On Constructing Spanners from Random Gaussian Projections

Sepehr Assadi
Department of Computer Science, Rutgers University, New Brunswick, NJ, USA

Michael Kapralov
School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland

Huacheng Yu
Department of Computer Science, Princeton University, NJ, USA

Abstract

Graph sketching is a powerful paradigm for analyzing graph structure via linear measurements introduced by Ahn, Guha, and McGregor (SODA'12) that has since found numerous applications in streaming, distributed computing, and massively parallel algorithms, among others. Graph sketching has proven to be quite successful for various problems such as connectivity, minimum spanning trees, edge or vertex connectivity, and cut or spectral sparsifiers. Yet, the problem of approximating shortest path metric of a graph, and specifically computing a spanner, is notably missing from the list of successes. This has turned the status of this fundamental problem into one of the most longstanding open questions in this area.

We present a partial explanation of this lack of success by proving a strong lower bound for a *large family of graph sketching algorithms* that encompasses prior work on spanners and many (but importantly not also all) related cut-based problems mentioned above. Our lower bound matches the algorithmic bounds of the recent result of Filtser, Kapralov, and Nouri (SODA'21), up to lower order terms, for constructing spanners via the same graph sketching family. This establishes near-optimality of these bounds, at least restricted to this family of graph sketching techniques, and makes progress on a conjecture posed in this latter work.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Sparsification and spanners

Keywords and phrases sketching algorithm, lower bound, graph spanner

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.57

Category RANDOM

Related Version *Full Version*: https://arxiv.org/abs/2209.14775

1 Introduction

Analyzing structure of different objects via random linear projections, also known as *sketching*, is a fundamental paradigm that arise in various contexts. Canonical examples of this approach include dimensionality reduction results such as Johnson-Lindenstrauss lemma [25], sparse recovery results in compressed sensing [16], approximation algorithms for large matrices [40, 41], or various sketches for statistical estimation such as AMS sketch [4], count sketch [12], or count-min sketch [14] in data streams.

A pioneering work of [1] initiated *graph sketching* that considers this paradigm for graphs. A graph sketching algorithm samples a sketching matrix A from a fixed distribution, independent of the input graph G, and compute $A \cdot R(G)$ where R is a suitable representation of G chosen by the algorithm designer, say, its adjacency matrix, Laplacian, or (signed) edge-incidence matrix. The algorithm then uses $A \cdot R(G)$, referred to as the *sketch*, to (approximately) discover properties of G with no further access to G, e.g., to determine

whether or not G is connected. Assuming one can design a sketching matrix A with "few" rows, this approach leads to sketches that can be stored and updated efficiently and be used to recover fundamental properties of G.

The linearity of the sketches and the natural "composability" guarantee that comes with it makes graph sketching a versatile tool in many applications. For instance, graph sketching is the de facto method of algorithm design for *dynamic streaming* algorithms that process a graph specified via a sequence of edge insertions and deletions; see, e.g. [1, 2, 30, 6, 31, 21]. Graph sketching works seamlessly in this model as linearity of sketches allows one to update them easily after each update in the stream. It is even known that this method is universal for dynamic streams under certain (strong) assumptions on length of the stream [35, 3] (see also [28] for necessity of these assumptions). Another model that has benefited greatly from graph sketching is that of *distributed sketching* (a.k.a. simultaneous communication model or broadcast congested clique) wherein every vertex is a processor that sees only edges incident on the vertex and its task is to communicate a small message, simultaneously with other vertices, that allows a referee to solve the problem; see, e.g. [10, 1, 2, 11, 37, 7, 42]. Finally, graph sketching has also been a powerful tool for designing distributed or massively parallel algorithms; see, e.g. [1, 2, 24, 22, 38, 27, 20, 21].

All these considerations have turned graph sketching into a highly attractive solution concept in the last decade since their introduction in [1]. We now have efficient sketches, that often match existentially optimal bounds up to poly-log factors[1], for various fundamental problems such as connectivity [1], minimum spanning trees [1], edge connectivity [1], vertex connectivity [23], cut sparsifiers [2], spectral sparsifiers [30, 31], graph coloring [5], densest subgraph [36], and others.

Graph sketching for spanners

We study graph sketching for the problem of computing *spanners* that (approximately) preserve the shortest path metric of the input graph. Formally,

▶ **Definition 1.** *A subgraph H of a graph $G = (V, E)$ is a d-spanner of G for some integer $d \geq 1$, called the stretch of the spanner, if for every pair $u, v \in V$ one has*

$$dist_G(u, v) \leq dist_H(u, v) \leq d \cdot dist_G(u, v),$$

where $dist_*(\cdot, \cdot)$ stands for the shortest path metric of the corresponding graph.

For every integer $k \geq 1$, every n-vertex graph $G = (V, E)$ admits a $(2k-1)$-spanner with only $O(n^{1+1/k})$ edges which is also existentially optimal under the widely-believed Erdős Girth Conjecture. For instance, every graph admits an $O(\log n)$-spanner on $O(n)$ edges.

Spanners are notably absent from the list of successes in graph sketching. Indeed, despite the significant attention given to sketching spanners, see, e.g., [2, 34, 20, 21, 17], until very recently, it was not even known whether an $o(n)$-spanner can be recovered via sketches of $O(n)$ size. The work of [21] made the first progress on this problem in nearly a decade by presenting an $O(n^{2/3})$-spanner using sketches of $\widetilde{O}(n)$ size[2], or more generally a d-spanner using sketches of size $\widetilde{O}(n^2/d^{3/2})$. But such bounds are still quite far from existential bounds on spanners dictated by the girth conjecture. Yet, no non-trivial lower bounds are known for

[1] For instance, sketches of size $O(n \log^3 n)$ for spanning forests of n-vertex graphs [1] compared to existential bound of $\Omega(n \log n)$ bits to store the spanning forest.

[2] We use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ notation to hide poly-log factors.

this problem[3], beside the work of [37] (see also [42]) that proves that finding *any* spanning tree requires sketches of size $\Omega(n \log^3 n)$ bits (namely, a lower bound for any spanner of finite stretch).

The lack of progress on understanding graph sketching for spanners have also been consequential in other computing models that use graph sketching as their primary tool, most notably, the dynamic streaming model. Indeed, complexity of spanners has been a tantalizing open question in the dynamic streaming model already since its introduction in [1] (for insertion-only streams, optimal algorithms that essentially match existential bounds under Erdős girth conjecture have been known since the introduction of the model itself in [18]; see, also [9, 19]).

This state-of-affairs raises the following question: *What is the best stretch-vs-size tradeoff possible for constructing spanners via graph sketching?* We make progress on this longstanding open question by proving a nearly-tight lower bound for a large family of graph sketching algorithms that encompasses prior work on spanners in [21] and most other closely related problems.

1.1 Our Contribution

We prove **a lower bound on the size of a special case yet general family of sketches for graph spanners**. This family, that shall be defined shortly, contains the prior sketching algorithm of [21] for graph spanners – our lower bound matches their bound up to lower order terms and is thus **nearly-optimal**. In addition, this family also contains many prior sketching algorithms for "cut-based" problems such as connectivity [1], vertex connectivity [23], and spectral sparsifiers [30] (and thus also cut sparsifiers). We now elaborate more on our results, starting with the definition of our sketches, which we call *random Gaussian sketches*.

Random Gaussian sketches

To date, the main success of graph sketching has been for *cut-based* problems [1, 2, 30, 23]. These sketches all work by encoding a graph G as its $\binom{n}{2} \times n$ *signed edge-incidence matrix* $B(G)$ (see Section 3.1) and then apply a sketching matrix A with few rows on the left to obtain the sketch $A \cdot B(G)$. The power of these sketches comes from surprisingly powerful cancellations that the use of the signed edge incidence matrix enables. In addition, the sketching matrix A of in these approaches implements a *sparse recovery* scheme on carefully chosen random subgraphs of the input graphs (e.g. uniformly random subgraphs of the input graph in the case of connectivity [1], cut sparsifiers [2], or spectral sparsifiers [30], and sampled vertex induced subgraphs in the case of spanners [21]).

To give a concrete example, let us consider the AGM sketches [1] for finding spanning forests. For any graph $G = (V, E)$ and any set of vertices $S \subseteq V$, adding up the columns of $B(G)$ corresponding to vertices in S, i.e., $\sum_{v \in S} B(G)^v$ gives us a vector with non-zero entries corresponding to edges of the cut $(S, V \setminus S)$. The linearity of matrix A then allows us to obtain

$$A \cdot \left(\sum_{v \in S} B(G)^v \right) = \sum_{v \in S} A \cdot B(G)^v,$$

[3] This state-of-affairs is in sharp contrast with another widely-studied problem of finding large matchings which is also absent from the list of successes in graph sketching; for the matching problem, *asymptotically tight* lower bounds which are much stronger than existential bounds are known; see [6, 15, 8].

for a cut S specified in the recovery phase. The sketching matrix A itself is an ℓ_0-sampler sketch that samples a non-zero entry of a vector v given $A \cdot v$ (see [26, 32]). An ℓ_0-sampler sketch is typically implemented via a simple sparse recovery sketch combined with a sampling matrix that samples the edges of the graph at $O(\log n)$ geometrically decreasing rates. Combined with the above approach, we can thus sample an edge from any cut of the graph specified in the recovery phase. The algorithm of [1] heavily builds on this subroutine by implementing Borůvka's algorithm for growing connected components via using these sketches to find an outgoing edge from each component in each step.

In this paper, we focus on this family of sketches where the sparse recovery scheme is implemented using *random Gaussian projections*. This means that each *row* of the sketching matrix is of the type $g \cdot S$ where S is an $\binom{n}{2} \times \binom{n}{2}$-dimensional diagonal sampling matrix – where $S_{(u,v),(u,v)} = 1$ iff (u, v) is sampled – and g is an $\binom{n}{2}$-dimensional vector of independent Gaussian variables:

$$\begin{bmatrix} \quad g \quad \end{bmatrix}_{1\times\binom{n}{2}} \times \begin{bmatrix} \quad S \quad \end{bmatrix}_{\binom{n}{2}\times\binom{n}{2}} \times \begin{bmatrix} \quad B(G) \quad \end{bmatrix}_{\binom{n}{2}\times n} = \begin{bmatrix} \quad g \cdot S \cdot B(G) \quad \end{bmatrix}_{1\times n}.$$

The entire sketch is obtained by taking s such rows where sampling matrices can be correlated but Gaussian vectors are independent. The recovery algorithm is given sampling matrices and the sketch but *not* Gaussian variables. We refer to s as the *dimension* of the sketch (thus size of the sketch is $O(s \cdot n)$). See Section 3.1 for formal definitions.

General "power" of random Gaussian sketches? In Appendix A, we show this family of sketches can implement many (but importantly *not* all) prior cut-based sketching algorithms in [1, 30, 23], and most importantly the spanner sketch of [21]. But we also point out that these sketches are *not* universal and one can easily construct problems where the power of these sketches does not match general sketching algorithms[4]. Perhaps more importantly, we assume that the recovery algorithm of these sketches is *oblivious* to the Gaussian vectors used in the sketching matrix which means that the recovery algorithm has a *partial* knowledge of the sketching matrix. A particular shortcoming of this is that while these sketches handle the "main" source of cancelations enabled by edge-incidence matrix, they do *not* handle a "secondary" source of cancelation: to obtain sketches of subgraphs of the input by generating the sketching matrix again at the recovery phase, apply it on some recovered part of the input, and subtract it from the original sketch (this approach is used in the edge connectivity and cut sparsifier sketch of [2] – although we note that random Gaussian sketches can recover a cut sparsifier by instead implementing the algorithm of [30]). We thus see the merit of study of this family as arguably the "most natural" candidate for finding spanners, given their past successes for closely related problems.

Our result

We prove a near-optimal lower bound on the dimension of random Gaussian sketches for constructing spanners, or even returning the distance of two fixed vertices (see also Theorem 5).

[4] Consider recovering the induced subgraph of the input on the first $\sqrt{n}$ vertices. A sparse recovery algorithm that spends $O(\sqrt{n})$ bits per each of these $\sqrt{n}$ vertices gives a sketch of size $O(n)$ for this problem. However, any random Gaussian sketch requires a dimension of $\Theta(\sqrt{n})$ that *cannot* be amortized over all vertices, leading to a sketch of size $O(n^{3/2})$ instead.

▶ **Result 1.** *Any random Gaussian sketch for constructing a d-spanner with constant probability of success requires dimension $\Omega(n^{1-o(1)}/d^{3/2})$, or put differently, any random Gaussian sketch of dimension s can only achieve a stretch of $\Omega((n/s)^{2/3-o(1)})$. The lower bound applies even to the problem of approximating the distance of two fixed vertices.*

Our lower bounds in Result 1 matches algorithmic bounds of [21] up to the $n^{o(1)}$ term for computing spanners via graph sketching (whose sketches fit the framework of random Gaussian sketches) for all stretch d. This establishes the optimality of these bounds at least among this popular family of graph sketching algorithms. We note that [21] conjectured optimality of their algorithmic bounds among *all* graph sketching techniques; our bounds in Result 1 makes partial progress towards settling this conjecture.

Before moving on, we note that for the case when dimension $s = \text{polylog}(n)$, corresponding to sketches of size $\widetilde{O}(n)$, Result 1 implies a lower bound of $n^{2/3-o(1)}$ on the stretch; this should be contrasted with the $O(\log n)$ bound of existential results on the stretch of spanners with $O(n)$ edges, suggesting that computing spanner is much harder using graph sketching (specifically via random Gaussian sketches) compared to existential bounds and arbitrary algorithms. Finally, the lower bound holds even for the algorithmically easier problem of simply estimating distance of two fixed vertices in the graph, as opposed to recovering the entire shortest path metric via a spanner.

Our techniques

We consider a family of hard instances that form a random chain of cliques of size (n/d) with diameter d, and a single edge e^* that connects two vertices at distance $\Theta(d)$ together (see Figure 1). It is easy to see that such e^* should belong to every $o(d)$-spanner of the graph and we prove that no random Gaussian sketch of "small" dimension can recover e^*. The proof is through analyzing how much a *single* random Gaussian projection can reveal information about e^*, or a bit more formally, the KL-divergence between the resulting sketches of two neighboring graphs that only differ on e^*. The rest follows by summing up this information across the s projections.

To prove the bound for a single projection, we use properties of Gaussian variables and KL-divergence to bound the information revealed about the edge e^* by the *effective resistance* of the *sampled* subgraph of the input after applying the sampling matrix. We prove that the distribution of our input, combined with a *hierarchical expander decomposition* of all edges of sampling matrix, implies that the sampled subgraph of the input form a chain of *expanders* (with proper lower bounds on both expansion and minimum degree). This step requires analyzing expansion of vertex-sampled subgraphs of an expander which can be of independent interest. Lastly, we bound the effective resistance of the edge e^* in this chain of expanders by exhibiting a proper *electrical flow* in the graph using properties of expanders.

Related work

In a recent independent work Chen, Khanna and Li [13] showed, similarly to our work, a lower bound matching the sketching dimension of [21] for linear sketches that can support *continuous* weight updates (as opposed to sketches that are only required to work for unweighted graphs). Thus, from the perspective of the ultimate result, the lower bound of [13] is incomparable to ours. Their lower bound works for more general sketches than ours (although still not universal), but assumes that these sketches work in the continuous weight update model; our lower bound assumes a special sketch structure, but works in the mode

standard setting of unweighted graphs. There is quite a bit of overlap in techniques: both papers use expander decompositions and prove that expanders are preserved under vertex sampling (but the actual proofs of the corresponding lemmas are different).

2 Preliminaries

Notation

We use $\mathcal{N}(\mu, \sigma)$ to denote the Gaussian distribution with mean μ and variance σ^2. For any distributions P and Q, $\mathbb{D}(P \mid\mid Q)$ denotes the *KL-divergence* of P from Q and $\|P - Q\|_{\text{tvd}}$ is the *total variation distance* between P and Q.

For a graph $G = (V, E)$ on n vertices, we use $d_1, \ldots, d_n$ to denote the degrees of vertices in G. For any sets of vertices $S, T \subseteq V$, $E(S, T)$ denotes the set of edges between S and T and $\text{vol}_G(S) := \sum_{v \in S} d_v$ denotes the *volume* of S (we drop the subscript when clear). The *conductance* of G is defined as

$$\varphi(G) := \min_{S \subseteq V} \frac{|E(S, V \setminus S)|}{\min\{\text{vol}(S), \text{vol}(V \setminus S)\}}.$$

We say that G is a φ-*expander* if its conductance is at least φ.

For a graph G, $\mathbf{A}$ is the *adjacency matrix*, $\mathbf{D}$ is the *degree diagonal matrix*, $\mathbf{B}$ is the *signed edge-incidence matrix*, $\mathbf{L}$ is the *Laplacian matrix*, and $\widetilde{\mathbf{L}}$ is the *normalized Laplacian matrix*. The *spectral gap* of G is defined as the second smallest eigenvalue of $\widetilde{\mathbf{L}}$ which is related to the conductance via Cheeger's inequality. Finally, $R^G_{\text{eff}}(u, v)$ denotes the *effective resistance* between u, v when treating edges of G as resistors with unit resistance.

We also use the following (variant of) expander decomposition that bounds the minimum degree of resulting expanders. The proof is a simple modification of standard decompositions, e.g. in [29, 39].

▶ **Proposition 2.** *Let $G = (V, E)$ be any graph on n vertices and m edges, and $\varepsilon \in (0, 1/2)$ and $d_{\min} \geq 1$ be parameters. The vertices of G can be partitioned into subgraphs $H_1, \ldots, H_k$ such that:*

- **(i)** *Each H_i is an ε-expander with minimum degree $d_{\min}$;*
- **(ii)** *At most $8\varepsilon \cdot m \log n + n \cdot d_{\min}$ edges E_0 of G do not belong to any subgraph $\{H_i\}_{i \in [k]}$.*

3 Main Result

We formalize Result 1 in this section. We start by defining the sketching model, using random Gaussian projections, that we study. We then present our lower bound for constructing spanners (and in general preserving shortest path metric) using these sketches. Finally, we give the proof outline of this result here and postpone the proof of its main ingredients to the subsequent sections.

3.1 Random Gaussian Projections and Sketches

For an n-vertex graph $G = (V, E)$, its **signed edge-incidence** matrix is an $\binom{n}{2} \times n$-dimensional matrix $\mathbf{B} = \mathbf{B}(G)$ defined as follows:

- Each column corresponds to a vertex v and each row corresponds to a pair of vertices (u, w);
- The entry $\mathbf{B}_{(u,w),v}$ is either $+1$ if (u, w) is an edge in G and $v = u$, -1 if (u, w) is an edge in G and $v = w$, and 0 otherwise.

Note that for any edge $e = (u, v)$ of G, the corresponding row (u, v) in $\mathbf{B}$ has exactly one $+1$ at column u, one -1 at column v, and is otherwise 0. A row (u, v) of $\mathbf{B}$ which does not have a corresponding edge in G is the all-0 row.

Our sketches are based on taking random Gaussian projections of matrix $\mathbf{B}$, which roughly speaking correspond to sampling edge of G (using any sampling scheme oblivious to the graph), and multiply a Gaussian vector with signed edge-incidence matrix of the resulting graph. Formally,

▶ **Definition 3.** *Let $G = (V, E)$ be an n-vertex graph and consider the following:*

- **(i) Sampling matrix:** *Let $\mathbf{S}$ be a $\binom{n}{2} \times \binom{n}{2}$-dimensional* diagonal *matrix with* 0-1*-values on the diagonal. Notice that the matrix $\mathbf{S} \cdot \mathbf{B}(G)$ is the edge-incidence matrix of the subgraph of G obtained by picking only those edges of G that their corresponding (diagonal) value in $\mathbf{S}$ is 1.*
- **(ii) Gaussian projection:** *Let $\mathbf{g}$ be a $\binom{n}{2}$-dimensional vector of Gaussian random variables, where each entry is sampled independently from $\mathcal{N}(0, 1)$.*

A **random Gaussian projection** *of G with respect to $\mathbf{S}$ is an n-dimensional vector obtained by sampling $\mathbf{g} \sim \mathcal{N}(0,1)^{\binom{n}{2}}$, and returning $\boldsymbol{p} := \mathbf{g} \cdot \mathbf{S} \cdot \mathbf{B}(G)$.*

Using Definition 3, we can define the sketches we focus on as follows.

▶ **Definition 4.** *Let Π be a problem defined on n-vertex graphs $G = (V, E)$. A* ***random Gaussian sketch*** *for Π is defined via the following pair:*

- **(i) Sketching matrices:** *A distribution $\mathcal{D}^{\texttt{smpl}}$ on s-tuples of sampling matrices for some $s \geq 1$.*
- **(ii) Recovery algorithm:** *An algorithm that given s sampling matrices $\mathcal{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_s) \sim \mathcal{D}^{\texttt{smpl}}$ and s random Gaussian projection $\mathcal{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_s)$ of any graph G with respect to these sampling matrices, returns a solution to $\Pi(G)$.*

We refer to s as the dimension of the sketch (note that a sketch of dimension s has size $O(s \cdot n)$).

A random Gaussian sketch for a graph G then consists of sampling the sketching matrices $\mathcal{S}$ from $\mathcal{D}^{\texttt{smpl}}$ (independent of G), receiving random Gaussian projections $\mathcal{P}$, and running the recovery algorithm on $(\mathcal{S}, \mathcal{P})$ to return the solution.

We emphasize that in Definition 4, the recovery algorithm is given the sketching matrices used for random Gaussian projections explicitly, but is *not* given the Gaussian vectors themselves.

We note that our formalization of random Gaussian sketches is new to this paper, albeit it has been used implicitly in prior algorithmic results for in graph sketching literature. In Appendix A, we elaborate more on this connection and point out that how these sketches can be used to solve many of the canonical problems in graph sketching literature such as connectivity, minimum spanning tree, cut or spectral sparsifiers, and most closely related to ours, spanners. But we also emphasize that these sketches are *not* universal – see the discussion on the power of these sketches in Section 1.1.

3.2 The Lower Bound

The following is the formalization of Result 1 that we prove.

▶ **Theorem 5.** *For any absolute constant $\delta \in (0, 1)$, and integers $n \geq 1$ and $1 \leq d \leq n^{2/3-\delta}$, any random Gaussian sketch (Definition 4) that outputs a d-spanner of every given n-vertex graph G with probability at least $2/3$ has dimension (i.e., number of rows)*

$$\Omega\left(\frac{n^{1-\delta}}{d^{3/2}}\right).$$

Moreover, the lower bound continues to hold even if the algorithm is only required to answer the shortest path distance *between two* prespecified *vertices up to a factor of d.*

Theorem 5 can alternatively be seen as proving that any random Gaussian sketch of dimension s can only achieve a stretch of

$$\Omega((\frac{n}{s})^{2/3-\delta}),$$

for any constant $\delta > 0$. In light of the result of [21], the bounds obtained in Theorem 5 are optimal, up to $n^{o(1)}$-factors, for the entire range of dimension s or stretch d. In particular, Theorem 5 implies that to obtain a $n^{2/3-\Omega(1)}$-spanner, one needs random Gaussian sketches of dimension $n^{\Omega(1)}$. This makes progress on a conjecture of [21] that stated the same bounds for *arbitrary* sketches.

Finally, we also mention that Theorem 5 works even for the problem wherein we are given two vertices a and b of the graph, and our goal is to simply determine the distance of a and b in the graph using the sketches. This problem is algorithmically easier than finding a spanner of the graph in that firstly, we do not need to pick subset of edges of the graph G and can preserve the shortest path metric in any desired way, and secondly that we only need to maintain the distance between two vertices and not all pairs. Yet, effectively the entirety of our effort is to prove the result for spanners already and we get this stronger lower bound almost for free using standard ideas.

In the rest of this section, we first present a hard input distribution used to establish Theorem 5. We then state our main technical lemma that bounds the information revealed by a single random Gaussian projection on the graphs sampled from this distribution and show how this lemma easily implies the theorem. The next subsection then includes the proof outline of this technical lemma, whose main ingredients are postponed to the next sections.

3.3 A Hard Input Distribution

For any sufficiently large $n, d > 0$, we define a hard distribution $\mu = \mu(n, d)$ over n-vertex graphs. For simplicity, we prove the lower bound for $(d/2)$-spanners instead – re-parameterizing d then implies the same asymptotic lower bound for exact d-spanners as well (see Figure 1).

Distribution $\boldsymbol{\mu(n, d)}$. A hard input distribution for $(d/2)$-spanners of n-vertex graphs.

1. Partition the vertices V into d groups $V_1, \ldots, V_d$: each $v \in V$ is sent to one of the groups chosen uniformly at random.
2. Let G be a graph obtained by placing a clique on each $V_i \cup V_{i+1}$ for $i \in [d-1]$.
3. Sample a pair of vertices $(u^*, v^*) \in \binom{V}{2}$ independently and return the graph $G + e^*$ for $e^* = (u^*, v^*)$.

In the following, we use $\mathbf{B} = \mathbf{B}(G)$ to the denote the signed edge-incidence matrix of G; we also use $\mathbf{B}(e^*)$ as the edge-incidence matrix of the n-vertex graph consisting of the single edge $e^* = (u^*, v^*)$. We emphasize that the final graph output by the distribution is $G + e^*$ (this notation will be make the latter parts of the proof cleaner).

We first establish a straightforward property of graphs sampled from μ in context of spanners.

▶ **Lemma 6.** *With constant probability over the choice of $(G, e^*) \sim \mu(n, d)$, every $(d/2)$-spanner of $G + e^*$ contains the edge $e^* = (u^*, v^*)$.*

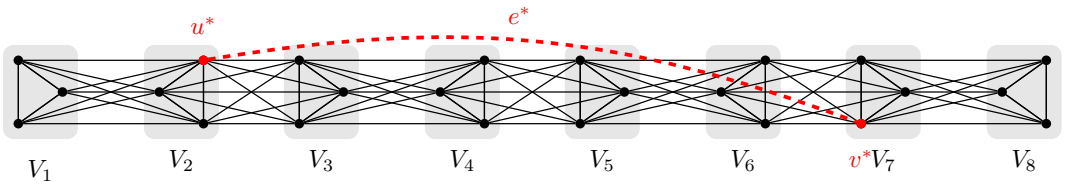

Figure 1 An illustration of $\mu = \mu(n, d)$ for $n = 24$ and $d = 8$. Any 4-spanner of G contains e^*.

Proof. For a graph G and pairs (u^*, v^*) sampled from $\mu(n, d)$,

$$\Pr_{u^*,v^*}(dist_G(u^*, v^*) > d/2) = \frac{1}{d^2} \cdot \left(O(d) + \sum_{i=1}^{d} |d/2 - i| \right) > \frac{1}{5},$$

where the equality holds since when $u^* \in V_1$, v^* can be in $V_{d/2+2}, \ldots, V_d$, when $u^* \in V_2$, v^* can be in $V_{d/2+3}, \ldots, V_d$, and so on ($O(d)$ handles the differences of even or odd choices of d and $d/2$).

Moreover, whenever the distance of u^*, v^* in G is more than $d/2$, any $(d/2)$-spanner of $G + e^*$ should contain the edge e^*, as otherwise the distance between u^* and v^* in the spanner will be more than $d/2$ times their distance in $G + e^*$, violating the bound on the stretch of the spanner. ◀

The following lemma is the main technical contribution of our work. Roughly speaking, this lemma bounds the "information" that can be learned about the edge e^* in μ using a *single* sub-sampled Gaussian projection of a graph sampled from μ.

▶ **Lemma 7.** *Let* $\mathbf{S}$ *be any sampling matrix and consider a single random Gaussian projection with respect to* $\mathbf{S}$*. For* (G, e^*) *sampled from* $\mu = \mu(n, d)$*,*

$$\mathop{\mathbb{E}}_{G,e^*}\left[\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g} \cdot \mathbf{S} \cdot \mathbf{B}(G) \mid\mid \mathbf{g} \cdot \mathbf{S} \cdot (\mathbf{B}(G) + \mathbf{B}(e^*)))\}\right] = O(\frac{d^{3/2}}{n^{1-\delta}}),$$

for any constant $\delta > 0$*, where the KL-divergence is taken only over the Gaussian variables.*

Before getting to the proof of Lemma 7, we show that it implies Theorem 5 immediately.

Proof of Theorem 5 (assuming Lemma 7). Let $(\mathcal{D}^{\texttt{smpl}}, \mathcal{A})$ be any sub-sampled Gaussian sketch of dimension $s \geq 1$ for recovering a $(d/2)$-spanner. Consider a distribution μ' on n-vertex graphs defined as follows:

- Distribution μ': Sample (G, e^*) from μ and $\theta \in \{0, 1\}$ uniformly at random; if $\theta = 0$, return G, otherwise return $G + e^*$.

Let G' be a graph sampled from μ'. Suppose we sample $\mathcal{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_s)$ from $\mathcal{D}^{\texttt{smpl}}$ and receive sub-sampled Gaussian projections $\mathcal{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_s)$ where for every $i \in [s]$, $\boldsymbol{p}_i = \mathbf{g}_i \cdot \mathbf{S}_i \cdot \mathbf{B}(G')$ for a Gaussian vector $\mathbf{g}_i$. Additionally, suppose we are even given $(\mathcal{S}, G, e^*)$, and thus the only unknown information is whether or not $e^* \in G'$ also, i.e., whether $\theta = 1$ or not. This way, we can run $\mathcal{A}$, using $\mathcal{S}, \mathcal{P}$ as input, to obtain a $(d/2)$-spanner of G': if e^* belongs to this spanner, we declare e^* is in G' and otherwise we say it is not. By Lemma 6, we are going to be able to determine the value of θ with probability $1/2 + \Theta(1)$. This implies that over the distribution μ',

$$\|[(\mathcal{S}, G, e^*, \mathcal{P}) \mid \theta = 0] - [(\mathcal{S}, G, e^*, \mathcal{P}) \mid \theta = 1]\|_{\text{tvd}} = \Omega(1), \tag{1}$$

as otherwise, we cannot estimate the value of θ with probability better than $1/2 + o(1)$ given our input $(\mathcal{S}, G, e^*, \mathcal{P})$ which is sampled from either $\mu' \mid \theta = 0$ or $\mu' \mid \theta = 1$. We now have,

$$\begin{aligned}
\text{LHS of Eq (1)} &\leq \mathop{\mathbb{E}}_{(\mathcal{S},G,e^*)} \|[\mathcal{P} \mid \theta = 0, \mathcal{S}, G, e^*] - [\mathcal{P} \mid \theta = 1, \mathcal{S}, G, e^*]\|_{\text{tvd}} \\
&\qquad \text{(as the distribution of } (\mathcal{S}, G, e^*) \text{ is the same under both } \theta = 0 \text{ and } \theta = 1) \\
&\leq \sqrt{\mathop{\mathbb{E}}_{(\mathcal{S},G,e^*)} \min\{1, \mathbb{D}(\mathcal{P} \mid \theta = 0, \mathcal{S}, G, e^* \mid\mid \mathcal{P} \mid \theta = 1, \mathcal{S}, G, e^*)\}} \\
&\qquad \text{(by Pinsker's inequality, the fact that TVD is bounded by 1, and concavity of } \sqrt{\cdot}) \\
&= \sqrt{\mathop{\mathbb{E}}_{(\mathcal{S},G,e^*)} \left[\min\{1, \sum_{i=1}^{s} \mathbb{D}(\boldsymbol{p}_i \mid \theta = 0, \mathcal{S}, G, e^* \mid\mid \boldsymbol{p}_i \mid \theta = 1, \mathcal{S}, G, e^*)\}\right]} \\
&\qquad \text{(by chain rule of KL-divergence as } \boldsymbol{p}_i\text{'s are now only function of } \mathbf{g}_i\text{'s and so are independent)} \\
&\leq \sqrt{\sum_{i=1}^{s} \mathop{\mathbb{E}}_{(\mathcal{S},G,e^*)} \left[\min\{1, \mathbb{D}(\boldsymbol{p}_i \mid \theta = 0, \mathcal{S}, G, e^* \mid\mid \boldsymbol{p}_i \mid \theta = 1, \mathcal{S}, G, e^*)\}\right]} \\
&\qquad \text{(we can take min inside the summation to get an upper bound)} \\
&= \sqrt{\sum_{i=1}^{s} \mathop{\mathbb{E}}_{(\mathbf{S}_i,G,e^*)} \left[\min\{1, \mathbb{D}(\boldsymbol{p}_i \mid \theta = 0, \mathbf{S}_i, G, e^* \mid\mid \boldsymbol{p}_i \mid \theta = 1, \mathbf{S}_i, G, e^*)\}\right]} \\
&\qquad \text{(as } \boldsymbol{p}_i \text{ is only a function of } \mathbf{S}_i \text{ conditioned on } G, e^*) \\
&= \sqrt{\sum_{i=1}^{s} \mathop{\mathbb{E}}_{(\mathbf{S}_i,G,e^*)\sim\mu} \min\{1, \mathbb{D}_{\mathbf{g}_i}(\mathbf{g}_i \cdot \mathbf{S}_i \cdot \mathbf{B}(G) \mid\mid \mathbf{g}_i \cdot \mathbf{S}_i \cdot (\mathbf{B}(G) + \mathbf{B}(e^*)))\}},
\end{aligned}$$

where the last equality is because input graph G' in μ' is G when $\theta = 0$ and $G + e^*$ when $\theta = 1$, and distribution of $(\mathbf{S}_i, G, e^*, \mathbf{g}_i)$ is the same under μ and μ'.

Now given that $\mathbf{S}_i \perp (G, e^*)$ in μ, each term in the RHS above is the same quantity upper bounded in Lemma 7. Thus, combining Eq (1), the above equation, and Lemma 7, we get that

$$\Omega(1) = \|[(\mathcal{S}, G, e^*, \mathcal{P}) \mid \theta = 0] - [(\mathcal{S}, G, e^*, \mathcal{P}) \mid \theta = 1]\|_{\text{tvd}} \leq \sqrt{s \cdot O(\frac{d^{3/2}}{n^{1-\delta}})},$$

which implies that $s = \Omega(n^{1-\delta}/d^{3/2})$ as desired. This implies the first part of Theorem 5.

The proof of the second part follows almost immediately from the above argument as follows. Consider the following distribution:

- Distribution μ'': Sample (G', e^*, θ) from μ'. Add two new vertices a and b to the graph and add edges (a, u^*) and (v^*, b) to the graph as well.

Let G'' be a graph sampled from μ''. Consider the distance between a and b in G'': if $\theta = 1$ in the sampled G', distance of a and b is 3, otherwise, if $\theta = 0$, by the same argument as Lemma 6, the distance between a and b is more than $(d/2)$ with constant probability. This means that if our algorithm could simply estimate the distance of a and b to within a factor of $(d/6)$, it can determine the value of θ with probability $1/2 + \Theta(1)$.

Now if we further give u^*, v^*, and the Gaussian variables on all edges incident to a or b to the recovery algorithm, what the algorithm knows becomes the sketches of $G'' \setminus \{a, b\}$ (by simply subtracting the corresponding Gaussians). Since the Gaussians revealed are independent of the sketch of $G'' \setminus \{a, b\}$, the same exact argument as the first part now implies that the same lower bound of $s = \Omega(d^{3/2}/n^{1-\delta})$ on the sketch dimension. Given that the number of vertices in G'' is $n + 2$, and by re-parameterizing d with a constant factor, we obtain the desired lower bound. This concludes the proof of Theorem 5. ◀

3.4 Proof Outline of Lemma 7

We now present the proof outline of Lemma 7, postponing the proof of its two main ingredients to the next two sections. For convenience, we restate Lemma 7 below.

▶ **Lemma** (Restatement of Lemma 7). *Let* $\mathbf{S}$ *be any sampling matrix and consider a single random Gaussian projection with respect to* $\mathbf{S}$*. For* (G, e^*) *sampled from* $\mu = \mu(n, d)$*,*

$$\mathop{\mathbb{E}}_{G,e^*}\Big[\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G) \mid\mid \mathbf{g}\cdot\mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e^*)))\}\Big] = O(\frac{d^{3/2}}{n^{1-\delta}}),$$

for any constant $\delta > 0$*, where the KL-divergence is taken only over the Gaussian variables.*

To continue, we define the following notation for the sampling matrix $\mathbf{S}$:

- $graph(\mathbf{S})$: the graph on V containing all edges $e \in \binom{V}{2}$ where $\mathbf{S}_{e,e} = 1$.
- $m(\mathbf{S})$: the number of *edges* in $graph(\mathbf{S})$.
- $G(\mathbf{S})$: the subgraph of G on edges that belong to $graph(\mathbf{S})$, i.e., $G(\mathbf{S}) := G \cap graph(\mathbf{S})$. Note that this way we have, $\mathbf{B}(G(\mathbf{S})) = \mathbf{S}\cdot\mathbf{B}(G)$ and $\mathbf{B}(G(\mathbf{S})+e^*) = \mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e^*))$.

Ingredient one: from KL-divergence to effective resistances

The first key step of the proof of Lemma 7 is to relate the KL-divergence term of Lemma 7 to *effective resistance* of the edge e^* in the underlying sampled graph. Formally,

▶ **Lemma 8.** *For any sampling matrix* $\mathbf{S}$*, any fixed* G*, and any pair of vertices* $e = (u, v) \in \binom{V}{2}$*,*

$$\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G) \mid\mid \mathbf{g}\cdot\mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e)))\} \leq 2\cdot R_{\text{eff}}^{G(\mathbf{S})+e}(u, v).$$

We will apply Lemma 8 to the choice of edge $e^* = e$ whenever e^* belongs to $graph(\mathbf{S})$, i.e., when e^* is sampled by the sampling matrix $\mathbf{S}$. To prove Lemma 8, we first calculate the KL-divergence between two high-dimensional Gaussians in terms of their covariance matrices. Then we observe that the covariance matrix of $\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G)$ is simply the *Laplacian matrix* of $G(\mathbf{S})$. The lemma is proved by plugging in the Laplacian matrices of $G(\mathbf{S})$ and $G(\mathbf{S})+e$, and applying the connection between effective resistance and Laplacian matrix. The proof is provided in the full version.

Ingredient two: bounding effective resistances via expanders

Our strategy is now to bound the effective resistance of the edge e^* in $G(\mathbf{S})+e^*$. To do so, we will identify a "good"-*expander* subgraph H of the $graph(\mathbf{S})$ that contains the edge e^*, and then primarily focus on the edges of H that appear in $G(\mathbf{S})$ to bound the effective resistance of e^* also. The following lemma is the heart of the proof.

▶ **Lemma 9.** *For any sampling matrix* $\mathbf{S}$*, suppose* H *is any subgraph of* $graph(\mathbf{S})$ *which is an* ε*-expander with min-degree* D *for some* $\varepsilon > 0$ *and* $D \geq (\varepsilon^{-8}\cdot n^{\delta})\cdot d$ *for a constant* $\delta > 0$*. For any edge* $e = (u, v) \in H$*,*

$$\mathop{\mathbb{E}}_{G}\Big[R_{\text{eff}}^{G(\mathbf{S})+e}(u, v)\Big] = O(\varepsilon^{-4})\cdot\Big(\frac{d}{D}+\frac{d^3}{D^2}\Big).$$

We will use a hierarchical expander decomposition of $graph(\mathbf{S})$ to identify an expander that contains the edge e^* and then apply Lemma 9 to this expander and edge e^*. To prove Lemma 9, we first observe that adding edges to a graph could only decrease the

effective resistance, and thus, it suffices to study the effective resistance of (u, v) in $G \cap H$. Note that $G \cap H$ randomly partitions the vertices into d sets, and only keeps the edges of H with endpoints in the same set or adjacent sets. We then show that because H is an expander with large min-degree, $G \cap H$ restricted to any two adjacent sets must also be an expander with large min-degree with high probability. Hence, $G \cap H$ looks like a "chain of expanders" (which we call *a balanced path of expanders*), where adjacent expanders have a constant fraction overlap. Finally, we show that since $G \cap H$ overall is well-connected, if we place a unit electric flow from u to v, the flow will be well-spread across the graph. Most edges have a small current, i.e., a low potential difference. Therefore, it allows us to argue that the potential difference between u and v is also small, i.e., the effective resistance between u and v is small. The detailed proof is provided in the full version.

Putting everything together

We now put these two ingredients together to prove Lemma 7. In order to be able to apply our second tool in Lemma 9, we need a *hierarchical* expander decomposition of $graph(\mathbf{S})$, which shows that the edge e^* is "more likely" to land in "better" expanders of $graph(\mathbf{S})$ for the purpose of Lemma 9 – here, "better" means an expander with a higher minimum degree (the parameter in Lemma 9 that governs the final bound).

▶ **Lemma 10.** *For every $t \geq 1$, we can partition edges of $graph(\mathbf{S})$ into t sets $E_1(\mathbf{S}), \ldots, E_t(\mathbf{S})$ such that:*

- **(i)** *For any $i \leq t$, define $m_i(\mathbf{S}) := |E_i(\mathbf{S})|$; then, $m_1(\mathbf{S}) \leq m(\mathbf{S})$ and $m_{i+1}(\mathbf{S}) \leq m_i(\mathbf{S})/2$.*
- **(ii)** *For any $i < t$, there is some $k_i \geq 1$ such that edges in $E_i(\mathbf{S})$ can be partitioned into ε-expanders $H_1^i, \ldots, H_{k_i}^i$ with minimum degree at least D_i for parameters*[5]

$$\varepsilon := \frac{1}{36 \log n} \quad \text{and} \quad D_i \geq \frac{m_i(\mathbf{S})}{36n}.$$

Proof. For simplicity of exposition, we drop $(\mathbf{S})$ when denoting $E_i(\mathbf{S})$'s in the following. We construct $E_1, \ldots, E_t$ inductively using an auxiliary set of edges $F_0, \ldots, F_t$. Start with F_0 being the set of all edges in $graph(\mathbf{S})$ and for $i = 1$ to t do:

1) Apply the expander decomposition of Proposition 2 to F_{i-1} with parameters

$$\varepsilon = \frac{1}{36 \log n} \quad \text{and} \quad d_{\min} = D_i = \frac{|F_{i-1}|}{36n},$$

to get ε-expanders $H_1^i, \ldots, H_{k_i}^i$ each with minimum degree at least D_i.

2) Let E_i be the union of edges assigned to the expanders in the decomposition of Proposition 2 in the previous step, and F_i be the leftover edges. Continue to iteration $i + 1$.

We argue that $|F_i| \leq |F_{i-1}|/4$ for all $i \leq t$. For $i > 0$, we have that $|F_i|$ is the number of leftover edges of the decomposition and thus by Proposition 2,

$$|F_i| \leq 8\varepsilon \cdot |F_{i-1}| \cdot \log n + n \cdot D_i = \frac{8\,|F_{i-1}|}{36} + \frac{|F_{i-1}|}{36} = \frac{|F_{i-1}|}{4}.$$

Now firstly, $E_i = F_{i-1} \setminus F_i$ and so by the above bound, $|E_i| \geq 2\,|F_i|$. At the same time, $E_{i+1} \subseteq F_i$ for and thus $|E_i| \geq 2\,|E_{i+1}|$. This proves the first part.

Secondly, we get property (ii) of the lemma by the choice of $\varepsilon = 1/36 \log n$ in the decomposition and since $D_i = |F_{i-1}|/36n \geq |E_i|/36n$ as $E_i \subseteq F_{i-1}$. ◀

[5] Notice that the edges $E_t(\mathbf{S})$ admit no such type of expander decomposition in our partitioning.

We now have all the tools needed to prove Lemma 7. For the rest of the proof, we fix a partitioning $(E_1(\mathbf{S}), \ldots, E_t(\mathbf{S}))$ of $graph(\mathbf{S})$ using Lemma 10 for some $t \geq 1$ such that:

$$t \text{ is the } \underline{\text{largest}} \text{ index where:} \quad m_{t-1}(\mathbf{S}) \geq n^{1+\delta} \cdot d^{3/2}, \tag{2}$$

where $\delta > 0$ is the absolute constant in Lemma 7. This means that for every $e \in E_i(\mathbf{S})$ for $i < t$, the edge e belongs to some ε-expander H^i_j for $j \in [k_i]$ with min-degree D_i such that,

$$\varepsilon = \frac{1}{36 \log n} \qquad \text{and} \qquad D_i \geq \frac{m_i(\mathbf{S})}{36n}. \tag{3}$$

This also implies that $D_i \geq (1/36) \cdot d^{3/2} \cdot n^{\delta} \geq (\varepsilon^{-10} \cdot n^{\delta'}) \cdot d$ for some absolute constant $\delta' > 0$ which allows us to apply Lemma 9 to each expander H^i_j in the proof.

We now have,

$$\begin{aligned}
\text{LHS of Lemma 7} &= \mathop{\mathbb{E}}_{G,e^*}\Big[\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G) \mid\mid \mathbf{g}\cdot\mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e^*)))\}\Big] \\
&= \sum_{e \in graph(\mathbf{S})} \Pr(e^* = e) \cdot \mathop{\mathbb{E}}_{G|e^*=e}\Big[\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G) \mid\mid \mathbf{g}\cdot\mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e^*)))\}\Big] \\
&\qquad \text{(whenever } e^* \notin graph(\mathbf{S}), \text{ both terms of the KL-divergence will be the same and thus it will be 0)} \\
&= \sum_{i=1}^{t} \sum_{e \in E_i(\mathbf{S})} \Pr(e^* = e) \cdot \mathop{\mathbb{E}}_{G}\Big[\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G) \mid\mid \mathbf{g}\cdot\mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e^*)))\}\Big] \\
&\qquad \text{(by the partitioning of edges of } graph(\mathbf{S}) \text{ and since } G \perp e^* \text{ in } \mu) \\
&= \frac{1}{\binom{n}{2}} \sum_{i=1}^{t} \sum_{e \in E_i(\mathbf{S})} \Big[\min\{1, \mathbb{D}_{\mathbf{g}}(\mathbf{g}\cdot\mathbf{S}\cdot\mathbf{B}(G) \mid\mid \mathbf{g}\cdot\mathbf{S}\cdot(\mathbf{B}(G)+\mathbf{B}(e)))\}\Big] \\
&\qquad \text{(as the marginal distribution of } e^* \text{ is uniform over } \tbinom{V}{2} \text{ and we conditioned on } e^* = e) \\
&\leq \frac{m_t(\mathbf{S})}{\binom{n}{2}} + \frac{1}{\binom{n}{2}} \cdot \sum_{i=1}^{t-1} \sum_{e=(u,v) \in E_i(\mathbf{S})} \mathop{\mathbb{E}}_{G}\Big[2 \cdot R^{G(\mathbf{S})+e}_{\text{eff}}(u,v)\Big] \\
&\qquad \text{(using the trivial upper bound of 1 for } E_t(\mathbf{S}) \text{ and Lemma 8 for } E_1(\mathbf{S}), \ldots, E_{t-1}(\mathbf{S})) \\
&= \frac{m_t(\mathbf{S})}{\binom{n}{2}} + \frac{2}{\binom{n}{2}} \cdot \sum_{i=1}^{t-1} \sum_{j=1}^{k_i} \sum_{e=(u,v) \in H^i_j} \mathop{\mathbb{E}}_{G}\Big[R^{G(\mathbf{S})+e}_{\text{eff}}(u,v)\Big] \\
&\qquad \text{(as each } E_i(\mathbf{S}) \text{ for } i < t \text{ is partitioned into expanders } H^i_1, \ldots, H^i_{k_i} \text{ by Lemma 10)} \\
&= \frac{m_t(\mathbf{S})}{\binom{n}{2}} + \frac{2}{\binom{n}{2}} \cdot \sum_{i=1}^{t-1} \sum_{j=1}^{k_i} \sum_{e=(u,v) \in H^i_j} O(\varepsilon^{-4}) \cdot \Big(\frac{d}{D_i} + \frac{d^3}{D_i^2}\Big) \\
&\qquad \text{(by Lemma 9 as each } H^i_j \text{ is an } \varepsilon\text{-expander with min-degree } D_i \text{ (and by Eq (3) we can use the lemma))} \\
&= \frac{m_t(\mathbf{S})}{\binom{n}{2}} + \frac{2}{\binom{n}{2}} \cdot \sum_{i=1}^{t-1} m_i(\mathbf{S}) \cdot O(\log^4 n) \cdot \Big(\frac{d \cdot n}{m_i(\mathbf{S})} + \frac{d^3 \cdot n^2}{m_i(\mathbf{S})^2}\Big) \\
&\qquad \text{(as } \varepsilon = \Theta(1/\log n) \text{ and } D_i \geq m_i(\mathbf{S})/12n \text{ by Eq (3) and } H^i_1, \ldots, H^i_{k_i} \text{ have } m_i(\mathbf{S}) \text{ edges in total)} \\
&= \frac{m_t(\mathbf{S})}{\binom{n}{2}} + O(\log^5 n \cdot \frac{d}{n}) + O(\log^4 n) \cdot \frac{d^3}{m_{t-1}(\mathbf{S})} \\
&\qquad \text{(as } m_i(\mathbf{S}) \text{'s decrease (at least) by a geometric series and } t = O(\log n) \text{ by Lemma 10)} \\
&\leq \frac{n^{1+\delta} \cdot d^{3/2}}{\binom{n}{2}} + O(\log^5 n \cdot \frac{d}{n}) + O(\log^4 n) \cdot \frac{d^3}{n^{1+\delta} \cdot d^{3/2}} \qquad \text{(by the choice of } t \text{ in Eq (2))} \\
&= O(\frac{d^{3/2}}{n^{1-\delta}}).
\end{aligned}$$

This concludes the proof of Lemma 7. Due to page limits, we omit the proof of Lemma 8 and Lemma 9 here. They can be found in the full version.

References

1. Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 459–467, 2012.
2. Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 5–14, 2012.
3. Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 20:1–20:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.
4. Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 20–29. ACM, 1996.
5. Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786. SIAM, 2019.
6. Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016.
7. Sepehr Assadi, Gillat Kol, and Rotem Oshman. Lower bounds for distributed sketching of maximal matchings and maximal independent sets. In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 79–88. ACM, 2020.
8. Sepehr Assadi and Vihan Shah. An asymptotically optimal algorithm for maximum matching in dynamic streams. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 – February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 9:1–9:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
9. Surender Baswana and Sandeep Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Struct. Algorithms*, 30(4):532–563, 2007.
10. Florent Becker, Martín Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, and Ioan Todinca. Adding a referee to an interconnection network: What can(not) be computed in one round. In *25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA, 16-20 May, 2011 – Conference Proceedings*, pages 508–514. IEEE, 2011.
11. Florent Becker, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. The simultaneous number-in-hand communication model for networks: Private coins, public coins and determinism. In Magnús M. Halldórsson, editor, *Structural Information and Communication Complexity – 21st International Colloquium, SIROCCO 2014, Takayama, Japan, July 23-25, 2014. Proceedings*, volume 8576 of *Lecture Notes in Computer Science*, pages 83–95. Springer, 2014.
12. Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan J. Eidenbenz, and Ricardo Conejo, editors, *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, volume 2380 of *Lecture Notes in Computer Science*, pages 693–703. Springer, 2002.

13 Yu Chen, Sanjeev Khanna, and Huan Li. On weighted graph sparsification by linear sketching. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 474–485. IEEE, 2022.

14 Graham Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. In Martin Farach-Colton, editor, *LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, Buenos Aires, Argentina, April 5-8, 2004, Proceedings*, volume 2976 of *Lecture Notes in Computer Science*, pages 29–38. Springer, 2004.

15 Jacques Dark and Christian Konrad. Optimal lower bounds for matching and vertex cover in dynamic graph streams. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

16 David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

17 Michael Elkin and Chhaya Trehan. $(1+\epsilon)$-approximate shortest paths in dynamic streams. *CoRR*, abs/2107.13309, 2021. `arXiv:2107.13309`.

18 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 531–543. Springer, 2004.

19 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008.

20 Manuel Fernandez, David P. Woodruff, and Taisuke Yasuda. Graph spanners in the message-passing model. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 77:1–77:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

21 Arnold Filtser, Michael Kapralov, and Navid Nouri. Graph spanners by sketching in dynamic streams and the simultaneous communication model. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10–13, 2021*, pages 1894–1913. SIAM, 2021.

22 Mohsen Ghaffari and Merav Parter. MST in log-star rounds of congested clique. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 19–28. ACM, 2016.

23 Sudipto Guha, Andrew McGregor, and David Tench. Vertex and hyperedge connectivity in dynamic graph streams. In Tova Milo and Diego Calvanese, editors, *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 – June 4, 2015*, pages 241–247. ACM, 2015.

24 James W. Hegeman, Gopal Pandurangan, Sriram V. Pemmaraju, Vivek B. Sardeshmukh, and Michele Scquizzato. Toward optimal bounds in the congested clique: Graph connectivity and MST. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21–23, 2015*, pages 91–100. ACM, 2015.

25 William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space 26. *Contemporary mathematics*, 26:28, 1984.

26 Hossein Jowhari, Mert Saglam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In Maurizio Lenzerini and Thomas Schwentick, editors, *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 49–58. ACM, 2011.

27 Tomasz Jurdzinski and Krzysztof Nowicki. MST in $O(1)$ rounds of congested clique. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2620–2632. SIAM, 2018.

28 John Kallaugher and Eric Price. Separations and equivalences between turnstile streaming and linear sketching. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1223–1236. ACM, 2020.

29 Ravi Kannan, Santosh S. Vempala, and Adrian Vetta. On clusterings – good, bad and spectral. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 367–377. IEEE Computer Society, 2000.

30 Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 561–570. IEEE Computer Society, 2014.

31 Michael Kapralov, Aida Mousavifar, Cameron Musco, Christopher Musco, Navid Nouri, Aaron Sidford, and Jakab Tardos. Fast and space efficient spectral sparsification in dynamic streams. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1814–1833. SIAM, 2020.

32 Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P. Woodruff, and Mobin Yahyazadeh. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 475–486. IEEE Computer Society, 2017.

33 Michael Kapralov, Navid Nouri, Aaron Sidford, and Jakab Tardos. Dynamic streaming spectral sparsification in nearly linear time and space. *CoRR*, abs/1903.12150, 2019. `arXiv:1903.12150`.

34 Michael Kapralov and David P. Woodruff. Spanners and sparsifiers in dynamic streams. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 272–281. ACM, 2014.

35 Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*, pages 174–183. ACM, 2014.

36 Andrew McGregor, David Tench, Sofya Vorotnikova, and Hoa T. Vu. Densest subgraph in dynamic graph streams. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *Mathematical Foundations of Computer Science 2015 – 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part II*, volume 9235 of *Lecture Notes in Computer Science*, pages 472–482. Springer, 2015.

37 Jelani Nelson and Huacheng Yu. Optimal lower bounds for distributed and streaming spanning forest computation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1844–1860. SIAM, 2019.

38 Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. Fast distributed algorithms for connectivity and MST in large graphs. *ACM Trans. Parallel Comput.*, 5(1):4:1–4:22, 2018.

39 Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2616–2635. SIAM, 2019.

40 Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 143–152. IEEE Computer Society, 2006.

41 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014.

42 Huacheng Yu. Tight distributed sketching lower bound for connectivity. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10–13, 2021*, pages 1856–1873. SIAM, 2021.

A Implementing Prior Work Via Random Gaussian Sketches

We now outline implementations of existing works on graph sketching in our model.

A.1 ℓ_0-samplers and connectivity sketches

Recall that in the ℓ_0-sampling problem one needs to design a sketching matrix A such that for every $x \in \mathbb{R}^n$ one can recover a uniformly random element of x from Ax (to within total variation distance δ) or output FAIL (with failure probability bounded by δ)[6]. We outline a construction of an ℓ_0-sampler in our model, i.e. where every rows of the sketch A is of the form $g \cdot S$, where S is an arbitrary matrix with zeros and ones on the diagonal and zeros on off-diagonal entries (S is known to the decoder) and g is a vector with i.i.d. unit variance Gaussian entries (g is not known to the decoder). Note that our ℓ_0-sampler only needs to work for vectors x whose entries are in $\{-1, 0, +1\}$, as this is the case in all applications of graph sketching.

We first recall the construction of a basic ℓ_0 sampler (see [26] for a space-optimal construction). For integer j between 0 and $\lceil \log_2 n \rceil$ let $x^j \in \mathbb{R}^n$ denote the restriction of x^j to elements of a subset of the universe $[n]$ that includes every element independently with probability 2^{-j}. There exists j^* such that with constant probability x^j contains exactly one nonzero. To determine the value of j^* or conclude that such an index does not exist, it suffices to estimate the ℓ_2^2 norm of x to within a $1 \pm 1/3$ factor, for example (since nonzero entries of x equal 1 in absolute value). The latter can be achieved (with at most inverse polynomial failure probability) by averaging squared dot products of $O(\log n)$ independent Gaussian vectors with x^j, which is allowed by our model. Note that here the decoder indeed does not need to know the Gaussian vectors, as required. If j^* exists, one must recover the identity of the nonzero element. The typical way to do it is to compute the dot product of x^{j^*} with the vector whose i-th coordinate equals i, for every $i \in [n]$. This is not available in our model. To replace this approach, for every $j = 0, \ldots, \lceil \log_2 n \rceil$ and $b = 0, \ldots, \lceil \log_2 n \rceil$ approximate the ℓ_2^2 norm of the vector x^j restricted to the set of elements in $[n]$ that have 1 in the b-th position in their binary representation using $O(\log n)$ dot products with i.i.d. Gaussians. This allows one to read off the binary representation of the nonzero in x^{j^*}, and therefore yields an ℓ_0 sampler.

Graph connectivity and spanning trees

Since an ℓ_0-sampler is the only sketch used by the connectivity sketch of [1], it follows that a spanning forest of the input graph can be recovered by a sketch that fits our model and has a polylogarithmic number of rows.

[6] Note that these two parameters appear differently in the space complexity of ℓ_0-sampling, and are therefore treated separately in works that obtain optimal space bounds for ℓ_0-samplers [1]. We set both parameters to δ for simplicity.

Approximate vertex connectivity

The result of [23] uses the spanning tree sketch of [1] black box (the sketch is applied to random vertex induced subgraphs) to approximate vertex connectivity. Since the sketch of [1] can be implemented in our model, as described above, the result of [23] also can.

A.2 ℓ_2-heavy hitters, spectral sparsifiers and spanners

Recall that in the φ-heavy hitters problem in ℓ_2 one needs to design a sketching matrix A such that for every $x \in \mathbb{R}^n$ one can recover a list of elements $L \subseteq [n]$ such that every $i \in [n]$ satisfying $x_i^2 \geq \varphi \|x\|_2^2$ belongs to L and no $i \in [n]$ with $x_i^2 < c\varphi\|x\|_2^2$ for a constant $c > 0$ belongs to L.

A basic ℓ_2 heavy hitters sketch works by first hashing elements of $[n]$ to $B \approx 1/\varphi$ buckets, i.e. effectively defining x^b for $b \in [B]$ to be the restriction of x to bucket b, and computing the sum of elements of x^b with random signs. In our model we can replace the random signs with random Gaussians, so that the resulting dot product is Gaussian with variance $\|x^b\|_2^2$. Fixing any $j \in [n]$ and letting b denote the bucket that j hashes to we get that a single hashing can be used to obtain an estimate of its absolute value that is correct up to constant factor and an additive $O(1/\sqrt{B})\|x\|_2$ term with probability[7] at least 9/10. We can now repeat the estimator $O(\log n)$ times and include in L elements that are estimated as larger than a $c'\varphi\|x\|_2$ for a sufficiently small constant $c' > 0$ in absolute value. Therefore, setting $B = O(1/\varphi)$ achieves the required bounds. This yields an ℓ_2-heavy hitters sketch with decoding time nearly linear in the size n of the universe. The decoding time can be improved to $(1/\varphi) \cdot \mathrm{poly}(\log n)$ using a bit-encoding approach similar to the one from Section A.1 above.

Spectral sparsifiers and spanners

Spectral sparsification sketches [30, 33, 31] require graph connectivity sketches, which we already implemented in Section A.1, as well as ℓ_2-heavy hitters sketches, and therefore can also be implemented in our model. Non-adaptive sketching algorithms for spanner construction [21] rely on spectral sparsification sketches that are applied to vertex-induced subgraphs of the input graph. Thus, these sketches can also be implemented in our model with at most a polylogarithmic loss in the number of rows.

[7] We use the fact that the dot product of x^b with a random Gaussian vector will be distributed as $g_j x_j + N(0, \|x^b_{-j}\|_2^2)$, and $|g_j|$ is at least a constant with probability at least 9/10. Here x^b_j stands for the vector obtained from x^b by zeroing out entry j.