $\mathbf{a}, \mathbf{z} \in \mathbb{R}^V$,

$$\left|\mathbf{a}^\top \mathbf{B}_{\vec{G}}^\top (\mathbf{W}' - \mathbf{W})\mathbf{H}_{\vec{G}}\mathbf{z}\right| \le C_{\mathrm{BFS}} \cdot \alpha\bar{w}\sqrt{\log\left(\frac{1}{\delta}\right)} \|\mathbf{a}\|_{\mathbf{\Pi}_{V(\vec{C})}} \|\mathbf{z}\|_{\mathbf{\Pi}_{V(\vec{C})}} + \varepsilon \cdot \|\mathbf{a}\|_{\mathbf{L}_G} \|\mathbf{z}\|_{\mathbf{L}_G}. \tag{31}$$

Proof. Items 2 and 3 and the second claim in Item 1 of Lemma 34 are not affected by the change in input. Further, the first claim in Item 1 follows by Lemma 24, where the relevant edge-vertex incidence matrix remains $\mathbf{B}_{\vec{G}}$ (since T is a subgraph of G). By the same argument in the proof of the second equation in Lemma 46, the assumption that $\vec{G}$ is a bipartite lift gives $|\mathbf{B}_{\vec{G}}|\mathbf{w} = |\mathbf{B}_{\vec{G}}|\mathbf{w}'$. Thus, it suffices to discuss Item 4 and (31).

We now prove Item 4. Note that the key difference is that we assume $\vec{H}_\star$ is a $(\bar{w}, \rho)$-cluster, when ERs are measured through C instead of G. Under the assumption that ProjMinusRankOne is exact, the same argument as in the proof of Lemma 34 gives that

$$\left\|\mathbf{L}_C^{\dagger\frac{1}{2}} \mathbf{B}_{\vec{C}}^\top (\mathbf{W}' - \mathbf{W})\mathbf{H}_{\vec{C}} \mathbf{L}_C^{\dagger\frac{1}{2}}\right\|_{\mathrm{op}} \le C_{\mathrm{BFS}}\sqrt{\alpha\rho\log\left(\frac{m}{\delta}\right)}. \tag{32}$$

Since $\mathrm{supp}(\mathbf{w}' - \mathbf{w}) \subseteq E(\vec{C})$ and $\mathbf{L}_C \preccurlyeq \mathbf{L}_G$, we obtain the first term in the inequality in Item 4 of Lemma 34 (i.e., without the additive ε) under an exact ProjMinusRankOne. The error due to inexactness is then handled the same way as in Item 4 of Lemma 34, since the spectral error guarantees of Lemma 24 are measured with respect to G, not C.

In the remainder of the proof, we handle (31). We first consider the sketching error assuming ProjMinusRankOne is exact. In iteration t, the difference to the directed Laplacian is:

$$\begin{aligned}\mathbf{B}_{\vec{C}}^\top (\mathbf{W}_{t+1} - \mathbf{W}_t)\mathbf{H}_{\vec{C}} &= \eta \sum_{e\in E(\vec{H}_t)} [\mathbf{w}_t]_e \left[\mathbf{P}_{\vec{H}_t, \mathbf{w}_t}\mathbf{s}\right]_e \mathbf{b}_e \mathbf{h}_e^\top \\ &= \eta \sum_{e\in E(\vec{H}_t)} \mathbf{s}_e \sum_{f\in E(\vec{H}_t)} [\mathbf{P}_{\vec{H}_t, \mathbf{w}_t}]_{fe} [\mathbf{w}_t]_f \mathbf{b}_f \mathbf{h}_f^\top.\end{aligned} \tag{33}$$

By the third equality of Lemma 27, $\mathbf{B}_{\vec{C}}^\top \mathbf{W}_t [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e} = \mathbf{0}_E$ for any $e \in E(\vec{H}_t)$, i.e., $\mathbf{W}_t [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}$ is a circulation on the graph $\vec{C}$. Again, by the same argument in Lemma 46, we have

$$|\mathbf{B}_{\vec{C}}|^\top \mathbf{W}_t [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e} = \mathbf{0}_V, \tag{34}$$

i.e., both in-degrees and out-degrees are preserved.

Next, because all edges in $\vec{C}$ are from A to B, we have $[|\mathbf{B}_{\vec{C}}|]_{:A} = \mathbf{H}_{\vec{C}}$. Then, we compute that

$$\begin{aligned} \sum_{f \in E(\vec{H}_t)} [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{fe} \mathbf{w}_f \mathbf{h}_f \mathbf{h}_f^\top &= [|\mathbf{B}_{\vec{C}}|]_{:A}^\top \mathbf{W}_t \mathbf{diag}\left([\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}\right) \mathbf{H}_{\vec{C}} \\ &= \mathbf{diag}\left([|\mathbf{B}_{\vec{C}}|]_{:A}^\top \mathbf{W}_t [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}\right). \end{aligned} \tag{35}$$

Combining (34) and (35) shows:

$$\sum_{f \in E(\vec{H}_t)} [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{fe} \mathbf{w}_f \mathbf{b}_f \mathbf{h}_f^\top = \sum_{f \in E(\vec{H}_t)} [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{fe} \mathbf{w}_f \mathbf{t}_f \mathbf{h}_f^\top. \tag{36}$$

In addition, we have by Fact 6 and that $\mathbf{W}_t [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}$ is a circulation that the following hold:

$$\begin{aligned} \mathbf{B}_{\vec{C}}^\top \mathbf{W}_t \mathbf{diag}\left([\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}\right) \mathbf{H}_{\vec{C}} \mathbf{1}_V &= \mathbf{B}_{\vec{C}}^\top \mathbf{W}_t [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e} = \mathbf{0}_V, \\ \mathbf{1}_V^\top \mathbf{B}_{\vec{C}}^\top \mathbf{W}_t \mathbf{diag}\left([\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}\right) \mathbf{H}_{\vec{C}} &= -\mathbf{1}_V^\top \mathbf{T}_{\vec{C}}^\top \mathbf{W}_t \mathbf{diag}\left([\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}\right) \mathbf{B}_{\vec{C}} \\ &= -[\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}^\top \mathbf{W}_t \mathbf{B}_{\vec{C}} = \mathbf{0}_V^\top. \end{aligned} \tag{37}$$

Recalling the formula (33), and summing (37) over all $e \in E(\vec{H}_t)$, gives

$$\mathbf{B}_{\vec{C}}^\top (\mathbf{W}_{t+1} - \mathbf{W}_t) \mathbf{H}_{\vec{C}} \mathbf{1}_V = \sum_{e \in E(\vec{H}_t)} \mathbf{s}_e \mathbf{0}_V = \mathbf{0}_V, \quad \mathbf{1}_V^\top \mathbf{B}_{\vec{C}}^\top (\mathbf{W}_{t+1} - \mathbf{W}_t) \mathbf{H}_{\vec{C}} = \mathbf{0}_V^\top. \tag{38}$$

Define for each $e \in E(\vec{H}_t)$ a scalar $\mathbf{x}_e^{(t)}$ by:

$$\mathbf{x}_e^{(t)} \stackrel{\text{def}}{=} [\mathbf{w}_t]_e \mathbf{a}^\top \mathbf{\Pi}_{V(\vec{C})} \mathbf{t}_e \mathbf{h}_e^\top \mathbf{\Pi}_{V(\vec{C})} \mathbf{z},$$

where we recall that $\mathbf{\Pi}_{V(\vec{C})} = \mathbf{I}_{V(\vec{C})} - \frac{1}{|V(\vec{C})|} \mathbf{1}_{V(\vec{C})} \mathbf{1}_{V(\vec{C})}^\top$. We showed in (38) that $\mathbf{1}_V$, and hence $\mathbf{1}_{V(\vec{C})}$, is in the left and right kernel of $\mathbf{B}_{\vec{C}}^\top (\mathbf{W}_{t+1} - \mathbf{W}_t) \mathbf{H}_{\vec{C}}$. Combining (33) and (36) then yields

$$\begin{aligned} \mathbf{a}^\top \mathbf{B}_{\vec{C}}^\top (\mathbf{W}_{t+1} - \mathbf{W}_t) \mathbf{H}_{\vec{C}} \mathbf{z} &= \mathbf{a}^\top \mathbf{\Pi}_{V(\vec{C})} \mathbf{B}_{\vec{C}}^\top (\mathbf{W}_{t+1} - \mathbf{W}_t) \mathbf{H}_{\vec{C}} \mathbf{\Pi}_{V(\vec{C})} \mathbf{z} \\ &= \eta \sum_{e \in E(\vec{H}_t)} \mathbf{s}_e \sum_{f \in E(\vec{H}_t)} [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{fe} \mathbf{x}_f^{(t)}. \end{aligned} \tag{39}$$

Each $\eta \sum_{f \in E(\vec{H}_t)} [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{fe} \mathbf{x}_f \cdot \mathbf{s}_e$ is sub-Gaussian with parameter $\sigma_e \stackrel{\text{def}}{=} \eta |\langle [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}, \mathbf{x}^{(t)} \rangle|$. Therefore, the left-hand side of (39) is sub-Gaussian with parameter

$$\eta^2 \sum_{e \in E(\vec{H}_t)} \left\langle [\mathbf{P}_{\vec{H}_t,\mathbf{w}_t}]_{:e}, \mathbf{x}^{(t)} \right\rangle^2 = \eta^2 \mathbf{x}^\top \mathbf{P}_{\vec{H}_t,\mathbf{w}_t} \mathbf{x}^{(t)} \le \eta^2 \left\| \mathbf{x}^{(t)} \right\|_2^2,$$

where $\left\|\mathbf{x}^{(t)}\right\|_2^2$ can be bounded, using the definition of α and Item 1, by

$$\begin{aligned}\left\|\mathbf{x}^{(t)}\right\|_2^2 &= \sum_{e\in E(\vec{H}_t)} \mathbf{w}_e^2 [\mathbf{\Pi}_{V(\vec{C})}\mathbf{a}]_{t(e)}^2 [\mathbf{\Pi}_{V(\vec{C})}\mathbf{z}]_{h(e)}^2 \\ &\le 8\alpha^2\bar{w}^2 \left(\sum_{v\in V(\vec{C})} [\mathbf{\Pi}_{V(\vec{C})}\mathbf{a}]_v^2\right)\left(\sum_{u\in V(\vec{C})} [\mathbf{\Pi}_{V(\vec{C})}\mathbf{z}]_u^2\right) \\ &= 8\alpha^2\bar{w}^2 \left\|\mathbf{a}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 \left\|\mathbf{z}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 .\end{aligned}$$

Summing over at most τ iterations, the total sub-Gaussian parameter of $\mathbf{a}^\top \mathbf{B}_{\vec{C}}^\top (\mathbf{W}' - \mathbf{W})\mathbf{H}_{\vec{C}}\mathbf{z}$ is:

$$\tau \cdot 8\alpha^2\bar{w}^2\eta^2 \cdot \left\|\mathbf{a}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 \left\|\mathbf{z}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 \le 6000\alpha^2\bar{w}^2 \cdot \left\|\mathbf{a}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 \left\|\mathbf{z}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 .$$

Standard sub-Gaussian concentration finally yields, with probability $1 - \frac{\delta}{4}$, the desired

$$\left|\mathbf{a}^\top \mathbf{B}_{\vec{G}}^\top (\mathbf{W}' - \mathbf{W})\mathbf{H}_{\vec{G}}\mathbf{z}\right| \le C_{\text{BFS}} \cdot \alpha\bar{w}\sqrt{\log\left(\frac{1}{\delta}\right)} \left\|\mathbf{a}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 \left\|\mathbf{z}\right\|_{\mathbf{\Pi}_{V(\vec{C})}}^2 .$$

Following the notation in Lemma 34, conditioning on the events $\bigcup_{0\le t\le\tau} \mathcal{E}_t$ does not affect the proof, for the same reason as outlined in Lemma 34: if any $\mathcal{E}_t$ fails, we set all future weight updates to zero in the scalar martingale. Finally, as T is edge-disjoint from C, the additive spectral error term due to the inexactness of ProjMinusRankOne and the final rounding in each iteration is measured with respect to $\mathbf{L}_G$, as is done in Lemma 29. Applying this additive spectral error to vectors $\mathbf{a}$ and $\mathbf{z}$ gives an additive term of $\varepsilon \cdot \left\|\mathbf{a}\right\|_{\mathbf{L}_G} \left\|\mathbf{z}\right\|_{\mathbf{L}_G}$. This completes our proof. □

Corollary 54. *Consider calling* BasicFastSparsify *I times, all with shared parameters $\mathbf{w}_\star, \ell, \delta, \varepsilon$, but on edge-disjoint subgraphs $\{\vec{H}_i\}_{i\in[I]}$ and $\{\vec{C}_i\}_{i\in[I]}$ of $\vec{G} = (V, E, \mathbf{w})$, a bipartite lift of a directed graph, with $V = A \cup B$ and $E \subseteq A \times B$, so that each corresponding $[\vec{H}_\star]_i$ is a $(\bar{w}_i, \rho)$-cluster in $\vec{C}_i$ for some value of $\bar{w}_i$. Then with probability $\ge 1 - \delta I$, the runtime and spectral error guarantee in Corollary 35 still hold. In addition, $|\mathbf{B}_{\vec{G}}|^\top \mathbf{w} = |\mathbf{B}_{\vec{G}}|^\top \mathbf{w}'$, and for fixed $\mathbf{a}, \mathbf{z} \in \mathbb{R}^V$, for all $i \in I$,*

$$\left|\mathbf{a}^\top \mathbf{B}_{\vec{G}}^\top (\mathbf{W}' - \mathbf{W})\mathbf{H}_{\vec{G}}\mathbf{z}\right| \le C_{\text{BFS}} \cdot \alpha\sqrt{\log\left(\frac{1}{\delta}\right)} \sum_{i\in[I]} \bar{w}_i \left\|\mathbf{a}\right\|_{\mathbf{\Pi}_{V(\vec{C}_i)}} \left\|\mathbf{z}\right\|_{\mathbf{\Pi}_{V(\vec{C}_i)}} + \varepsilon I \cdot \left\|\mathbf{a}\right\|_{\mathbf{L}_G} \left\|\mathbf{z}\right\|_{\mathbf{L}_G} ,$$

where $G \stackrel{\text{def}}{=} \text{und}(G)$ and for each $i \in [I]$, $C_i \stackrel{\text{def}}{=} \text{und}(C_i)$.

Proof. This claim follows by an analogous argument as in the proof of Corollary 35, where we use the sketching error claim (31) from Lemma 53 summed across each subgraph. □

We are now ready to give the main algorithm of this section, as well as its analysis. To clarify the role of the expander decomposition (and degree-based spectral bounds), we note that the sketching guarantee provided to a fixed pair of vectors in (40) scales as β^{-1}, as opposed to the $\beta^{-\frac{1}{2}}$ bound one would naïvely apply from our ER decomposition-based guarantee in Lemma 37.

Lemma 55. *There is a universal constant C_{ESS} such that if $C_{\text{ESS}} \cdot \beta^{-1}\phi^{-2}\log(\frac{nW}{\delta\varepsilon})\log^2\log(\frac{nW}{\varepsilon}) \le 1$, and $\vec{G} = (V, E, \mathbf{w})$ is a bipartite lift of a directed graph,* ExpanderSpectralSketch *(Algorithm 8) returns $\vec{G}' = (V, E, \mathbf{w}')$ satisfying the following guarantees with probability $\ge 1 - \delta$.*

Algorithm 8: ExpanderSpectralSketch$(\{\vec{G}_i\}_{i\in[I]}, \vec{G}, T, \delta, \varepsilon, W, \beta)$

Input: $\{\vec{G}^{(i)}\}_{i\in[I]}$, subgraphs of simple $\vec{G} = (V, E, \mathbf{w})$, with $V = A \cup B$, $E \subseteq A \times B$, $\max_{e\in\mathrm{supp}(\mathbf{w})} \mathbf{w}_e \le W$, and such that $\{G^{(i)} \stackrel{\text{def}}{=} \mathrm{und}(\vec{G}^{(i)})\}_{i\in[I]}$ are a $(\phi, 2, J)$-expander decomposition of $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$, T a tree subgraph of G with $\min_{e\in E(T)} \mathbf{w}_e \ge 1$, $\delta, \varepsilon \in (0, \frac{1}{100})$, $\beta > 0$

$m \leftarrow E(\vec{G})$, $n \leftarrow V(\vec{G})$, $R \leftarrow \emptyset$

for $i \in [I]$ **do**

  $V_i \subseteq V(\vec{G}^{(i)})$ be vertices in $\vec{G}^{(i)}$ with combinatorial (unweighted) degrees at least β

  $\vec{H} \leftarrow \vec{C} \stackrel{\text{def}}{=} \vec{G}^{(i)}[V_i]$, $\hat{m} \leftarrow |E(\vec{H})|$, $\hat{n} \leftarrow |V(\vec{H})|$, $\mathbf{w}_\star \leftarrow \mathbf{w}$

  if $\hat{m} \ge 40\hat{n}$ **then**

    $\mathbf{w}_0 \leftarrow \mathbf{w}$, $\vec{C}_0 \leftarrow \vec{C}$, $\vec{H}_0 \leftarrow \vec{H}$, $\ell_1 \leftarrow \frac{1}{2\log^2(\frac{nW}{\varepsilon})}$, $\tau_1 \leftarrow \log(\frac{2}{\ell_1})$

    for $0 \le t < \tau_1$ **do**

      $\mathbf{w}_{t+1} \leftarrow \textsc{BasicFastSparsify}(\vec{H}_t, \vec{C}_t, \mathbf{w}_\star, \ell_1, \frac{\delta}{2I\tau_1}, \frac{\varepsilon}{4I\tau_1}, E(\vec{H}), T)$

      $\vec{C}_{t+1} \leftarrow (V(\vec{C}), E(\vec{C}), [\mathbf{w}_{t+1}]_{E(\vec{C})})$, $\vec{H}_{t+1} \leftarrow (V(\vec{H}), E(\vec{H}), [\mathbf{w}_{t+1}]_{E(\vec{H})})$

    $F \leftarrow \{e \in E(\vec{H}) \mid [\mathbf{w}_t]_e \le \ell_1[\mathbf{w}_\star]_e\}$

    $\mathbf{w}_0 \leftarrow \mathbf{w}_t$, $t \leftarrow 0$, $\vec{G}_0 \leftarrow \vec{G}_t$, $\vec{H}_0 \leftarrow \vec{H}$, $\ell_2 \leftarrow \frac{\varepsilon}{4nmW}$, $\tau_2 \leftarrow \log(\frac{2}{\ell_2})$

    for $0 \le t < \tau_2$ **do**

      $\mathbf{w}_{t+1} \leftarrow \textsc{BasicFastSparsify}(\vec{H}_t, \vec{C}_t, \mathbf{w}_\star, \ell_2, \frac{\delta}{2I\tau_2}, \frac{\varepsilon}{4I\tau_2}, F, T)$

      $\vec{C}_{t+1} \leftarrow (V(\vec{C}), E(\vec{C}), [\mathbf{w}_{t+1}]_{E(\vec{C})})$, $\vec{H}_{t+1} \leftarrow (V(\vec{H}), E(\vec{H}), [\mathbf{w}_{t+1}]_{E(\vec{H})})$;

    $R \leftarrow R \cup \{e \in E(\vec{H}) \mid [\mathbf{w}_t]_e \le \frac{\varepsilon}{4nm}\}$, $\mathbf{w} \leftarrow \mathbf{w}_t$

return $\vec{G}' \leftarrow (V, E, \mathbf{w}_{E\setminus R} + \textsc{Rounding}(\vec{G}, \mathbf{w}_R, T))$

1. $\mathbf{B}_{\vec{G}}^\top \mathbf{w}' = \mathbf{B}_{\vec{G}}^\top \mathbf{w}$, $|\mathbf{B}_{\vec{G}}|^\top \mathbf{w}' = |\mathbf{B}_{\vec{G}}|^\top \mathbf{w}$.

2. $\mathrm{nnz}(\mathbf{w}') \le \frac{31}{32}\mathrm{nnz}(\mathbf{w}) + C_{\mathrm{ESS}} \cdot nJ\beta$.

3. *For* $G \stackrel{\text{def}}{=} \mathrm{und}(\vec{G})$,

$$\left\| \mathbf{L}_G^{\frac{\dagger}{2}} \mathbf{B}_{\vec{G}}^\top \left(\mathbf{W}' - \mathbf{W}\right) \mathbf{H}_{\vec{G}} \mathbf{L}_G^{\frac{\dagger}{2}} \right\|_{\mathrm{op}} \le C_{\mathrm{ESS}} \beta^{-\frac{1}{2}} \phi^{-1} \sqrt{\log\left(\frac{nW}{\delta\varepsilon}\right)} \log\log\left(\frac{nW}{\varepsilon}\right) + \varepsilon.$$

4. *For any fixed* $\mathbf{a}, \mathbf{z} \in \mathbb{R}^V$,

$$\left| \mathbf{a}^\top \mathbf{B}_{\vec{G}}^\top \left(\mathbf{W}' - \mathbf{W}\right) \mathbf{H}_{\vec{G}} \mathbf{z} \right| \le \left(C_{\mathrm{ESS}} \beta^{-1} \phi^{-2} \sqrt{\log\left(\frac{nW}{\delta\varepsilon}\right)} \log\log\left(\frac{nW}{\varepsilon}\right) + \varepsilon \right) \cdot \|\mathbf{a}\|_{\mathbf{L}_G} \|\mathbf{z}\|_{\mathbf{L}_G}. \tag{40}$$

Moreover, $\max_{e\in E} \frac{\mathbf{w}'_e}{\mathbf{w}_e} \le C_{\mathrm{ESS}}$. *The runtime of* ExpanderSpectralSketch *is*

$$\breve{O}\left(|E| \log^2\left(\frac{nW}{\delta\varepsilon}\right) \log\left(\frac{nW}{\varepsilon}\right) \right).$$

Proof. We closely follow the arguments in the proof of Lemma 37. In light of Lemma 52, we let $\rho \stackrel{\text{def}}{=} 8\beta^{-1}\phi^{-2}$. Further, throughout the proof, we condition on the success of all calls to BasicFastSparsify, assuming their input conditions are met, which gives the failure probability. We claim that every $\vec{G}_t$ satisfies $0.9\mathbf{L}_{G_t} \preccurlyeq \mathbf{L}_G \preccurlyeq 1.1\mathbf{L}_{G_t}$, where $G \stackrel{\text{def}}{=} \text{und}(G)$ and $G_t \stackrel{\text{def}}{=} \text{und}(\vec{G}_t)$. Again, we defer proving this statement to the end of the proof.

For a fixed $i \in [I]$, consider the first τ_1 loops from line 8 to line 10. Since we claimed $0.9\mathbf{L}_{G_t} \preccurlyeq \mathbf{L}_G \preccurlyeq 1.1\mathbf{L}_{G_t}$ for all t, Lemma 52 gives that each $\vec{G}^{(i)}[V_i]$ is an ER decomposition piece with parameters $(1.2\rho, 2)$. Then, in all calls to BasicFastSparsify, the conditions on $\mathbf{w}_\star$ are met by assumption. Moreover, BasicFastSparsify is only called if $\hat{m} \geq 40\hat{n}$, and the conditions in Equation (18) are preserved inductively by Lemma 34, since the ℓ_1 norm of the weights does not change by more than a $\frac{\varepsilon}{4\tau_1}$ factor in each iteration. Thus, the τ_1 loops all satisfy their input conditions and we may assume they succeed. We then show that F on Line 11 must have $|F| \geq \frac{\hat{m}}{4}$. Suppose for contradiction that $|F| < \frac{\hat{m}}{4}$, which means the second part of Item 3 in Lemma 34 holds for all iterations $0 \leq t < \tau_1$. However, since Lemma 34 also guarantees

$$\sum_{e \in E(\vec{H})} \log\left(\frac{[\mathbf{w}_\tau]_e}{\mathbf{w}_e}\right) > -\hat{m}\log\left(\frac{2}{\ell}\right) = -\hat{m}\tau_1,$$

we arrive at a contradiction after τ_1 iterations. By using a similar argument, we also show that after τ_2 loops from line 13 to line 15 have finished, at least $\frac{\hat{m}}{16}$ edges are added to R. Notice that for each $i \in [I]$, at most $n_i\beta$ edges are not included in $E(\vec{H}^{(i)})$, we then have the total number of remaining edges (i.e., $\text{nnz}(\mathbf{w}')$) is bounded by

$$\left(m - \sum_{i \in I} \hat{m}_i\right) + \sum_{i \in I}\left(\hat{n}_i\beta + \frac{15}{16}\hat{m}_i\right) \leq m - \frac{1}{16}\cdot\frac{1}{2}m + nJ\beta = \frac{31}{32}m + nJ\beta,$$

where the inequality follows by Item 3 of Definition 48. Similarly, conditioned on all calls to BasicFastSparsify succeeding, by Item 1 of Lemma 34 and the first additional guarantee in Lemma 53, we obtain Item 1.

Now, consider both error bounds in Items 3 and 4. Note that we have $\alpha = 2$ in the first τ_1 calls to BasicFastSparsify for each cluster and $\alpha = \frac{1}{\log^2(\frac{nW}{\varepsilon})}$ in the last τ_2 calls. Since $I \leq m$, we have by Corollaries 35 and 54 and our decomposition parameters that the total spectral error in all intermediate iterations across all decomposition pieces is bounded by

$$O\left(\tau_1 \cdot \sqrt{\rho\log\left(\frac{\tau_1}{\delta}\right)} + \tau_2 \cdot \sqrt{\rho\frac{\log\left(\frac{\tau_1}{\delta}\right)}{\log^2\left(\frac{nW}{\varepsilon}\right)}}\right) = O\left(\beta^{-\frac{1}{2}}\phi^{-1}\right) \cdot \sqrt{\log\left(\frac{mW}{\delta\varepsilon}\right)} \cdot \log\log\left(\frac{mW}{\varepsilon}\right), \quad (41)$$

where we used $\rho = 8\beta^{-1}\phi^{-2}$. Additionally, there is an $\frac{\varepsilon}{4\tau_1 I} \cdot \tau_1 I + \frac{\varepsilon}{4\tau_2 I} \cdot \tau_2 I$ additive spectral error term, which is $\leq \frac{2\varepsilon}{3}$ after accounting for the change in the graph Laplacian by Fact 8. For appropriate C_{ESS}, this both proves the desired spectral error bound by the triangle inequality, as well as the claimed $0.9\mathbf{L}_{G_t} \preccurlyeq \mathbf{L}_G \preccurlyeq 1.1\mathbf{L}_{G_t}$ throughout the algorithm by Fact 7.

Consider now the sketching error bound (40). For each cluster $\vec{G}^{(i)}$ with $G^{(i)} \stackrel{\text{def}}{=} \text{und}(\vec{G}^{(i)})$, let $\mathbf{L}_i, \mathbf{D}_i$ and $\widehat{\mathbf{L}}_i, \widehat{\mathbf{D}}_i$ be the corresponding undirected Laplacians and weighted degrees of $G^{(i)}$ and $\text{und}(\vec{G}^{(i)}[V_i])$ respectively. Following the notation of Corollary 54, and using that $\|\mathbf{x}\|_{\mathbf{\Pi}_{V(\vec{C}_i)}} \leq$

$\|[\mathbf{x} - \hat{x}\mathbb{1}]_{V(\vec{C}_i)}\|_2$ for any $\hat{x} \in \mathbb{R}$, we have that if all calls to BasicFastSparsify succeed,

$$\left|\mathbf{a}^\top \mathbf{B}_{\vec{G}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{T}_{\vec{G}} \mathbf{z}\right| \leq O\left(\sqrt{\log\left(\frac{mW}{\delta\varepsilon}\right)} \cdot \log\log\left(\frac{mW}{\varepsilon}\right)\right) \cdot \sum_{i\in[I]} \bar{w}_i \, \|[\mathbf{a} - \widehat{a}_i \mathbf{1}]_{V_i}\|_2 \, \|[\mathbf{z} - \widehat{z}_i \mathbf{1}]_{V_i}\|_2 + \frac{\varepsilon}{4} \|\mathbf{a}\|_{\mathbf{L}_G} \|\mathbf{z}\|_{\mathbf{L}_G},$$

for any choices of scalars $\{\widehat{a}_i\}_{i\in[I]}$, $\{\widehat{z}_i\}_{i\in[I]}$. In the above, we used a calculation analogous to (41) to bound the first term on the right-hand side. Now, we choose each $\widehat{a}_i$ and $\widehat{z}_i$ as defined in Lemma 51, so that for all $i \in [I]$,

$$\begin{aligned} \bar{w}_i \, \|[\mathbf{a} - \widehat{a}_i \mathbf{1}]_{V_i}\|_2^2 &\leq \bar{w}_i \beta^{-1} \sum_{v\in V_i} [\mathbf{deg}_{G^{(i)}}]_v (\mathbf{a}_v - \widehat{a}_i)^2 \leq 2\beta^{-1}\phi^{-2} \mathbf{a}^\top \mathbf{L}_i \mathbf{a}, \\ \bar{w}_i \, \|[\mathbf{z} - \widehat{z}_i \mathbf{1}]_{V_i}\|_2^2 &\leq \bar{w}_i \beta^{-1} \sum_{v\in V_i} [\mathbf{deg}_{G^{(i)}}]_v (\mathbf{z}_v - \widehat{z}_i)^2 \leq 2\beta^{-1}\phi^{-2} \mathbf{z}^\top \mathbf{L}_i \mathbf{z}. \end{aligned} \tag{42}$$

By the Cauchy-Schwarz inequality and (42), we then obtain the first term in (40):

$$\begin{aligned} \sum_{i\in[I]} \bar{w}_i \, \|[\mathbf{a} - \widehat{a}_i \mathbf{1}]_{V_i}\|_2 \, \|[\mathbf{z} - \widehat{z}_i \mathbf{1}]_{V_i}\|_2 &\leq \sqrt{\sum_{i\in[I]} \bar{w}_i \, \|[\mathbf{a} - \widehat{a}_i \mathbf{1}]_{V_i}\|_2^2} \cdot \sqrt{\sum_{i\in[I]} \bar{w}_i \, \|[\mathbf{z} - \widehat{z}_i \mathbf{1}]_{V_i}\|_2^2} \\ &\leq 2\beta^{-1}\phi^{-2} \cdot \sqrt{\mathbf{a}^\top \left(\sum_{i\in[I]} \mathbf{L}_i\right) \mathbf{a}} \cdot \sqrt{\mathbf{z}^\top \left(\sum_{i\in[I]} \mathbf{L}_i\right) \mathbf{z}} \\ &\leq 2\beta^{-1}\phi^{-2} \cdot \|\mathbf{a}\|_{\mathbf{L}_G} \|\mathbf{z}\|_{\mathbf{L}_G}. \end{aligned}$$

The second term in (40) comes from our earlier application of Corollary 54.

Finally, Rounding incurs at most $\frac{\varepsilon}{4}$ spectral error through the final graph by Lemma 24, which is at most $\frac{\varepsilon}{3}$ spectral error through the original graph by Fact 8. Here we again used that G is connected (Section 2), which implies each G_t is connected via our earlier bound $0.9\mathbf{L}_{G_t} \preceq \mathbf{L}_{G_t} \preceq 1.1\mathbf{L}_{G_t}$. Using an analogous argument from above, this also gives an additive sketching error of at most $\frac{\varepsilon}{3} \cdot \|\mathbf{a}\|_{\mathbf{L}_G} \|\mathbf{z}\|_{\mathbf{L}_G}$. By the first claim in Lemma 24, Item 1 remains true. Item 2 in the lemma statement is clear as we only modify weights within clusters, and Item 2 of Lemma 34 shows no edge weight grows by more than a factor of 60. The runtime follows by applying Corollary 54 to each of the $\tau_1 + \tau_2 = O(\log \frac{nW}{\varepsilon})$ times we call BasicFastSparsify on each expander. □

9.3 Complete spectral sketching algorithm

We are now ready to give our main guarantee on improved constructions of graphical spectral sketches (Definition 44), as well as their Eulerian generalization (Definition 45).

Theorem 56. *Given Eulerian $\vec{G} = (V, E, \mathbf{w})$ with $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, U]^E$ and $\varepsilon, \delta \in (0, \frac{1}{100})$,* SpectralSketch *(Algorithm 9) returns a distribution over $\vec{H}$ that is an (ε, δ)-Eulerian graphical sketch. Moreover, with probability $\geq 1 - \delta$, $\vec{H}$ is a $\sqrt{\varepsilon}$-approximate Eulerian sparsifier of $\vec{G}$, and*

$$|E(\vec{H})| = O\left(\frac{n}{\varepsilon} \log^7(n) \log^2\left(\frac{nU}{\delta}\right) \log^2 \log\left(\frac{nU}{\delta}\right)\right),$$

$$\log\left(\frac{\max_{e\in\mathrm{supp}(\mathbf{w}')} \mathbf{w}'_e}{\min_{e\in\mathrm{supp}(\mathbf{w}')} \mathbf{w}_e}\right) = O(\log(nU)).$$

Algorithm 9: SpectralSketch$(\vec{G}, \varepsilon, \delta)$

1 **Input:** $\vec{G} = (V, E, \mathbf{w})$ with $\mathbf{w}_e \in [1, U]$ for all $e \in E$, $\varepsilon, \delta \in (0, \frac{1}{100})$
2 $n \leftarrow |V|$, $m \leftarrow |E|$
3 $\vec{G}^{\uparrow} \stackrel{\text{def}}{=} (V \cup V', E^{\uparrow}, \mathbf{w}) \leftarrow \text{blift}(\vec{G})$
4 $T \leftarrow$ arbitrary spanning tree of $G^{\uparrow} \stackrel{\text{def}}{=} \text{und}(\text{blift}(\vec{G}))$, $\hat{E}^{\uparrow} \leftarrow E^{\uparrow} \setminus E(T)$
5 $R \leftarrow 6 \log n$, $U_{\max} \leftarrow U \cdot C_{\text{ESS}}^{R}$ for C_{ESS} in Lemma 55
6 $\beta \leftarrow \frac{400000 C_{\text{ESS}}^2}{C_{\text{ADK}}^2 \cdot \varepsilon} \cdot \log^6 n \log\left(\frac{nU_{\max}}{\delta}\right) \log^2 \log\left(\frac{nU_{\max}}{\delta}\right)$, $t \leftarrow 0$, $\mathbf{w}_0 \leftarrow \mathbf{w}$
7 **while** $t < R$ **and** $\text{nnz}([\mathbf{w}_t]_{\hat{E}^{\uparrow}}) > 4 C_{\text{ESS}} n \beta \log_2(\frac{32 m n U_{\max}}{\varepsilon})$ **do**
8 $\quad \vec{G}_t^{\uparrow} \leftarrow (V \cup V', E^{\uparrow}, \mathbf{w}_t)$, $G_t^{\uparrow} \leftarrow \text{und}(\vec{G}_t^{\uparrow})$
9 $\quad S \leftarrow$ ExpanderDecompADK$([G_t^{\uparrow}]_{\hat{E}^{\uparrow}}, 2, \frac{\delta}{4R})$ ▷ See Proposition 49.
10 $\quad (\vec{G}_t^{\uparrow})' \stackrel{\text{def}}{=} (V \cup V', E^{\uparrow}, \mathbf{w}_t') \leftarrow$ ExpanderSpectralSketch$(S, \vec{G}_t^{\uparrow}, T, \frac{\delta}{4R}, \frac{\varepsilon}{4R}, U_{\max}, \beta)$
11 $\quad D \leftarrow \{e \in E^{\uparrow} \mid [\mathbf{w}_t']_e \le \frac{\varepsilon}{4mn}\}$
12 $\quad \mathbf{w}_{t+1} \leftarrow [\mathbf{w}_t']_{E^{\uparrow} \setminus D} +$ Rounding$(\vec{G}^{\uparrow}, [\mathbf{w}_t']_D, T)$
13 $\quad t \leftarrow t + 1$
14 **return** $\vec{H} \leftarrow (V, E, \mathbf{w}_t)$

The runtime of SpectralSketch *is*

$$\breve{O}\left(m \log^2\left(\frac{nU}{\delta}\right) \log(nU) + m \log^8(n) \log\left(\frac{n}{\delta}\right)\right).$$

Proof. Throughout, we condition on the event that all of the R calls to ExpanderDecompADK and ExpanderSpectralSketch succeed, which happens with probability $\ge 1 - \delta$. Since ExpanderSpectralSketch guarantees no weight grows by more than a C_{ESS} factor in each call, $U_{\max}$ is an upper bound for the maximum possible weight throughout the algorithm. As we remove any edge with weight below $\frac{\varepsilon}{4mn}$ on line 12, the number of expander pieces is upper bounded by $J_{\max} \stackrel{\text{def}}{=} \log_2(\frac{32 m n U_{\max}}{\varepsilon})$ by Proposition 49.

When $\text{nnz}(\mathbf{w}_t) \ge 4 C_{\text{ESS}} \cdot n \beta J_{\max}$, Item 3 of Definition 48 and Item 2 of Lemma 55 guarantees that the number of edges in each iteration decreases by at least a $\frac{1}{64}$ factor. Since $m \le n^2$, we may assume for the rest of the proof that $\varepsilon > \frac{1}{n^2}$. Therefore, after R iterations, we are guaranteed that the number of edges at termination is at most $O(n \beta J_{\max})$. Plugging in the definition of β and $\phi = C_{\text{ADK}} \log^{-2}(m) \ge \frac{1}{4} C_{\text{ADK}} \log^{-2}(n)$ and noting that $\log(U_{\max}) = O(\log(nU))$ gives the desired sparsity bound. The runtime follows from combining Proposition 49, Lemma 55 and noting that the number of edges decreases geometrically until the lower bound on line 7.

The degree-preserving property follows from Item 1 of Lemma 55 and the first claim of Lemma 24. This guarantees that if $\vec{G}$ is Eulerian, then $\vec{H}$ is also Eulerian.

Next, consider the spectral error bound. By Item 3 of Lemma 55, the total spectral error incurred within each iteration of the while loop from line 7 to line 13 with respect to the current Laplacian is bounded by

$$5 C_{\text{ESS}} \cdot \beta^{-1/2} \phi^{-1} \sqrt{\log\left(\frac{nU_{\max}}{\delta}\right) \log\log(nU_{\max}) + \frac{\varepsilon}{4R}} \le \frac{1}{4}\frac{\sqrt{\varepsilon}}{R} + \frac{1}{4}\frac{\varepsilon}{R} \le \frac{1}{2}\frac{\sqrt{\varepsilon}}{R}.$$

We condition on $0.9 \mathbf{L}_G \preccurlyeq \mathbf{L}_{G_t} \preccurlyeq 1.1 \mathbf{L}_G$ for all $t < R$, which shows a total spectral error of at most $\frac{2}{3}\sqrt{\varepsilon}$ over all iterations due to Fact 7. Similarly, the rounding on line 12 also contributes at most $\frac{\varepsilon}{4}$

by Lemma 24 and Fact 7. This also shows our assumption $0.9\mathbf{L}_G \preccurlyeq \mathbf{L}_{G_t} \preccurlyeq 1.1\mathbf{L}_G$ holds, as $\varepsilon < \frac{1}{100}$. As before, we achieve the sparsity bound by dropping edges with zero weight in $\mathbf{w}_t$.

Finally, consider the sketching error bound. We take the same definition of $\mathbf{Q}$ as in Lemma 46. Let $\mathbf{a}, \mathbf{z} \in \mathbb{R}^V$ be arbitrary fixed vectors. We have, by Item 4 of Lemma 55, the sketching error for $\mathbf{Qa}$ and $\mathbf{Qz}$ in $\vec{G}^{\uparrow}$ is bounded by $\|\mathbf{Qa}\|_{\mathbf{L}_{G^{\uparrow}}} \|\mathbf{Qz}\|_{\mathbf{L}_{G^{\uparrow}}}$ times

$$\frac{4}{3} \cdot R \cdot \left(5C_{\text{ESS}} \cdot \beta^{-1}\phi^{-2} \sqrt{\log\left(\frac{nU_{\max}}{\delta}\right)} \log\log(nU_{\max}) + \frac{\varepsilon}{4R} \right) \leq \frac{2}{3}\varepsilon,$$

where the factor of $\frac{4}{3}$ again comes from the valid assumption of $0.9\mathbf{L}_G \preccurlyeq \mathbf{L}_{G_t} \preccurlyeq 1.1\mathbf{L}_G$ and Fact 7 for each factor of the form $\|\mathbf{x}\|_{\mathbf{L}_{G_t}}$. By an analogous argument in the proof of Lemma 55, the additive error by ROUNDING on line 12 is bounded by $\frac{4}{3} \cdot \frac{\varepsilon}{4} \|\mathbf{Qa}\|_{\mathbf{L}_{G^{\uparrow}}} \|\mathbf{Qz}\|_{\mathbf{L}_{G^{\uparrow}}}$. Now, the fact that (29) implies (30) gives the desired sketching error bound in the unlifted graph. □

Our spectral sketch algorithm has additional desirable properties in the undirected graph setting, where we can ensure that the sketched graph is also undirected via the following reduction.

Lemma 57. *For an undirected graph $G = (V, E, \mathbf{w})$, let $\vec{H} = (V, E', \mathbf{w})$ be a directed graph where each edge $e' \in E'$ has the same endpoints as an undirected edge $e \in E$ with an arbitrary orientation. Let $\mathbf{w}' \in \mathbb{R}^E$ satisfy $\mathbf{B}_{\vec{H}}^\top \mathbf{w}' = \mathbf{B}_{\vec{H}}^\top \mathbf{w}$. Then for any $\mathbf{x} \in \mathbb{R}^V$,*

$$\left\| \mathbf{L}_G^{\frac{\dagger}{2}} \mathbf{B}_G^\top (\mathbf{W}' - \mathbf{W}) \mathbf{B}_G \mathbf{L}_G^{\frac{\dagger}{2}} \right\|_{\text{op}} \leq 4 \left\| \mathbf{L}_H^{\frac{\dagger}{2}} \mathbf{B}_{\vec{H}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{H}} \mathbf{L}_H^{\frac{\dagger}{2}} \right\|_{\text{op}},$$

$$\left| \mathbf{x}^\top \mathbf{L}_G^{\frac{\dagger}{2}} \mathbf{B}_G^\top (\mathbf{W}' - \mathbf{W}) \mathbf{B}_G \mathbf{L}_G^{\frac{\dagger}{2}} \mathbf{x} \right| \leq 4 \left| \mathbf{x}^\top \mathbf{L}_H^{\frac{\dagger}{2}} \mathbf{B}_{\vec{H}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{H}} \mathbf{L}_H^{\frac{\dagger}{2}} \mathbf{x} \right|,$$

where $H \stackrel{\text{def}}{=} \text{und}(\vec{H}) = (V, E, 2\mathbf{w})$.

Proof. Without loss of generality, we assume that orientations are chosen so that $\mathbf{B}_G \stackrel{\text{def}}{=} \mathbf{B}_{\vec{H}}$. Since $\mathbf{w}' - \mathbf{w}$ is a circulation, we have by Fact 6, $\mathbf{B}_{\vec{H}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{H}} = -\mathbf{T}_{\vec{H}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{B}_{\vec{H}}$. Further, as

$$\mathbf{B}_G^\top (\mathbf{W}' - \mathbf{W}) \mathbf{B}_G = \mathbf{B}_{\vec{H}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{H}_{\vec{H}} - \mathbf{B}_{\vec{H}}^\top (\mathbf{W}' - \mathbf{W}) \mathbf{T}_{\vec{H}},$$

applying the triangle inequalities on operator norms and absolute values, combined with the fact $\mathbf{L}_H = 2\mathbf{L}_G$, gives both desired inequalities. □

Moreover, we can use the following claim from [CGPSSW18] to show that the output of our algorithm in the undirected case is an approximate *inverse sketch* of G, i.e., it preserves quadratic forms with the Laplacian pseudoinverse. This is useful for approximating effective resistances.

Lemma 58 (Lemma 6.8, [CGPSSW18])**.** *Let $\mathbf{M}, \mathbf{N}$ be symmetric PSD matrices of the same dimension, and let $\mathbf{x}$ be a vector of the same dimension such that for some $\varepsilon \in (0, 0.1)$,*

$$\left\| \mathbf{M}^{\frac{\dagger}{2}} (\mathbf{M} - \mathbf{N}) \mathbf{M}^{\frac{\dagger}{2}} \right\|_{\text{op}} \leq \sqrt{\varepsilon}, \quad |\mathbf{x}^\top \mathbf{M}^\dagger (\mathbf{M} - \mathbf{N}) \mathbf{M}^\dagger \mathbf{x}| \leq \varepsilon \cdot \mathbf{x}^\top \mathbf{M}^\dagger \mathbf{x}.$$

Then,

$$|\mathbf{x}^\top (\mathbf{M}^\dagger - \mathbf{N}^\dagger) \mathbf{x}| \leq 7\varepsilon \cdot \mathbf{x}^\top \mathbf{M}^\dagger \mathbf{x}.$$[8]

[8] The original Lemma 6.8 in [CGPSSW18] uses a different, more symmetric definition of ε-approximation, involving multiplicative factors of e^{ε}, but it is straightforward to check that the same constant factors hold for our definition.

The following theorem is a refined version of Theorem 5. To obtain these results, we crucially use the fact that the guarantees of Lemma 55 continue to hold even if the input directed graph is not Eulerian (as the signed variant of an undirected graph, as in Lemma 57, need not be Eulerian).

Corollary 59. *There is an algorithm that, given undirected graph $G = (V, E, \mathbf{w})$ with $|V| = n$, $|E| = m$, $\mathbf{w} \in [1, U]^E$ and $\varepsilon, \delta \in (0, \frac{1}{100})$, returns a distribution over graphs H which is an (ε, δ)-graphical spectral sketch, and*

$$|E(H)| = O\left(\frac{n}{\varepsilon}\log^7(n)\log^2\left(\frac{nU}{\delta}\right)\log^2\log\left(\frac{nU}{\delta}\right)\right),$$
$$\log\left(\frac{\max_{e\in\mathrm{supp}(\mathbf{w}')}\mathbf{w}'_e}{\min_{e\in\mathrm{supp}(\mathbf{w}')}\mathbf{w}_e}\right) = O(\log(nU)).$$

Moreover, with probability $\geq 1-\delta$, H is a $\sqrt{\varepsilon}$-approximate spectral sparsifier of G, and for an arbitrary fixed $\mathbf{x} \in \mathbb{R}^V$, $|\mathbf{x}^\top(\mathbf{L}_H^\dagger - \mathbf{L}_G^\dagger)\mathbf{x}| \leq \varepsilon \cdot \mathbf{x}^\top\mathbf{L}_H^\dagger\mathbf{x}$. The runtime of the algorithm is $\breve{O}\left(m\log^2\left(\frac{nU}{\delta}\right)\log(nU) + m\log^8(n)\log(\frac{n}{\delta})\right)$.

Proof. This is a direct consequence of Lemmas 57 and 58 and Theorem 56. Here, instead of the standard transformation of doubling the edges and taking both directions of each edge for G, we keep one edge each and set an arbitrary direction as in Lemma 57. We remark that the input directed graph, say $\vec{G} = (V, E', \mathbf{w})$, to SpectralSketch need not be Eulerian. Let $\vec{G}' = (V, E', \mathbf{w}')$ be the resulting directed graph, then $\mathbf{B}_{\vec{G}}^\top\mathbf{w}' = \mathbf{B}_{\vec{H}}^\top\mathbf{w}$ by Item 1 of Lemma 55 and the first claim of Lemma 24. Scaling ε by a factor of $\frac{1}{30}$ then guarantees our desired approximation factors. □

Acknowledgments

We thank Richard Peng [Pen23] for clarifying the dependence in Proposition 39 on δ and U. We thank the authors of [CKPPRSV17] for helpful discussions. Sushant Sachdeva's research is supported by an Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant RGPIN-2018-06398, an Ontario Early Researcher Award (ERA) ER21-16-283, and a Sloan Research Fellowship. Aaron Sidford was supported in part by a Microsoft Research Faculty Fellowship, NSF CAREER Grant CCF-1844855, NSF Grant CCF-1955039, and a PayPal research award. Part of this work was conducted while authors were visiting the Simons Institute for the Theory of Computing Fall 2023 Program *Data Structures and Optimization for Fast Algorithms*. The authors are grateful to the Simons Institute for its support.

References

[AALG18] Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. "Graph Clustering using Effective Resistance". In: *9th Innovations in Theoretical Computer Science Conference, ITCS 2018*. Vol. 94. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 41:1–41:16 (cit. on pp. 12, 25, 59, 60).

[ACKQWZ16] 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, 2016*. ACM, 2016, pp. 311–319 (cit. on pp. 3, 13, 40).

[ADK23] Daniel Agassy, Dani Dorfman, and Haim Kaplan. "Expander Decomposition with Fewer Inter-Cluster Edges Using a Spectral Cut Player". In: *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*. Vol. 261. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 9:1–9:20 (cit. on pp. 2, 41, 43).

[AJSS19] AmirMahdi Ahmadinejad, Arun Jambulapati, Amin Saberi, and Aaron Sidford. "Perron-Frobenius Theory in Nearly Linear Time: Positive Eigenvectors, M-matrices, Graph Kernels, and Other Applications". In: *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2019, pp. 1387–1404 (cit. on pp. 1, 3, 6, 7, 39).

[AKMPSV20] AmirMahdi Ahmadinejad, Jonathan A. Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil P. Vadhan. "High-precision Estimation of Random Walks in Small Space". In: *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*. IEEE, 2020, pp. 1295–1306 (cit. on p. 1).

[AN19] Ittai Abraham and Ofer Neiman. "Using Petal-Decompositions to Build a Low Stretch Spanning Tree". In: *SIAM J. Comput.* 48.2 (2019), pp. 227–248 (cit. on p. 13).

[APPSV23] AmirMahdi Ahmadinejad, John Peebles, Edward Pyne, Aaron Sidford, and Salil P. Vadhan. "Singular Value Approximation and Sparsifying Random Walks on Directed Graphs". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*. IEEE, 2023, pp. 846–854 (cit. on pp. 2, 4, 5, 7).

[BBH23] Afonso S Bandeira, March T Boedihardjo, and Ramon van Handel. "Matrix concentration inequalities and free probability". In: *Inventiones mathematicae* (2023), pp. 1–69 (cit. on p. 61).

[BJM23] Nikhil Bansal, Haotian Jiang, and Raghu Meka. "Resolving Matrix Spencer Conjecture Up to Poly-logarithmic Rank". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*. ACM, 2023, pp. 1814–1819 (cit. on pp. 3, 6, 12, 19, 20, 23, 59–61).

[BK96] András A. Benczúr and David R. Karger. "Approximating S-t Minimum Cuts in $\tilde{O}$(N2) Time". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Association for Computing Machinery, 1996, pp. 47–55 (cit. on p. 1).

[BS03] Surender Baswana and Sandeep Sen. "A Simple Linear Time Algorithm for Computing a (2k-1)-Spanner of O($n^{1+1/k}$) Size in Weighted Graphs". In: *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003*. Vol. 2719. Lecture Notes in Computer Science. Springer, 2003, pp. 384–296 (cit. on p. 1).

[BSS12] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. "Twice-Ramanujan Sparsifiers". In: *SIAM J. Comput.* 41.6 (2012), pp. 1704–1721 (cit. on pp. 1, 3, 6).

[CGPSSW18] 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: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018, pp. 361–372 (cit. on pp. 1–6, 10, 13, 40, 41, 43, 51).

[CKKPPRS18] 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: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2018, pp. 898–909 (cit. on p. 6).

[CKMPPRX14] 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 log1/2 n time". In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 2014, pp. 343–352 (cit. on p. 6).

[CKPPRSV17] 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 SIGACT Symposium on Theory of Computing*. 2017, pp. 410–419 (cit. on pp. 1–6, 39, 52).

[CKPPSV16] 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: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2016, pp. 583–592 (cit. on pp. 1, 3, 6, 7, 38, 39).

[CKST19] Charles Carlson, Alexandra Kolla, Nikhil Srivastava, and Luca Trevisan. "Optimal Lower Bounds for Sketching Graph Cuts". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*. SIAM, 2019, pp. 2565–2569 (cit. on pp. 1, 3).

[DMSY23] Ran Duan, Jiayi Mao, Xinkai Shu, and Longhui Yin. "A Randomized Algorithm for Single-Source Shortest Path on Undirected Real-Weighted Graphs". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*. IEEE Computer Society, 2023 (cit. on p. 13).

[FGLPSY22] Sebastian Forster, Gramoz Goranci, Yang P Liu, Richard Peng, Xiaorui Sun, and Mingquan Ye. "Minor sparsifiers and the distributed laplacian paradigm". In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 989–999 (cit. on p. 6).

[Gia97] Apostolos A. Giannopoulos. "On some vector balancing problems". In: *Studia Mathematica* 122.3 (1997), pp. 225–234 (cit. on p. 59).

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*. Springer, 1988 (cit. on p. 23).

[GVY96] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. "Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications". In: *SIAM J. Comput.* 25.2 (1996), pp. 235–251 (cit. on pp. 12, 14, 16).

[HRS22] Samuel B. Hopkins, Prasad Raghavendra, and Abhishek Shetty. "Matrix discrepancy from Quantum communication". In: *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2022, pp. 637–648 (cit. on p. 59).

[JRT23] Arun Jambulapati, Victor Reis, and Kevin Tian. "Linear-Sized Sparsifiers via Near-Linear Time Discrepancy Theory". In: *arXiv preprint arXiv:2305.08434* (2023) (cit. on pp. 1, 6, 7, 10).

[JS18] Arun Jambulapati and Aaron Sidford. "Efficient $\widetilde{O}(n/\varepsilon)$ Spectral Sketches for the Laplacian and its Pseudoinverse". In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*. SIAM, 2018, pp. 2487–2503 (cit. on pp. 3, 13, 40, 41).

[JS21] Arun Jambulapati and Aaron Sidford. "Ultrasparse Ultrasparsifiers and Faster Laplacian System Solvers". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*. SIAM, 2021, pp. 540–559 (cit. on pp. 1, 6, 14, 15).

[KLPSS16] 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, pp. 842–850 (cit. on p. 6).

[KLS20] Rasmus Kyng, Kyle Luh, and Zhao Song. "Four deviations suffice for rank 1 matrices". In: *Advances in Mathematics* 375.2 (2020), pp. 557–567 (cit. on p. 59).

[KMP11] Ioannis Koutis, Gary L Miller, and Richard Peng. "A nearly-m log n time solver for sdd linear systems". In: *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. IEEE. 2011, pp. 590–598 (cit. on p. 6).

[KMP14] Ioannis Koutis, Gary L Miller, and Richard Peng. "Approaching optimality for solving SDD linear systems". In: *SIAM Journal on Computing* 43.1 (2014), pp. 337–354 (cit. on p. 6).

[KOSZ13] Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. "A simple, combinatorial algorithm for solving SDD systems in nearly-linear time". In: *Symposium on Theory of Computing Conference, STOC'13*. ACM, 2013, pp. 911–920 (cit. on p. 21).

[KS16] 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)*. IEEE. 2016, pp. 573–582 (cit. on p. 6).

[LS17] Yin Tat Lee and He Sun. "An sdp-based algorithm for linear-sized spectral sparsification". In: *Proceedings of the 49th annual acm sigact symposium on theory of computing*. 2017, pp. 678–687 (cit. on pp. 1, 6).

[LS18] Yin Tat Lee and He Sun. "Constructing linear-sized spectral sparsification in almost-linear time". In: *SIAM Journal on Computing* 47.6 (2018), pp. 2315–2336 (cit. on p. 6).

[LSY19] Yang P Liu, Sushant Sachdeva, and Zejun Yu. "Short cycles via low-diameter decompositions". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2602–2615 (cit. on pp. 2–4, 6).

[LWZ24] Lap Chi Lau, Robert Wang, and Hong Zhou. *Spectral Sparsification by Deterministic Discrepancy Walk*. 2024. arXiv: 2408.06146 [cs.DS] (cit. on p. 7).

[Moh91] Bojan Mohar. "Eigenvalues, diameter, and mean distance in graphs". In: *Graphs and combinatorics* 7.1 (1991), pp. 53–64 (cit. on p. 58).

[MPX13] Gary L. Miller, Richard Peng, and Shen Chen Xu. "Parallel graph decompositions using random shifts". In: *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '13*. ACM, 2013, pp. 196–203 (cit. on p. 13).

[Pen23] Richard Peng. Personal communication. 2023 (cit. on p. 52).

[PS14] Richard Peng and Daniel A Spielman. "An efficient parallel solver for SDD linear systems". In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 2014, pp. 333–342 (cit. on p. 6).

[PS22] Richard Peng and Zhuoqing Song. "Sparsified block elimination for directed laplacians". In: *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2022, pp. 557–567 (cit. on pp. 3, 6, 7, 38).

[PV23] Lucas Pesenti and Adrian Vladu. "Discrepancy Minimization via Regularization". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2023, pp. 1734–1758 (cit. on p. 7).

[PY19] Merav Parter and Eylon Yogev. "Optimal short cycle decomposition in almost linear time". In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2019 (cit. on pp. 2–4, 6).

[RH17] Philippe Rigollet and Jan-Christian Hütter. *High-Dimensional Statistics*. 2017 (cit. on p. 29).

[Rot17] Thomas Rothvoss. "Constructive Discrepancy Minimization for Convex Sets". In: *SIAM J. Comput.* 46.1 (2017), pp. 224–234 (cit. on pp. 5, 7, 12, 19, 59).

[RR20] Victor Reis and Thomas Rothvoss. "Linear Size Sparsifier and the Geometry of the Operator Norm Ball". In: *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*. SIAM, 2020, pp. 2337–2348 (cit. on pp. 7, 10).

[RR23] Victor Reis and Thomas Rothvoss. "Vector balancing in Lebesgue spaces". In: *Random Structures & Algorithms* 62.3 (2023), pp. 667–688 (cit. on pp. 12, 19, 20, 58).

[SS11] Daniel Spielman and Nikhil Srivastava. "Graph Sparsification by Effective Resistances". In: *SIAM Journal on Computing* 40.6 (2011), pp. 1913–1926 (cit. on pp. 1, 6, 14, 15).

[ST04] Daniel A. Spielman and Shang-Hua Teng. "Nearly-Linear Time Algorithms for Graph Partitioning, Graph Sparsification, and Solving Linear Systems". In: *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*. STOC '04. Association for Computing Machinery, 2004, pp. 81–90 (cit. on pp. 1, 2, 6).

[STZ24] Sushant Sachdeva, Anvith Thudi, and Yibin Zhao. "Better Sparsifiers for Directed Eulerian Graphs". In: *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*. Ed. by Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson. Vol. 297. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 119:1–119:20 (cit. on pp. 2, 4, 6, 7).

[SW19] Thatchaphol Saranurak and Di Wang. "Expander Decomposition and Pruning: Faster, Stronger, and Simpler". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*. Ed. by Timothy M. Chan. SIAM, 2019, pp. 2616–2635 (cit. on p. 2).

[SZ23] Sushant Sachdeva and Yibin Zhao. "A Simple and Efficient Parallel Laplacian Solver". In: *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures*. SPAA '23. 2023, pp. 315–325 (cit. on p. 6).

[Tho99] Mikkel Thorup. "Undirected Single-Source Shortest Paths with Positive Integer Weights in Linear Time". In: *J. ACM* 46.3 (1999), pp. 362–394 (cit. on p. 13).

[Tro11] Joel A. Tropp. *User-friendly tail bounds for sums of random matrices*. 2011 (cit. on p. 28).

[Tro18] Joel A. Tropp. "Second-order matrix concentration inequalities". In: *Applied and Computational Harmonic Analysis* 44.3 (2018), pp. 700–736 (cit. on pp. 20, 61).

[TZ05] Mikkel Thorup and Uri Zwick. "Approximate distance oracles". In: *J. ACM* 52.1 (2005), pp. 1–24 (cit. on p. 1).

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018 (cit. on p. 29).

A Deferred proofs from Section 2

Fact 6. *Let* $\mathbf{B} = \mathbf{H} - \mathbf{T}$ *be the edge-vertex incidence matrix of a graph, let* $\mathbf{x}$ *be a circulation in the graph (i.e.* $\mathbf{B}^\top \mathbf{x} = \mathbf{0}$*), and let* $\mathbf{X} \stackrel{\text{def}}{=} \mathbf{diag}\,(\mathbf{x})$*. Then* $\mathbf{H}^\top \mathbf{X} \mathbf{H} = \mathbf{T}^\top \mathbf{X} \mathbf{T}$ *and* $\mathbf{B}^\top \mathbf{X} \mathbf{H} = -\mathbf{T}^\top \mathbf{X} \mathbf{B}$*.*

Proof. We observe that $\mathbf{H}^\top \mathbf{X} \mathbf{H} = \mathbf{diag}\left(\mathbf{H}^\top \mathbf{x}\right)$ and $\mathbf{T}^\top \mathbf{X} \mathbf{T} = \mathbf{diag}\left(\mathbf{T}^\top \mathbf{x}\right)$. The first claim then follows from $\mathbf{H}^\top \mathbf{x} = \mathbf{T}^\top \mathbf{x}$ as $\mathbf{x}$ is a circulation. The second claim then follows from

$$\mathbf{B}^\top \mathbf{X} \mathbf{H} = \mathbf{H}^\top \mathbf{X} \mathbf{H} - \mathbf{T}^\top \mathbf{X} \mathbf{H} = \mathbf{T}^\top \mathbf{X} \mathbf{T} - \mathbf{T}^\top \mathbf{X} \mathbf{H} = -\mathbf{T}^\top \mathbf{X} \mathbf{B}.$$

□

Fact 7. *Suppose* $\vec{G} = (V, E, \mathbf{w}_{\vec{G}}), \vec{H} = (V, F, \mathbf{w}_{\vec{H}})$ *share the same vertex set and* $G \stackrel{\text{def}}{=} \text{und}(\vec{G})$*,* $H \stackrel{\text{def}}{=} \text{und}(\vec{H})$*. If* $\mathbf{B}_{\vec{G}}^\top \mathbf{w}_{\vec{G}} = \mathbf{B}_{\vec{H}}^\top \mathbf{w}_{\vec{H}}$*, then* $\|\mathbf{L}_G^{\dagger\frac{2}{2}}(\mathbf{L}_G - \mathbf{L}_H)\mathbf{L}_G^{\dagger\frac{2}{2}}\|_{\text{op}} \le 2\|\mathbf{L}_G^{\dagger\frac{2}{2}}(\vec{\mathbf{L}}_{\vec{G}} - \vec{\mathbf{L}}_{\vec{H}})\mathbf{L}_G^{\dagger\frac{2}{2}}\|_{\text{op}}$*.*

Proof. Throughout the proof, let

$$\mathbf{B} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{B}_{\vec{G}} \\ \mathbf{B}_{\vec{H}} \end{pmatrix} \in \{0,1\}^{(E \cup F) \times V}, \ \mathbf{w} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{w}_{\vec{G}} \\ -\mathbf{w}_{\vec{H}} \end{pmatrix} \in \mathbb{R}^{E \cup F},$$

and define $\mathbf{H}, \mathbf{T} \in \{0,1\}^{(E \cup F) \times V}$ to be appropriate concatenations such that $\mathbf{B} = \mathbf{H} - \mathbf{T}$. Observe that $\mathbf{B}^\top \mathbf{w} = \mathbf{B}_{\vec{G}}^\top \mathbf{w}_{\vec{G}} - \mathbf{B}_{\vec{H}}^\top \mathbf{w}_{\vec{H}} = \mathbf{0}_V$. By Fact 6, we have

$$\vec{\mathbf{L}}_{\vec{G}} - \vec{\mathbf{L}}_{\vec{H}} = \mathbf{B}^\top \mathbf{W} \mathbf{H}, \quad \vec{\mathbf{L}}_{\text{rev}(\vec{G})} - \vec{\mathbf{L}}_{\text{rev}(\vec{H})} = -\mathbf{B}^\top \mathbf{W} \mathbf{T} = \mathbf{H}^\top \mathbf{W} \mathbf{B} = \vec{\mathbf{L}}_{\vec{G}}^\top - \vec{\mathbf{L}}_{\vec{H}}^\top.$$

It then suffices to apply the triangle inequality, that transposition preserves the operator norm, and the characterization $\mathbf{L}_G = \vec{\mathbf{L}}_{\vec{G}} + \vec{\mathbf{L}}_{\mathrm{rev}(\vec{G})}$ (with a similar equality for $\vec{H}$ and H). □

Fact 8. *Suppose G, H are connected graphs on the same vertex set V, and $\|\mathbf{L}_G^{\dagger/2}(\mathbf{L}_G - \mathbf{L}_H)\mathbf{L}_G^{\dagger/2}\|_{\mathrm{op}} \leq \varepsilon$. Then for any $\mathbf{M} \in \mathbb{R}^{V\times V}$, we have $\|\mathbf{L}_G^{\dagger/2}\mathbf{M}\mathbf{L}_G^{\dagger/2}\|_{\mathrm{op}} \leq (1+\varepsilon)\|\mathbf{L}_H^{\dagger/2}\mathbf{M}\mathbf{L}_H^{\dagger/2}\|_{\mathrm{op}}$.*

Proof. Since $\mathbf{L}_G$ and $\mathbf{L}_H$ share a kernel, the given condition implies $(1-\varepsilon)\mathbf{L}_G \preceq \mathbf{L}_H \preceq (1+\varepsilon)\mathbf{L}_G$. Hence, $\|v\|_{\mathbf{L}_G} \leq 1$ implies $\|v\|_{\mathbf{L}_H} \leq \sqrt{1+\varepsilon}$, and so the conclusion follows from

$$\left\|\mathbf{L}_G^{\frac{\dagger}{2}}\mathbf{M}\mathbf{L}_G^{\frac{\dagger}{2}}\right\|_{\mathrm{op}} = \sup_{\substack{u,v\perp\mathbf{1}_V\\ \|u\|_{\mathbf{L}_G},\|v\|_{\mathbf{L}_G}\leq 1}} u^\top\mathbf{M}v \leq (1+\varepsilon)\sup_{\substack{u,v\perp\mathbf{1}_V\\ \|u\|_{\mathbf{L}_H},\|v\|_{\mathbf{L}_H}\leq 1}} u^\top\mathbf{M}v = (1+\varepsilon)\left\|\mathbf{L}_H^{\frac{\dagger}{2}}\mathbf{M}\mathbf{L}_H^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}.$$

□

B Rounding

In this section, we prove Lemma 24, our guarantee on ROUNDING.

Lemma 24. *Given $\vec{G} = (V, E, \mathbf{w})$, a tree subgraph T of $G \stackrel{\mathrm{def}}{=} \mathrm{und}(\vec{G})$ with $\min_{e\in E(T)} \mathbf{w}_e \geq 1$,* ROUNDING *(Algorithm 1) returns in $O(n)$ time $\mathbf{y} \in \mathbb{R}^E$ with $\mathrm{supp}(\mathbf{y}) \subseteq T$ satisfying:*

1. $\mathbf{B}_{\vec{G}}^\top\mathbf{y} = \mathbf{d}$.
2. $\|\mathbf{y}\|_\infty \leq \frac{1}{2}\|\mathbf{d}\|_1$.
3. *For any $\mathbf{z} \in \mathbb{R}^E$ satisfying $\mathbf{B}_{\vec{G}}^\top\mathbf{z} = \mathbf{d}$, we have $\|\mathbf{L}_G^{\dagger/2}\mathbf{B}_{\vec{G}}^\top(\mathbf{Y}-\mathbf{Z})\mathbf{H}_{\vec{G}}\mathbf{L}_G^{\dagger/2}\|_{\mathrm{op}} \leq n\|\mathbf{z}\|_1$.*
4. $\|\mathbf{L}_G^{\dagger/2}\mathbf{B}_{\vec{G}}^\top\mathbf{Y}\mathbf{H}_{\vec{G}}\mathbf{L}_G^{\dagger/2}\|_{\mathrm{op}} \leq n\|\mathbf{y}\|_1$.

Proof. Throughout the proof we drop the subscripts $\vec{G}$, G from $\mathbf{B}$, $\mathbf{H}$, $\mathbf{L}$ for simplicity. The algorithm sets $\mathbf{y}$ to be the unique flow on the edges of tree T that satisfies $\mathbf{B}^\top\mathbf{y} = \mathbf{d}$. Such a vector $\mathbf{y}$ can be constructed in $O(n)$ time by recursively computing the flow required at each leaf, and then removing the leaf. By construction, $\mathrm{supp}(\mathbf{y}) \subseteq T$. Since $\mathbf{d} \perp \mathbf{1}_V$, we also have $\|\mathbf{y}\|_\infty \leq \frac{1}{2}\|\mathbf{d}\|_1$.

Next, recall $\mathbf{B}^\top\mathbf{z} = \mathbf{d}$, so $\|\mathbf{d}\|_1 = \|\mathbf{B}^\top\mathbf{z}\|_1 \leq 2\|\mathbf{z}\|_1$, and $\mathbf{y}-\mathbf{z}$ is a circulation on G. We now show that spectral error induced by this circulation $\mathbf{y}-\mathbf{z}$ is not significant in the directed Laplacians. For every edge $e \notin T$, we let $\mathbf{c}^{(T,e)} \in \{0,1\}^E$ denote the (signed) incidence vector of the unique cycle in $T \cup e$. We observe that $\mathbf{z}-\mathbf{y}$ can be expressed uniquely as $\sum_{e\notin T}\mathbf{z}_e\mathbf{c}^{(T,e)}$, so

$$\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top(\mathbf{Y}-\mathbf{Z})\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}} \leq \sum_{e\notin T}|\mathbf{z}_e|\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}.$$

It suffices to show that each operator norm in the right-hand side is bounded by n. Note that

$$\begin{aligned}\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}} &= \sqrt{\left\|(\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}})(\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}})^\top\right\|_{\mathrm{op}}}\\ &= \sqrt{\left\|\mathbf{L}^{\frac{\dagger}{2}}(\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H})\mathbf{L}^\dagger(\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H})^\top\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}}.\end{aligned} \tag{43}$$

We will bound the norm of the last matrix in the above expression. Observe that $\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H}$ is just the directed Laplacian of the cycle with unit weights. Denote it $\mathbf{M}$ for brevity. We further observe

that $\mathbf{M}^\top\mathbf{M}$ is twice the undirected Laplacian of the cycle with unit weights. Since the cycle with unit weights is a downweighted subgraph of (the undirected graph) G, we have $\mathbf{M}^\top\mathbf{M} \preceq 2\mathbf{L}$. Thus,

$$\mathbf{M}\mathbf{L}^\dagger\mathbf{M}^\top \preceq 2\mathbf{M}(\mathbf{M}^\top\mathbf{M})^\dagger\mathbf{M}^\top \preceq 2\mathbf{I}_V.$$

This implies

$$\mathbf{L}^{\frac{\dagger}{2}}(\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H})\mathbf{L}^\dagger(\mathbf{B}^\top\mathbf{C}^{(T,e)}\mathbf{H})^\top\mathbf{L}^{\frac{\dagger}{2}} \preceq 2\mathbf{L}^\dagger \preceq 2\mathbf{L}_T^\dagger.$$

Since T has edge weights ≥ 1 and diameter $\leq n$, $\|\mathbf{L}_T^\dagger\|_{\mathrm{op}} \leq \frac{n^2}{4}$ [Moh91]. By using this bound in (43) and taking square roots, we obtain the third result.

To see the last result, we bound using the triangle inequality:

$$\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{B}^\top\mathbf{Y}\mathbf{H}\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}} \leq \sum_{e\in T}|\mathbf{y}|_e\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}.$$

Note that $\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{e}_{h(e)}\mathbf{b}_e^\top = \mathbf{b}_e\mathbf{b}_e^\top \preceq \mathbf{L}$. Therefore, using $\|\mathbf{L}_T^\dagger\|_{\mathrm{op}} \leq \frac{n^2}{4} \leq n^2$, we have the claim:

$$\begin{aligned}\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}} &= \sqrt{\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{L}^\dagger\mathbf{e}_{h(e)}\mathbf{b}_e^\top\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}}\\ &\leq \sqrt{\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{L}_T^\dagger\mathbf{e}_{h(e)}\mathbf{b}_e^\top\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}}\\ &\leq n\sqrt{\left\|\mathbf{L}^{\frac{\dagger}{2}}\mathbf{b}_e\mathbf{e}_{h(e)}^\top\mathbf{e}_{h(e)}\mathbf{b}_e^\top\mathbf{L}^{\frac{\dagger}{2}}\right\|_{\mathrm{op}}} \leq n.\end{aligned}$$

□

C Potential improvements to Theorem 4

In this section, we discuss two natural avenues to improve the sparsity of our sparsifier construction in Theorem 4: improving the matrix discrepancy result in Proposition 20, and obtaining a graph decomposition with stronger guarantees than Proposition 10.

Partial coloring matrix Spencer. Consider the following conjecture.

Conjecture 60 (Partial coloring matrix Spencer)**.** *There is a constant $\gamma \in (0,1)$ such that for $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}^n$ with $\|\sum_{i\in[m]}\mathbf{A}_i^2\|_{\mathrm{op}} \leq 1$, there exists $\mathbf{x} \in [-1,1]^m$ such that*

$$|\{i \in [m] \mid |\mathbf{x}_i| = 1\}| \geq \gamma m, \text{ and } \left\|\sum_{i\in[m]}\mathbf{x}_i\mathbf{A}_i\right\|_{\mathrm{op}} \leq \frac{1}{\gamma}.$$

By observation, applying the posited coloring in Conjecture 60 in place of Proposition 20 and Corollary 22 when designing our ExistentialDecompSparsify (see the proof of Lemma 25)would remove the last low-order term in Theorem 4, giving a sparsity bound of $O(n\log U + n\log(n)\cdot\varepsilon^{-2})$, which is $O(n\log(n)\cdot\varepsilon^{-2})$ for $U = \mathrm{poly}(n)$. Conjecture 60 has already been stated implicitly or explicitly in the literature in several forms (see e.g., Conjecture 3 in [RR23] with $p = \infty$). Notably, it is stronger than the matrix Spencer conjecture, which asserts (in the most prominent special case) that for a set of matrices $\{\mathbf{A}_i\}_{i\in[n]} \in \mathbb{S}^n$ with $\|\mathbf{A}_i\|_{\mathrm{op}} \leq 1$ for all $i \in [n]$, there exists $\mathbf{x} \in \{\pm 1\}^n$ such that $\|\sum_{i\in[n]}\mathbf{x}_i\mathbf{A}_i\|_{\mathrm{op}} = O(\sqrt{n})$. In the context of Conjecture 60, considering the matrices $\frac{1}{\sqrt{n}}\mathbf{A}_i$,

the assumption is satisfied since $\frac{1}{n}\sum_i \mathbf{A}_i^2 \preceq \mathbf{I}$, and hence Conjecture 60 implies a partial coloring with spectral discrepancy $O(1)$ (i.e., $\mathbf{x} \in [-1,1]^n$ with a constant fraction of coordinate magnitudes equal to 1). Standard boosting techniques (see, e.g., [Gia97] or Section 4 of [Rot17]) show that we can recurse upon this partial coloring scheme to obtain a full coloring in $\{\pm 1\}^n$, since the matrix variance decreases by a constant factor in each iteration.

We also note that Conjecture 60 has already been established in prominent settings, when the matrices $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}^n$ are all low-rank. For example, Theorem 1.4 of [KLS20] proves Conjecture 60 for rank-1 matrices (with a precise constant $\gamma = \frac{1}{4}$), and if all $\{\mathbf{A}_i\}_{i\in[m]}$ have images supported in the same $O(\sqrt{n})$-dimensional subspace, Theorem 3.5 of [HRS22] also proves the claim. For completeness, using tools recently developed in [BJM23], we provide a proof of Conjecture 60 in one of the strongest settings we are aware of known in the literature.

Proposition 61 (Lemma 3.1, [BJM23]). *There is a constant $\gamma \in (0,1)$ such that for $\{\mathbf{A}_i\}_{i\in[m]} \subset \mathbb{S}^n$ with $\|\sum_{i\in[m]} \mathbf{A}_i^2\|_{\mathrm{op}} \le \sigma^2$ and with $\sum_{i\in[m]} \|\mathbf{A}_i\|_{\mathrm{F}}^2 \le mf^2$, there exists $\mathbf{x} \in [-1,1]^m$ such that*

$$|\{i \in [m] \mid |\mathbf{x}_i| = 1\}| \ge \gamma m, \text{ and } \left\|\sum_{i\in[m]} \mathbf{x}_i \mathbf{A}_i\right\|_{\mathrm{op}} \le \frac{1}{\gamma}\left(\sigma + \sqrt{\sigma f}\log^{\frac{3}{4}}(n)\right).$$

Corollary 62. *If the images of all $\mathbf{A}_i$ are supported in the same r-dimensional subspace and $m \ge r \cdot \log^3 n$, Conjecture 60 is true.*

Proof. By linearity of trace, we can choose f such that

$$f^2 = \frac{1}{m}\sum_{i\in[m]} \mathrm{Tr}\left(\mathbf{A}_i^2\right) = \frac{1}{m}\mathrm{Tr}\left(\sum_{i\in[m]} \mathbf{A}_i^2\right) \le \frac{r}{m}\left\|\sum_{i\in[m]} \mathbf{A}_i^2\right\|_{\mathrm{op}} \le \frac{r}{m},$$

where we use that the rank of $\sum_{i\in[m]} \mathbf{A}_i^2$ is at most r. The resulting discrepancy bound is

$$O\left(1 + \sqrt[4]{\frac{r}{m}} \cdot \log^{\frac{3}{4}}(n)\right)$$

which proves the claim for sufficiently small γ, under the assumed parameter bounds. □

For example, while Corollary 62 does not establish Conjecture 60 in full generality, it does establish it when m is larger than n by a polylogarithmic factor, as we may take $r = n$.

Stronger effective resistance decomposition. We further observe that another avenue to improving Theorem 4 is via strengthening Proposition 10, the graph decomposition result it is based on. We present one source of optimism that the parameters in Proposition 10, which gives an $(O(\frac{n\log n}{m}), O(1), O(\log U))$-ER decomposition, can be directly improved, though this remains an open question suggested by our work. In particular, we use the following claim in [AALG18].

Proposition 63 (Theorem 3, [AALG18]). *Given $G = (V, E, \mathbf{w})$ with $n = |V|$ and sufficiently large $C > 1$, there is a constant $\alpha \in (0,1)$ and a polynomial-time algorithm which finds a partition $V = \{V_j\}_{j\in[J]}$ such that if $\{G_j \overset{\text{def}}{=} G[V_j]\}_{j\in[J]}$ are the corresponding induced subgraphs, we have*

$$\sum_{e \in E \setminus \bigcup_{j\in[J]} E(G_j)} \mathbf{w}_e \le \frac{\sum_{e\in E} \mathbf{w}_e}{C\alpha}, \tag{44}$$

and

$$\max_{u,v\in V_j} \mathrm{ER}_G(u,v) \le \frac{n}{C^3\alpha \sum_{e\in E} \mathbf{w}_e} \text{ for all } j \in [J]. \tag{45}$$

Proposition 63 immediately implies an improvement of Proposition 10 when $\mathbf{w}$ is well-behaved.

Corollary 64. *If $G = (V, E, \mathbf{w})$ has $\mathbf{w} \in [1, U]^E$ for $U = \Theta(1)$, there exists a $(\frac{n}{\beta m}, \infty, 1)$-ER decomposition of G, for a constant $\beta \in (0, 1)$.*

Proof. Let $m \stackrel{\text{def}}{=} |E|$. Apply Proposition 63 to G with parameter $C \leftarrow \frac{2U}{\alpha} = \Theta(1)$. The guarantee (44) implies that the total cut weight is at most $\frac{m}{2}$, so less than half the edges are cut as $\min_{e \in E} \mathbf{w}_e \geq 1$. Further, (45) shows that the ρ parameter in Definition 9 is bounded by $U \cdot \frac{n}{C^3 \alpha m} = \Theta\left(\frac{n}{m}\right)$, as desired. Each vertex appears in at most one decomposition piece by definition. □

The main difference between the statement of Proposition 63 and that needed to generalize Corollary 64 beyond the bounded weight ratio case is that Proposition 63 measures the cut edges by the amount of total weight cut, rather than the number of edges cut. Indeed, for a general n-vertex, m-edge graph $G = (V, E, \mathbf{w})$ with $\mathbf{w} \in [1, U]^E$ but where U may be superconstant, let $W \stackrel{\text{def}}{=} \sum_{e \in E} \mathbf{w}_e$, and let $G' = (V, E', \mathbf{w}_{E'})$ where $E' \subseteq E$ removes any edge in E with weight larger than $\frac{4W}{m}$ (so $|E'| \geq \frac{3m}{4}$). Applying Proposition 63 with any constant C on G' yields

$$\left(\max_{e \in E(G_j)} \mathbf{w}_e\right)\left(\max_{u,v \in V_j} \mathrm{ER}_G(u, v)\right) = O\left(\frac{W}{m}\right) \cdot O\left(\frac{n}{W}\right) = O\left(\frac{n}{m}\right),$$

as desired. Unfortunately, the claim (44) does not imply few edges are cut in this case, though for sufficiently large C, it does imply only a small fraction of total weight is cut.

We conclude this section by mentioning one barrier to improving the guarantees of [AALG18], towards obtaining a variant of Corollary 64 which holds for superconstant weight ratios U. In particular, no single decomposition of G's vertices can simultaneously guarantee a bounded effective resistance diameter while cutting a small number of edges, as the following example demonstrates.

Let H be a path graph with all edge weights 1, and let G equal H plus a clique with edge weights n^{-4}. Since $\mathbf{L}_H \preceq \mathbf{L}_G \preceq 2\mathbf{L}_H$, we have $\frac{1}{2}\mathrm{ER}_H(u, v) \leq \mathrm{ER}_G(u, v) \leq \mathrm{ER}_H(u, v)$ for any vertices u, v. We claim that any vertex-disjoint partition of G which cuts at most $\frac{m}{2}$ edges must have one component with resistance diameter $\Omega(n)$. Indeed, any partition $P_1, P_2, \ldots P_k$ with $|P_i| = n_i$ does not cut exactly $\sum_{i=1}^{k} \frac{n_i(n_i-1)}{2}$ edges: as this must be more than $\frac{m}{2}$, we have

$$\frac{m}{2} \leq \sum_{i \in [k]} \frac{n_i(n_i - 1)}{2} \leq \frac{1}{2}\left(\max_{i \in [k]} n_i - 1\right) \sum_{i \in [k]} n_i = \frac{n}{2}\left(\max_{i \in [k]} n_i - 1\right).$$

Since $m = \frac{n(n-1)}{2}$, the largest partition piece has $\geq \frac{n}{2}$ vertices, and since any path of length k has resistance diameter k, this piece has diameter $\Omega(n)$. Thus any potential application of the techniques of [AALG18] towards improving Proposition 10 must partition its input by both vertices and edges; extending [AALG18]'s approach to subsets of edges is an intriguing open question.

Finally, we mention that the definition of a graph decomposition highlighted in this work, the ER decomposition of Definition 9, may not be the only useful notion of decomposition for constructing Eulerian sparsifiers. A potentially fruitful open direction is to explore other related decomposition notions, for which there may be better bounds bypassing difficulties with ER decompositions.

D Proof of Proposition 20

In this section, we show how to modify the proof of Lemma 3.1 in [BJM23] to yield the tighter concentration bound claimed in Proposition 20. In particular, we show how to obtain the second

argument in the minimum, since the first was already shown by [BJM23]. To do so, we recall the following known concentration bounds from [Tro18; BBH23].

Proposition 65 (Corollary 3.6, [Tro18]). *Let $n \geq 8$ and $\{\mathbf{A}_i\}_{i\in[m]} \in \mathbb{S}^n$ satisfy $\|\sum_{i\in[m]} \mathbf{A}_i^2\|_{\mathrm{op}} \leq \sigma^2$ and $\max_{\mathbf{U},\mathbf{V},\mathbf{W}\in\mathbb{U}^n} \|\sum_{i,j\in[m]} \mathbf{A}_i\mathbf{U}\mathbf{A}_j\mathbf{V}\mathbf{A}_i\mathbf{W}\mathbf{A}_j\|_{\mathrm{op}} \leq w$. Then, for $\mathbf{g} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$, there is a universal constant C_{tro} such that*

$$\mathbb{E}\left\|\sum_{i\in[m]} \mathbf{g}_i\mathbf{A}_i\right\|_{\mathrm{op}} \leq C_{\mathrm{tro}} \cdot \left(\sigma \log^{\frac{1}{4}} n + w \log^{\frac{1}{2}} n\right)$$

Lemma 66 (Proposition 4.6, [BBH23]). *For $\{\mathbf{A}_i\}_{i\in[m]} \in \mathbb{S}^n$,*

$$\max_{\mathbf{U},\mathbf{V},\mathbf{W}\in\mathbb{U}^n} \left\|\sum_{i,j\in[m]} \mathbf{A}_i\mathbf{U}\mathbf{A}_j\mathbf{V}\mathbf{A}_i\mathbf{W}\mathbf{A}_j\right\|_{\mathrm{op}} \leq \left\|\sum_{i\in[m]} \mathbf{A}_i^2\right\|_{\mathrm{op}}^{\frac{1}{2}} \cdot \left\|\sum_{i\in[m]} \mathrm{vec}(\mathbf{A}_i)\mathrm{vec}(\mathbf{A}_i)^\top\right\|_{\mathrm{op}}^{\frac{1}{2}}, \tag{46}$$

where $\mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{n^2}$ is the vectorization of $\mathbf{A}$.

By combining Proposition 65 and Lemma 66, we have the following corollary.

Corollary 67. *Let $n \geq 8$ and $\{\mathbf{A}_i\}_{i\in[m]} \in \mathbb{S}^n$ satisfy $\|\sum_{i\in[m]} \mathbf{A}_i^2\|_{\mathrm{op}} \leq \sigma^2$, $\sum_{i\in[m]} \|\mathbf{A}_i\|_{\mathrm{F}}^2 \leq mf^2$. Then, for $\mathbf{g} \sim \mathcal{N}(\mathbf{0}_m, \mathbf{I}_m)$, there is a universal constant C_{tro} such that*

$$\mathbb{E}\left\|\sum_{i\in[m]} \mathbf{g}_i\mathbf{A}_i\right\|_{\mathrm{op}} \leq C_{\mathrm{tro}} \cdot \left(\sigma \log^{\frac{1}{4}} n + \sqrt{\sigma f} \log^{\frac{1}{2}} n\right).$$

Proof. It suffices to combine Proposition 65 and Lemma 66, where we use

$$\left\|\sum_{i\in[m]} \mathrm{vec}(\mathbf{A}_i)\mathrm{vec}(\mathbf{A}_i)^\top\right\|_{\mathrm{op}} \leq \frac{1}{m}\mathrm{Tr}\left(\sum_{i\in[m]} \mathrm{vec}(\mathbf{A}_i)\mathrm{vec}(\mathbf{A}_i)^\top\right) = \frac{1}{m}\sum_{i\in[m]} \|\mathbf{A}_i\|_{\mathrm{F}}^2.$$

The first inequality used that the summed vectorized outer products has rank at most m. □

By replacing Theorem 1.2 of [BBH23] with Corollary 67 in the proof of Lemma 3.1 in [BJM23], we obtain the second term in Proposition 20; we may use the better of the two bounds. To handle the $n \geq 8$ constraint, for any smaller n, we can pad with zeroes up to dimension $n = 8$, which does not affect any operator norms and only changes constants in the claim.