Reachability Preservers: New Extremal Bounds and Approximation Algorithms

Amir Abboud* [1] and Greg Bodwin† [2]

[1]IBM Almaden Research Center
[2]MIT CSAIL

Abstract

In this paper we prove new results about the extremal structure of paths in directed graphs. Say we are given a directed graph $G = (V, E)$ on n nodes, a set of sources $S \subseteq V$ of size $|S| = n^{1/3}$, and a subset $P \subseteq S \times V$ of pairs (s, t) where $s \in S$, of size $O(n^{2/3})$, such that for all pairs $(s, t) \in P$, there is a path from s to t. Our goal is to remove as many edges from G as possible while maintaining the reachability of all pairs in P. How many edges will we have to keep? Can you always go down to $n^{1+o(1)}$ edges? Or maybe for some nasty graphs G you cannot even go below the simple bound of $O(n^{4/3})$ edges? Embarrassingly, in a world where graph reachability is ubiquitous in countless scientific fields, the current bounds on the answer to this question are far from tight.

In this paper, we make polynomial progress in both the upper and lower bounds for these *Reachability Preservers* over bounds that were implicit in the literature. We show that in the above scenario, $O(n)$ edges will always be sufficient, and in general one is even guaranteed a subgraph on $O(n + \sqrt{n \cdot |P| \cdot |S|})$ edges that preserves the reachability of all pairs in P. We complement this with a lower bound graph construction, establishing that the above result fully characterizes the settings in which we are guaranteed a preserver of size $O(n)$. Moreover, we design an efficient algorithm that can always compute a preserver of existentially optimal size.

The second contribution of this paper is a new connection between extremal graph sparsification results and classical Steiner Network Design problems. Surprisingly, prior to this work, the osmosis of techniques between these two fields had been superficial. This allows us to improve the state of the art approximation algorithms for the most basic Steiner-type problem in directed graphs from the $O(n^{0.6+\varepsilon})$ of Chlamatac, Dinitz, Kortsarz, and Laekhanukit (SODA'17) to $O(n^{0.577+\varepsilon})$.

1 Introduction

In this paper we prove new results about the extremal structure of paths in directed graphs. Suppose we are given a directed graph on n nodes, a set of sources S of size $|S| = n^{1/3}$, and a subset $P \subseteq S \times V$ of pairs (s, t) where $s \in S$, of size $O(n^{2/3})$, such that for all pairs there is a path from s to t in G. Our goal is to remove as many edges from G as possible while maintaining the reachability for all pairs in P, i.e. for all $(s, t) \in P$ there is still a path from s to t. How many edges will we have to keep? It is not hard to see that $O(n^{4/3})$ edges will be sufficient: for each source $s \in S$ we can keep a BFS tree at the cost of $O(n)$ edges, and this will guarantee that s still reaches all the nodes it used to reach. In general this observation gives an upper bound of $O(n|S|)$. Another simple observation is that $\Omega(n)$ edges might be necessary, if for example, the entire graph G is a path of length n and the endpoints are in the set P. But can we improve the $O(n^{4/3})$ bound to $O(n)$? Or are there graphs G with sets S, P that will force us to keep $\Omega(n^{4/3})$ edges?

Graph reachability is almost as basic of a notion as directed graphs themselves. It is ubiquitous in math, science, and technology. Computational questions related to graph reachability are central to various fields. For example, the classical NL vs. L open question asks if one can find a directed path using small space. We would arguably be in a much better shape for tackling all the fundamental questions involving reachability if we could give good answers to basic structural questions like the one above.

*amir.abboud@gmail.com. Work done while at Stanford University.
†gbodwin@mit.edu

1.1 A New Extremal Upper Bound Our main positive result is the following theorem, which improves on all previously known upper bounds by polynomial factors. It was known that all distances can be preserved (not just reachability) with $O(\min\{n^{2/3}|P|, n|P|^{1/2}\})$ edges [10, 21]. For the parameters above, those bounds do not beat the $n^{4/3}$ bound – yet ours show that $O(n)$ edges are sufficient. In general, our theorem states that whenever $|P| = o(n|S|)$, one can do much better than keeping spanning trees out of every source.

Theorem 1.1. *For any graph $G = (V, E)$ on n nodes, set of sources $S \subseteq V$, and set of pairs $P \subseteq S \times V$, there is a subgraph H of G with $O(n + \sqrt{n \cdot |P| \cdot |S|})$ edges that preserves the reachability of all pairs in P. That is, all pairs in P are connected by a path in H iff they are in G.*

It is interesting to compare our bounds for these *reachability preservers* to the ones known for *distance* preservers in *undirected* graphs. If we fix n, $|P| = p$, and $|S| = s$, which one should be sparser, in the extremal sense? On the one hand, reachability is a much easier requirement than distance. On the other hand, directed graphs can be much more difficult to handle than undirected graphs.

Combining Theorem 1.1 with previous results, we obtain an *unconditional separation* between the two, asserting that reachability preservers are extremally sparser than distance preservers, at least in some range of parameters. Consider the setting where $s = n^{1/3}$ and $p = n^{2/3}$. Theorem 1.1 implies that there will always be a reachability preserver with $O(n)$ edges, while the lower bounds of Coppersmith and Elkin [21] show that n^{1+c} edges are sometimes necessary (for some absolute $c > 0$) to preserve distances, even in undirected graphs.

An ostensible drawback of the proof of Theorem 1.1 is that it is non-constructive: while we prove *existence* of reachability preservers below our claimed sparsity threshold, the proof does not suggest a method for computing them efficiently (we remark that there are trivial algorithms that run in undesirably large polynomial time). We overcome this problem – even in a highly generalized setting – by showing the following complimentary algorithmic result:

Theorem 1.2. *Let $f(n, |P|, |S|)$ be a function such that every n-node graph $G = (V, E)$ and pair set $P \subseteq S \times V$ has a subgraph on at most $f(n, |P|, |S|)$ edges that preserves the reachability of all pairs in P. Then there is a randomized algorithm that always returns a reachability preserver of any $G, P \subseteq S \times V$ on $O(f(n, |P|, |S|))$ edges, and terminates in $\widetilde{O}(|E||S|)$ time with high probability.*

Thus the preservers promised by Theorem 1.1 can indeed be computed efficiently. Moreover, if any of our reachability preserver bounds are later improved by follow-up work, this algorithm immediately implies that the new preservers will be efficiently constructible, even if the new proof is non-constructive. Our algorithm utilizes known data structures for decremental single source reachability [37, 43, 49] and crucially relies on a parallelization trick.

1.2 Approximation Algorithms: The Directed Steiner Network Problem Let us consider the setting where we have a graph on n nodes with a set of pairs $P \subseteq S \times V$ for which we care about preserving the reachability, with parameters $|S| = n^{3/4}, |P| = n^{5/4}$ so that Theorem 1.1 guarantees the existence of a preserver on $O(n^{1.5})$ edges. Moreover, our algorithm can efficiently find such a preserver. But what if we could get an even better reachability preserver – say, $O(n)$ size? It is easy to observe that, in *the worst case* over all graphs $\Omega(|P|) = \Omega(n^{5/4})$ edges could be necessary, e.g. if the graph is a biclique. However, from a real-world point of view, why should we expect our graphs to be worst case? It could be that our particular graph and sets P, S enjoys a reachability preserver on much smaller size than what the extremal results guarantee. Denote the number of edges in the sparsest possible reachability preserver of our given graph by OPT. Are there efficient algorithms that can find a reachability preserver with density close to OPT?

This question is the most basic out of the many "Steiner" problems in directed graphs. Steiner-type problems are a central topic of study in combinatorial optimization. Perhaps the most well-known such problem is the Steiner Tree problem in undirected graphs, from Karp's original NP-complete problems: Given a weighted undirected graph G and a set of terminals $T \subseteq V(G)$ return a minimum weight subgraph H in which all the terminals are connected. A constant factor approximation algorithm for Steiner Tree is a mainstream topic in advanced algorithms courses. In directed graphs, Steiner-type problems become much harder to approximate. Perhaps the most natural and well-studied version is the Directed Steiner Network problem (DSN), also known as Directed Steiner Forest.

Definition 1. (Directed Steiner Network) *Given a weighted directed graph $G = (V, E)$ with nonnegative weights on the edges $w : E \to \mathbb{N}$ and a set of k pairs $P \subseteq V \times V$, find the subgraph H of* *minimum total weight* $\sum_{e \in E(H)} w(e)$ *such that for all pairs* $(s,t) \in P$ *there exists a path from* s *to* t *in* H.

Dozens of generalizations and special cases of DSN have been studied in the literature. We refer the reader to the survey of Kortsarz and Nutov [40]. It arises naturally when we have to satisfy certain connectivity demands at the lowest possible cost.

There is a long history of approximation algorithms for DSN. Charikar et al. [14] gave an $\tilde{O}(k^{2/3})$ approximation, where $k = |P|$, which was later improved by Chekuri, Even, Gupta, and Segev [15] to $\tilde{O}(k^{1/2+\varepsilon})$, where they introduced the influential notion of *junction trees*. Since k could be $\Omega(n^2)$, none of these algorithms achieve a sublinear in n approximation factor. The first sublinear algorithm was achieved by Feldman, Kortsarz, and Nutov [30] who achieved an $O(n^{4/5+\varepsilon})$ approximation (for any $\varepsilon > 0$). Most recently, Berman, Bhattacharyya, Makarychev, Raskhodnikova, and Yaroslavtsev [8] reduced the approximation factor to $O(n^{2/3+\varepsilon})$. The most fundamental case of DSN, which captures the essence of its difficulty, is when all the weights are the same, or equivalently, if the graph is unweighted (the UDSN problem). In a recent breakthrough, Chlamtac, Dinitz, Kortsarz, and Laekhanukit [18] achieved a better approximation factor of $O(n^{3/5+\varepsilon})$ for UDSN. On the negative side, it is quasi-NP-hard to approximate UDSN to within $2^{\log^{1-\varepsilon} n}$ for all $\varepsilon > 0$ [25].

Tying this back to the discussion in the beginning of this subsection, regarding extremal bounds versus approximation algorithms: The algorithm of Chlamtac et al. is guaranteed to find a sparsifier that has $O(n^{3/5+\varepsilon} \cdot OPT)$ edges, which could potentially be much less than our extremal bounds. Perhaps surprisingly, this difference between extremal upper bounds and approximation algorithms does not stop us from applying Theorem 1.1 in a rather simple way to break beyond the $n^{3/5}$ bound achieved by Chlamtac et al.

Theorem 1.3. *For all* $\varepsilon > 0$*, there is a polynomial time algorithm for the Directed Steiner Network problem in unweighted graphs with approximation factor* $O(n^{3/5-1/45+\varepsilon}) = O(n^{0.5777+\varepsilon})$.

We believe that our approach for improving these bounds will have further consequences for approximation algorithms and beyond. The previous algorithms use a procedure that attempts to connect a pair set P at a low cost, under the assumption that the pairs in P have many paths between them (called "thick" pairs; the "thin" pairs are handled using Linear Programming.) To do this, the algorithm randomly samples a small subset of the nodes S that is guaranteed to intersect at least one path for each pair in P, with high probability, and then it connects all nodes appearing in P to-and-from each node in S. All previous papers that follow this approach for Steiner-type problems (e.g. [30, 8, 18, 24]) upper bound the cost of this step by $O(n|S|)$, and our improvement comes from applying the upper bound of Theorem 1.1 instead. Since such hitting-set arguments are ubiquitous in algorithm design, we envision that our approach will have further use.

Our new approximation algorithm is probably not the final say on this fundamental problem; rather, it is a proof of concept that approximation algorithms can benefit from extremal results. Notably, all previous progress on this problem [30, 8, 18, 24] has come from better rounding and analysis of the complicated LP hammers, instead of tackling the simple extremal question about reachability preservers.

It is natural to ask: how far can our approach be pushed? A natural bound to hope for, suggested by Feldman et al. is $O(\sqrt{n})$ approximation: this would match the algorithm of Gupta et al. for Steiner-Network in *undirected* graphs [36], and undirected graphs seem better understood. Our approach would get an $O(\sqrt{n})$ approximation for UDSN, if we can get a positive answer to the fundamental extremal question, which we address in the next subsection: *Are linear size reachability preservers always possible?*

1.3 Linear Size Reachability Preservers

Recall that the upper bound of Theorem 1.1 was $O(n + \sqrt{n \cdot |P| \cdot |S|})$. Perhaps fewer edges are always sufficient? Most optimistically:

Does any n*-node graph and set of node pairs* P *admit a subgraph on* $O(n + |P|)$ *edges that preserves the reachability of all pairs in* P*?*

Note that this is certainly possible in undirected graphs via a spanning tree. In the case of distance preservers in undirected graphs, the possibility of such linear size distance preservers was refuted by Coppersmith and Elkin [21], and the construction for refutation has been crucial to the resolution of longstanding open questions in the field of spanners [2, 3].

One approach is to try to adapt the lower bounds for distances. This is challenging; the lower bounds are based on a construction of a graph on the integer lattice and a large subset of pairs P such that each pair in P has a unique shortest path, all these paths are edge-disjoint, and they are long. The density of the construction comes from the disjointness and length of these paths. The lower bound for distance preservers follows from the uniqueness of these shortest paths: removing any edge will have to increase the distance by +1. On the other hand, for reachability preservers, we do not care if paths increase by +1 or +100 or even $+n$, as long as a path still exists. Indeed, the Coppersmith and Elkin distance preserver lower bound instances admit linear size reachability preservers.

One can apply known gap amplification techniques to increase this gap from +1 to $+n$, such as simple *layering* or the recently introduced *obstacle product* framework [2, 3]. However, this only results in weak lower bounds that can rule out linear size preservers for a restricted range of parameters, and that are far from the upper bound in Theorem 1.1.

The most technical result in this paper is an almost matching lower bound to Theorem 1.1, refuting the possibility of linear size reachability preservers.

THEOREM 1.4. *For any desired p and $s = O(n^{1/3})$ (possibly depending on n) satisfying $ps = \omega(n)$, there is an infinite family of n-node graphs G and sets P of p node pairs, with $P \subseteq S \times V$ for some set S of s nodes, such that any subgraph of G that preservers the reachability for all pairs in P must have $\omega(n)$ edges.*

The starting point for our construction are the same integer lattices of [21], but we take our construction in a different direction. While in [21] the edges are simply defined by the convex hull of points in the ball of radius r away from the node, our choice of edges is much more delicate. We only allow edges that correspond to vectors in certain restricted cones within the ball, which allows us to have much more control over the structure of paths in the graph. In particular, we show that leaving one edge out from a path in our set P will force us to take a detour that is so long that we will have to "exit" the relevant piece of the grid. The full argument is quite lengthy and includes more ingredients from discrete geometry. A more detailed overview of the proof will be given in the next section.

An intriguing open question is to connect extremal results and approximation algorithms in another direction: Can we use our constructions of hard graphs to improve the inapproximability bounds for DSN?

Discussion. To highlight the tightness of our bounds, let us present what we consider the most gratifying corollary of this paper: for any choice of $|S|$ and $|P|$ that someone gives us and asks us whether for this particular pair of parameters, an extremal linear size bound of $O(n)$ is possible, our two theorems provide a confident and precise answer on whether the answer is positive or negative!

Of course, our results above are specific to the setting where $P \subseteq S \times V$, and are not as tight in other interesting settings (e.g. when the restriction $P \subseteq S \times V$ is dropped) that we discuss in the next subsection. Let us argue why our setting, in which we parametrize by $|S|$, is natural and important.

First, this is the relevant setting for our applications to approximation algorithms for Steiner-type problems. Like in our algorithm above, it is common for one to sample a set of nodes S that "hit" all paths with certain properties.

And second, we like to think of our Theorem 1.1 for the $S \times V$ setting as a generalization of the notion of single-source reachability trees (such as BFS or DFS). When $P = S \times V$, the naïve $O(n|S|)$ bound for reachability preservers is tight and it follows from the easy fact that a single-source reachability tree has $O(n)$ edges. Our results offer a generalization for the $P \subsetneq S \times V$ setting: when $|P| = \Theta(n|S|)$ our Theorem 1.1 still gives the correct $O(n|S|)$ bound, but when $|P| = o(n|S|)$ it offers a new bound that is non-trivially improved. Our lower bound suggests that our "generalization" might be the qualitatively right one, since it is tight in the two extreme settings: when P is as large ($P = S \times V$), there is a simple $\Omega(n|S|)$ lower bound (a biclique) establishing tightness, and when P is small, the lower bound of Theorem 1.4 suggests that we have correctly captured the settings in which $O(n)$ size preservers are possible. Moreover, note that we also obtain a nice generalization in terms of the running time for computing these structures. The standard way to build a reachability preserver in the "large P" ($P = S \times V$) case is with a BFS/DFS search, which takes $O(|E||S|)$ time. Our (more involved) algorithm in Theorem 1.2 achieves essentially the same runtime, while achieving the sparser structure guaranteed by Theorem 1.1.

1.4 Reachability Preservers and Related Objects

In slightly more general terms, the object we study can be defined as follows:

DEFINITION 2. *For a graph $G = (V, E)$ and a pair-set $P \subseteq V \times V$, a reachability preserver $H = (V, E'), E' \subseteq E$ is a subgraph of G that preservers the reachability of all pairs in P. That is, for all pairs $(s, t) \in P$ the subgraph H contains a path from s to t if and only if G contains one.*

The general extremal question is: If G has n nodes, and P contains p pairs, what sparsity can we guarantee for the sparsest reachability preserver of G, P? This problem has been implicitly studied before, as it is a more basic version of many extensively studied graph sparsification and compression problems in theoretical computer science. A *distance preserver* of a graph G and pair set P is a sparse subgraph that preserves all *distances* of pairs in P [12, 21, 11, 10, 1, 2]. A *pairwise spanner* must preserve all distances of pairs in P *approximately* [53, 23, 39, 45, 38, 2]. A *distance preserving minor* is a small minor of G that preservers all distances in P approximately [35, 13, 27, 5, 29, 41, 17, 34, 33, 42, 16, 31].

Other notions of sparsification for directed graphs have been studied. A roundtrip spanner is a sparse subgraph in which all pairwise roundtrip distances (u to v plus v to u) are approximately preserved [22, 50, 44]. Very recently, there has been progress on spectral sparsifiers of directed graphs [20]. Perhaps most related to ours are the *Transitive Closure Spanners* [9, 48] in which one also tries to preserve the reachability among pairs of nodes. However, the main objective there is to have a spanner with small diameter (by possibly adding edges to the graph) rather than make it sparse.

In the special case of $P = \{s\} \times V$, there has been exciting recent progress in the fault-tolerant setting [47, 46, 6, 19] which essentially studied the following question: Given a graph G and a source s, what is the sparsest subgraph H such that for all nodes in v there are at least k node (or edge) disjoint paths in H iff there are in G. The questions we study are the special case of $k = 1$, but we consider more than one source. A related question for planar graphs was studied by Thorup [52] in his seminal work on distance oracles. There is also a lot of recent interest in *terminal embeddings* where one tries to embed from one metric to another while approximately preserving the distances of a given set of terminals (see [28] and the references therein).

We remark that an alternative way to ask the extremal question is as follows. What is the densest graph that you can construct if you have n nodes and you get to add p paths, all of them starting from a set of sources of size s, such that every path is the unique path between its endpoints? (The extremal equivalence between this problem and the reachability preserver problem is slightly nontrivial, but can be shown.)

Our results above essentially settle the case of $P \subseteq S \times V$. For the more general case of arbitrary $P \subseteq V \times V$ we get the following bounds, which improve by polynomial factors both the upper and lower bounds that were known from previous work. The known upper bound for the more demanding problem of directed distance preservers is $O(\min\{n^{2/3}|P|, n|P|^{1/2}\})$ edges [10, 21]. There was no non-trivial lower bound known for reachability preservers.

THEOREM 1.5. *Given any n-node graph $G = (V, E)$ and pair set $P \subseteq V \times V$, there is a reachability preserver of G, P on $O(n + (n|P|)^{2/3})$ edges.*

For any integer $d \geq 2$, for any $p = p(n)$, there exists an infinite family of graphs $G = (V, E)$ and sets of node pairs $P \subseteq V \times V$ of size $|P| = O(p)$ such that every reachability preserver H of G, P has

$$\Omega\left(n^{2/(d+1)} p^{(d-1)/d}\right)$$

edges.

The lower bound part of this theorem follows as a corollary of a lower bound proved by Coppersmith and Elkin [21]. To obtain the best possible lower bound from this theorem, one must choose the setting of the parameter d that maximizes the lower bound for the particular pair set size $p = p(n)$ being considered. For example, in the range $p = O(n)$, the lower bound is optimized at $\Omega\left(n^{2/3}p^{1/2}\right)$ by setting $d = 2$. We note that this implies that $O(n)$-size reachability preservers are not possible in general if $p = \omega(n^{2/3})$; however, the upper bound portion of the above theorem only implies that they *are* possible when $p = O(n^{1/2})$. We consider this non-tightness to be an interesting open question.

New ideas seem to be required to close the embarrassing gaps in our understanding of this basic setting. Another particularly interesting setting that remains wide open is the possibility of linear size preservers under the $P = S \times T$ restriction.

OPEN QUESTION 1. *Can we always preserve the reachability among a set of pairs $S \times T$ in a graph on n nodes with $O(n + |S| \cdot |T|)$ edges?*

2 Reachability Preservers and Technical Overview

The main focus of this section is on proving Theorems 1.1 and 1.2. We split this section into three parts. First, we (non-constructively) prove that reachability preservers as promised in Theorem 1.1 always exist. Next, we complement our proof with an algorithm that constructs existentially optimal reachability preservers of a given instance $G = (V, E), P \subseteq S \times V$ in $O(|E||S| \log n)$ time. In other words, if every $G, P \subseteq S \times V$ has a reachability preserver on $f(n, |P|, |S|)$ edges, then our algorithm builds a reachability preserver on $O(f(n, |P|, |S|))$ edges for any given G, P. This algorithm is the opposite of our non-constructive existential proof, in the sense that it is "purely constructive:" we have existential optimality for the output graph produced by the algorithm, but the algorithm itself does not suggest what the right existential bound should be.

In the last part of this section, we give a brief overview of our results and techniques for our other existential bounds on reachability preservers. These proofs are more complicated and technically involved, so we defer full proofs to Section 4, giving only a flavor of them here.

2.1 An Existential Proof of Theorem 1.1 Let $H = (V, E)$ be the sparsest possible reachability preserver of some input graph and pair set $G = (V, E'), P \subseteq S \times V$ where $E \subseteq E'$. For each $(s,t) \in P$, arbitrarily choose some canonical $s \rightsquigarrow t$[1] path in H and denote this path by $\pi(s,t)$; we may clearly assume that each $\pi(s,t)$ is acyclic. Additionally, for every edge $e \in E$ there must be some pair $(s,t) \in P$ such that $s \not\rightsquigarrow t$ in $H \setminus \{e\}$; else we could safely delete the edge e from H without changing its salient reachability properties, thus obtaining a sparser reachability preserver of G, P. We may thus assign ownership of each edge e to some pair (s,t) with this property. Let D be the average in-degree of H. Say that an edge (u,v) is *light* if the in-degree of v is at most $D/2+1$, or *heavy* otherwise. Denote by $E^H_{(s,t)}$ the set of heavy edges owned by the pair $(s,t) \in P$.

Claim 1.

$$\left|E^H_{(s,t)}\right| \leq \frac{2|S|}{D} \quad \textit{for all } (s,t) \in P$$

Proof. Suppose towards a contradiction that

$$\left|E^H_{(s,t)}\right| > \frac{2|S|}{D}$$

for some $(s,t) \in P$. Let

$$F_{(s,t)} := \left\{(a,b) \in E \ \mid \ \text{there is an edge } (a',b) \in E^H_{(s,t)}, a \neq a'\right\}.$$

For any heavy edge $(a',b) \in E^H_{(s,t)}$ on the path $\pi(s,t)$, there are at least $D/2+1$ other edges (a,b) that are incoming to b, and all such edges belong to $F_{(s,t)}$. We then have

$$\left|F_{(s,t)}\right| \geq \left(\frac{D}{2}\right)\left|E^H_{(s,t)}\right| > \left(\frac{D}{2}\right)\left(\frac{2|S|}{D}\right) = |S|.$$

So $\left|F_{(s,t)}\right| \geq |S|+1$, and by the pigeonhole principle, there are two distinct edges $f_1 := (a_1,b_1) \neq f_2 :=$

[1]We use the standard $s \rightsquigarrow t$ notation throughout this paper to mean that there exists a directed path from s to t. Similarly, $s \not\rightsquigarrow t$ means that no such path exists.

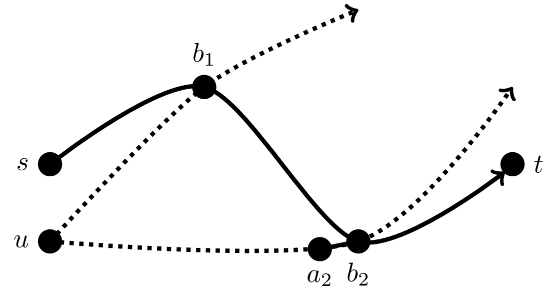

Figure 1: In the proof of Claim 1, we show that if $\left|E^H_{(s,t)}\right|$ is too large, then there are two paths leaving a common source u that intersect $\pi(s,t)$ at two different nodes b_1, b_2; additionally, the latter path owns the incoming edge (a_2, b_2). This is used to derive a contradiction.

$(a_2, b_2) \in F_{(s,t)}$ that are owned by pairs from a common source; that is,

$$f_1 \in E^H_{(u,v_1)} \quad \text{and} \quad f_2 \in E^H_{(u,v_2)}$$

for some $u \in S$ and $(u,v_1),(u,v_2) \in P$. We also assume without loss of generality that b_1 precedes b_2 in $\pi(s,t)$. See Figure 1 for a picture of what our paths must look like if these constraints are all satisfied.

We now argue a contradiction as follows. Since the pair (u,v_2) owns the edge (a_2,b_2), every $u \rightsquigarrow b_2$ path in H must include the edge (a_2,b_2). One possible $u \rightsquigarrow b_2$ path can be found by joining the prefix $\pi(u,v_1)[u \rightsquigarrow b_1]$ with the suffix $\pi(s,t)[b_1 \rightsquigarrow b_2]$. Thus $(a_2,b_2) \in \pi(s,t)$. Since $\pi(s,t)$ is acyclic it only contains one edge entering b_2. Since $(a_2,b_2) \in F_{(s,t)}$, it follows by definition of $F_{(s,t)}$ that (s,t) owns its edge entering b_2, so (a_2,b_2) is owned by (s,t). However, we note that $F_{(s,t)}$ is disjoint from $E^H_{(s,t)}$ by its definition, since for any edge $(a,b) \in F_{(s,t)}$ there is a different edge $(a',b) \in E^H_{(s,t)}$ and (by acyclicty of $\pi(s,t)$) $E^H_{(s,t)}$ only contains at most one edge entering any given node. So this implies that $(a_2,b_2) \notin F_{(s,t)}$. We thus have a contradiction and the claim follows.

At most $n\left(\frac{D}{2}+1\right)$ edges are light. If $D \geq 3$ then this is a constant fraction of the total number of edges, and we may now complete the proof of Theorem 1.1 by some straightforward algebra. We have

$$\begin{aligned}|E| &= O\left(\sum_{(s,t)\in P} \left|E^H_{(s,t)}\right|\right)\\ &= O\left(\sum_{(s,t)\in P} \frac{|S|}{D}\right)\\ &= O\left(\frac{|S||P|n}{|E|}\right)\end{aligned}$$

and so

$$\begin{aligned}|E|^2 &= O\left(|S||P|n\right)\\ |E| &= O\left(\sqrt{|S||P|n}\right).\end{aligned}$$

Alternately, if $D \le 3$ then by definition of D the graph has $O(n)$ edges. Putting these together, we have

$$|E| = O\left(n + \sqrt{|S||P|n}\right)$$

as claimed.

2.2 Constructing Reachability Preservers Here we observe that one can construct asymptotically existentially optimal reachability preservers in $O(|E| \cdot |S| \log n)$ time. Specifically:

THEOREM 2.1. *Suppose that every n-node graph $G = (V, E)$ and set $P \subseteq S \times V$ of node pairs (for some $S \subseteq V$) has a reachability preserver on at most $f(n, |P|, |S|)$ edges. Then there is a randomized algorithm that always constructs a reachability preserver of any input G, P on $O(f(n, |P|, |S|))$ edges and terminates in time $O(|E||S| \log n)$ with high probability.*

Let $G = (V, E), P \subseteq S \times V$ be an n-node graph and pair set taken on input. We assume that $|E| = \Omega(n)$ and that P is nonempty, since otherwise we may return G or the empty graph $(V, \emptyset)$, respectively.

Step 1: Contracting Cycles. The first step is to convert G to a DAG. To accomplish this, we run an algorithm to detect the strongly connected components of G in $O(|E|)$ time (e.g. [51]), and then we build a graph $G' = (V', E')$ whose nodes are the strongly connected components of H and which has directed edges (C_1, C_2) iff there is a directed edge (c_1, c_2) in G where $c_1 \in C_1$ and $c_2 \in C_2$. Additionally, for each pair $(s, t) \in P$, we have a pair (C_s, C_t) in our new pair set P' (where C_s, C_t are the strongly connected components holding s, t respectively). We will then compute an existentially optimal preserver of G', P', and then complete the construction by "uncontracting" the strongly connected components of G. That is, for each strongly connected component in G we include any sparse directed skeleton of the component that preserves its all-pairs strong reachability (it is easy to see that these skeletons for each strongly connected component cost only $O(n)$ edges in total), and then for each edge $(C_1, C_2) \in E'$ we arbitrarily choose representative nodes $c_1 \in C_1, c_2 \in C_2$ and include the edge (c_1, c_2) in our final preserver. The total cost is thus

$$O(n) + O(f(n', |P'|, |S'|));$$

since $n' \le n, |P'| \le |P|, |S'| \le |S|$ and f is clearly weakly increasing in all of its parameters, the total cost is then

$$O(n) + O(f(n, |P|, |S|))$$

which is asymptotically optimal (since clearly $f(n, |P|, |S|) = \Omega(n)$). It now "only" remains to compute an existentially optimal preserver of G', P', which is a DAG. For simplicity of notation we will drop the primes, and simply assume that G itself is a DAG.

Step 2: Building the Reachability Preserver. We now construct our reachability preserver of G, P decrementally; that is, we initially set $H \leftarrow G$ and we will iteratively delete edges of H. We use as a subroutine an algorithm of Italiano [37].

THEOREM 2.2. ([37]) *There is a deterministic algorithm that, given a DAG $G = (V, E)$ and a source node $s \in V$, explicitly maintains the set of nodes reachable from s over a sequence of edge deletions. The total amount of time needed to maintain this list is $O(|E|)$.*

For the sake of building intuition, we first consider Algorithm 2.1, which is perhaps the most natural method for sparsifying H while preserving its salient reachability properties (this is **not** the final algorithm that we use).

ALGORITHM 2.1. (WARMUP)

1. Initialize $H \leftarrow G$
2. For each $s \in S$:
   (a) Initialize a data structure D_s as in Theorem 2.2
3. While H has more than $f(n, |P|, |S|)$ edges remaining:
   (a) Choose an edge e still in H uniformly at random

(b) Delete e from H and update each data structure D_s accordingly, and record the changes made to each D_s during this process

(c) Let SUCCESSFUL $\leftarrow \bigwedge_{(s,t)\in P} s \overset{?}{\rightsquigarrow} t$

(d) If not SUCCESSFUL:

i. Add e back to H and undo the changes made to all data structures D_s

4. Return H

It is clear that Algorithm 2.1 is correct, in the sense that it eventually terminates within the sparsity bound $f(n, |P|, |S|)$ (since, by definition of f, at all times there exists a subgraph of the current graph H on $f(n, |P|, |S|)$ edges that is a reachability preserver of G, P). The trouble is that the runtime of Algorithm 2.1 is not very good. The successful iterations are not a problem: by Theorem 2.2 they take $O(|E||S|)$ time in total, which is good. However, with high probably have at *least* $f(n, |P|, |S|)$ unsuccessful iterations (possibly far more), and *each* of these might take $\Omega(|E||S|)$ time. That is, because the *worst case update time per deletion* in Theorem 2.2 is still $O(|E|)$, it is conceivable that we will pay $\Omega(|E||S|)$ work for a single unsuccessful deletion, but then we have to unwind all of this work and so we are not able to amortize this work over the runtime of the entire algorithm.

This failed attempt gives us the intuition that we are willing to perform some extra work in order to avoid unsuccessful iterations. The key insight here is that *parallelization* is useful. In particular, our final algorithm (Algorithm 2.2) works by maintaining $\Theta(\log n)$ different "universes" at a time, and it runs each loop through Algorithm 2.1 *simultaneously in all universes*. This is our key idea: we progress the computation in each universe in alternating steps so that none lags behind the others.

ALGORITHM 2.2. (FINAL)

1. Initialize $H \leftarrow G$

2. For each source $s \in S$:

   (a) Initialize $100 \log n$ identical data structures D_s^i from Theorem 2.2 ($i \in [100 \log n]$)

3. While H has more than $2f(n, |P|, |S|)$ edges remaining:

   (a) Let R be a random sample of $100 \log n$ edges still in H

   (b) For each edge $r_i \in R$ **in parallel:**

   i. Update the data structures D_s^i with the deletion of r_i for each $s \in S$

   ii. Let SUCCESSFUL$_i \leftarrow \bigwedge_{(s,t)\in P} s \overset{?}{\rightsquigarrow}_{H\setminus\{r_i\}} t$

   iii. If SUCCESSFUL:

   A. For each $j \neq i \in [100 \log n]$:

   - Halt the parallel process corresponding to the edge r_i
   - For each $s \in S$:

   – Unwind the updates to D_s^j during this parallel process

   – Update D_s^j by deleting r_i

4. Return H

The argument proceeds as follows.

Proof. [Proof of Theorem 2.1] We claim that Algorithm 2.2 terminates in $O(|E||S| \log n)$ time with high probability. As in the intuition given above, we say that an *iteration* of Algorithm 2.2 is a single round of computation through its main while loop, that each $i \in [100 \log n]$ represents a *universe*, and that a universe is *successful* in a given iteration if SUCCESSFUL$_i$ is set to true (or if it would be set to true, if the parallel computation in this universe were allowed to run to completion).

We first note that, with high probability, at least one of the i universes will be successful in each iteration. This holds because at least half of the remaining edges can be successfully deleted in each iteration (since H has more than $2f(n, |P|, |S|)$ edges) and we choose $\Theta(\log n)$ of these edges independently at random. We thus assume that some universe is successful in each iteration, and we let i be the successful universe whose data structure updates $\mathcal{D}_s^i$ are completed the fastest. Let t_i be the total amount of time taken by these updates. It follows that the total work done over all universes in this iteration is $O(t_i \log n)$, since each of $\Theta(\log n)$ universes performs at most $O(t_i)$ work, then unwinds it, and then reupdates based on the deletion of the edge r_i. Here, it is important to note that SUCCESSFUL$_i$ does not need to be computed explicitly (which would take $O(|P| \log n)$ time per iteration, possibly exceeding our claimed runtime); since the data structure from Theorem 2.2 explicitly maintains a reachable list, we may simply check each deletion as it occurs to see if it destroys reachability of a pair in P. That is, whenever the data structure updates its explicit output to indicate that some pair is no longer reachable, we check whether this pair belongs to P.

Let L be the total number of loops through the main while loop executed by Algorithm 2.2. Its total runtime is then

$$\sum_{1\leq j\leq L} O(t_i^j \log n) = O(\log n)\cdot \sum_{1\leq j\leq L} t_i^j$$

(where t_i^j is the value of t_i in iteration j). Note that

$$\sum_{1\leq j\leq L} t_i^j = O(m|S|)$$

since the left-hand side describes the runtime needed to maintain $|S|$ data structures from Theorem 2.2 over the sequence of edge deletions $\{r_i^j\}$ (without any rewinding). The runtime follows.

2.3 Overview of Remaining Existential Results

Matching Lower Bounds for Theorem 1.1. An interesting implication of Theorem 1.1 is that any $G, P \subseteq S\times V$ admits a reachability preserver on $O(n)$ edges whenever $|P|\cdot|S| = O(n)$. Clearly this $O(n)$ can't be improved to $o(n)$, due to the trivial lower bound of a path. However, it is conceivable that this could be improved along another dimension: maybe $O(n)$ size reachability preservers *still* always exist even in some broader range where $|P|\cdot|S| = \omega(n)$?

Our most important additional result is a refutation of this possibility in a broad range of values for $|P|, |S|$, establishing the optimality of Theorem 1.1 in the regime of linear-size reachability preservers. We show:

THEOREM 2.3. *For any desired $p = p_n, s = O(n^{1/3})$ with $p\cdot s = \omega(n)$, there is an infinite family of n-node graphs $G = (V,E)$ and sets $P \subseteq S\times V$ of $|P| = p$ node pairs (for some $S \subseteq V$ with $|S| = s$) such that every reachability preserver G, P has $\omega(n)$ edges.*

The proof of Theorem 2.3 is very involved and it constitutes one of the main technical contributions of this paper. The high-level idea is as follows. We build a graph whose nodes are represented as points in the integer lattice $\mathbb{Z}^2$, arranged in a long thin rectangle. We design P by choosing certain pairs of points (s,t) at either end of the rectangle. We then choose the edges of the graph using a fairly complicated method that we forgo in this overview. The key properties of our choice of edges are that (1) there is a unique shortest path from s to t, and (2) every single edge in the graph, viewed as a vector in Euclidean space, has a very large projection onto the vector $\vec{st}$. We then argue that the shortest $s \rightsquigarrow t$ path is in fact the *only* $s \rightsquigarrow t$ path in the graph. This holds because any alternate $s \rightsquigarrow t$ path must use at least one more edge than the shortest path; however, since each edge in this path still makes considerable Euclidean progress in the desired $\vec{st}$ direction, we argue that such a path will necessarily overshoot t. Thus no edges in the unique shortest $s \rightsquigarrow t$ path may be removed from G; by covering the edges of G with a small number of $s \rightsquigarrow t$ paths, we get our claimed sparsity bound. The last step in the proof uses some more involved arguments from discrete geometry to make the proof work with a pair set of the form $P \subseteq S \times V$.

The formalisms of this proof involve some careful trigonometric arguments alongside a high-dimensional parameter balance, since there is an implicit tension between the density of the graph and the worst-case projection of one edge onto another. However, with lots of care this can indeed be accomplished, yielding Theorem 2.3.

Upper Bounds in the General Pairwise Setting. In Theorem 1.1, we write $P \subseteq S\times V$ and allow our bounds to depend on the parameter $|S|$. It is also natural to consider the settting where no such guarantee is made, and we may only parametrize our upper bounds by n and $|P|$. In this setting, we show:

THEOREM 2.4. *Every n-node graph G and set P of node pairs has a reachability preserver on $O\left(n + (n|P|)^{2/3}\right)$ edges.*

The proof bears some similarity to that of Theorem 1.1, in the sense it also follows from the observation that we may choose paths $\pi(s,t)$ for pairs $(s,t)\in P$ in such a way that no three paths form a "triangle." However, it differs from Theorem 1.1 in that the density bound ultimately follows from a partition of pairs in P into families, where the common trait of a family is: there is a path $\pi(s,t)$ for some pair $(s,t)\in P$ such that every pair in the family admits a path between its endpoints that intersects $\pi(s,t)$. We then "batch process" a family at a time, which we show can be done using only $O(n)$ edges. This differs from the old approach of bounding the contribution of one path at a time (as in the proof of Theorem 1.1). This new and coarser-grained view of the problem leads to our claimed existential bound.

For the sake of marking the current state of the art of reachability preservers, we also remark that the following result is implied directly from a corresponding theorem for distance preservers in [10]:

THEOREM 2.5. ([10]) *Let $\mathrm{RS}(n)$ be the largest value such that every graph $G = (V,E)$ whose edge set can be partitioned into n induced matchings has $O\left(\frac{n^2}{\mathrm{RS}(n)}\right)$ edges. Then all G, P has a reachability preserver on $O(|P|)$ edges whenever $|P| = \Omega\left(\frac{n^2}{\mathrm{RS}(n)}\right)$.*

It is known that $2^{\Omega(\log^* n)} \leq \text{RS}(n) \leq 2^{O(\sqrt{\log n})}$ [32, 26, 7], so this result improves on the trivial statement that $O(|P|)$ edges suffice when $|P| = \Omega(n^2)$. Thus it also improves on Theorem 2.4 in a narrow regime of sufficiently large $|P|$. We omit the proof here, as it is no different from the one known for distance preservers.

Lower Bounds in the General Pairwise Setting. Finally, we observe that distance preserver lower bounds can be converted into reachability preserver lower bounds of [21], yielding:

THEOREM 2.6. *For any integer $d \geq 2$, there is an infinite family of n-node graphs and pair sets P for which every reachability preserver of G, P has*

$$\Omega\left(n^{2/(d+1)}|P|^{(d-1)/d}\right)$$

edges.

The basic idea behind this theorem is simple: one can take an lower bound graph for unweighted distance preservers and "layer" it by copying the base graph many times and directing edges along these copies. This yields a sparser graph, but it now functions as a reachability lower bound, since for any non-shortest path $\pi(s,t)$ in the original graph G for a pair $(s,t) \in P$, the corresponding path $\pi(s,t)$ in the layered version of G will overshoot its "destination layer." Hence, one merely needs to demand reachability to enforce that the edges of $\pi(s,t)$ are kept in the preserver.

Informally, it seems at first that the number of nodes in G will increase by a factor of L (where L is the length of the paths $\pi(s,t)$) when it is layered, while the edge count of G will not increase at all. Here, we introdue a trick: with a more careful layering setup, it is possible to ensure that the edge count of G also increases by a factor of L. This is still a penalty over the original distance preserver lower bound graph, but it is not nearly as bad as the naive approach to layering.

We note the following immediate consequence of Theorems 2.4 and 2.6:

COROLLARY 2.1. *Any n-node graph G and set P of $|P| = O(\sqrt{n})$ node pairs has a reachability preserver of size $O(n)$. This size bound on P could conceivably be improved up to $|P| = O(n^{2/3})$, but no further.*

We consider it a very interesting open problem to close this gap.

3 Directed Steiner Network

In this section we obtain new approximation algorithms for a classical network connectivity problem with a long history of prior work. Our algorithms builds on these works by identifying an ingredient that is common to all of them, and we show how it can be improved using our results on extremal graph sparsification.

The algorithm of Chlamatac et al. for UDSN performs two different algorithms, depending on whether $OPT \geq n^{4/5}$ or $OPT < n^{4/5}$, and in each case it achieves a factor $k = n^{3/5+\varepsilon}$ approximation. We show that using an extremal bound on the worst case density of reachability preservers, and for some constant $b = 1/45$, we can get a factor $k = n^{3/5-b+\varepsilon}$ approximation whenever $OPT \geq n^{4/5-3b}$. Then, we use the same procedure of previous work for the case of small OPT, but with a smaller upper bound on how large OPT could be, and get a k-approximation for that case as well. For this latter procedure we use the following lemma from previous work.

LEMMA 3.1. (FOLLOWS FROM [18, 8]) *If in an UDSN instance $OPT \leq O(n^{4/5-\alpha})$, for some $0 \leq \alpha < 4/5$, then for all $\varepsilon > 0$ we can get a k-approximation to OPT where $k \leq O(n^{3/5-\alpha/3+\varepsilon})$ in polynomial time.*

Now let us assume that OPT is at least $\Omega(n^{4/5-\alpha})$. We pick a threshold k and say that a pair $(s,t) \in P$ is *k-thick* if the set of all s-to-t paths in G contains at least k nodes, and otherwise the pair is *k-thin*.

The following lemma shows that all thin pairs can be handled with a k-factor approximation. The proof relies on considering an LP relaxation of the problem, and then performing a randomized rounding strategy to pick an approximate integral solution.

LEMMA 3.2. (FOLLOWS FROM [8]) *For all $k \geq 1$, given an instance of UDSN we can find a subgraph on $\tilde{O}(k \cdot OPT)$ edges, in which all k-thin pairs are connected with high probability.*

This allows us to focus on k-thick pairs. All previous works for DSN and related problems [30, 8, 18, 24] where this thin/thick pairs framework was used, handled the thick pairs in an extremely naive strategy: they sample a hitting set S of $\tilde{O}(n/k)$ nodes, and try to connect every terminal in the pair set P to every node in S. For instance, Chlamatac et al. take BFS trees in and out of each node in the hitting set. In their algorithm, k is set to $n^{3/5}$ and so their hitting set has size $\tilde{O}(n^{2/5})$, which makes the cost of this stage $\tilde{O}(n^{7/5})$.

But do we really need $O(n^{7/5})$ edges in order to connect all the terminals to the hitting set? This is where our work comes in: we use the extremal results to argue that fewer edges are always sufficient. For example, say that OPT is $n^{4/5}$ and that we have $n^{4/5}$ terminals that we want to connect to $n^{2/5}$ other nodes. Our Theorem 1.1 says that $O(n^{13/10})$ edges are sufficient, as opposed to the naive bound of $n^{14/10}$.

More concretely, set $k = O(n^{3/5-\alpha/3})$ and chose a hitting set S of size $\tilde{O}(n^{2/5+\alpha/3})$. Let T be the set of all terminals participating in k-thick pairs in P. We know that $|T| \leq OPT$ since any solution must keep at least one edge adjacent to each terminal in P. Now our goal is to connect all nodes in T to and from all nodes in S (if possible), that is, we consider the pair set $P' = \{(x, y) \in (S \times T) \cup (T \times S) \mid \text{there is a path from } x \text{ to } y \text{ in } G\}$, and ask for a Reachability Preserver in G for P'. Since our pair set has the $P' \subseteq S \times T$ structure, our Theorem 1.1 gives us an upper bound of $O(\sqrt{n|S|^2|T|})$ on the number of edges necessary, which can be upper bounded by

$$O\left(\sqrt{n \cdot (n^{2/5+\alpha/3})^2 \cdot OPT}\right) = O\left(\frac{n^{9/10+\alpha/3}}{\sqrt{OPT}}\right) \cdot OPT = O(n^{1/2+5\alpha/6}) \cdot OPT,$$

where the last step follows because $OPT = \Omega(n^{4/5-\alpha})$. By choosing $\alpha \leq 3/45$ we get that this bound is smaller than $k \cdot OPT$, since $1/2 + 5\alpha/6 \leq 3/5 - \alpha/3$.

Lemma 3.3. (new) *For all $0 \leq \alpha \leq 3/45$, if in an UDSN instance $OPT \geq \Omega(n^{4/5-\alpha})$ then we can get a k-approximation to OPT where $k \leq O(n^{3/5-\alpha/3+\varepsilon})$ in polynomial time.*

Finally, we can run both algorithms for small and large OPT, and return the sparser solution. This gives our new approximation algorithm for UDSN which breaks the $n^{3/5}$ barrier.

Theorem 3.1. *For any fixed constant $\varepsilon > 0$, there is a polynomial time algorithm for UDSN with approximation factor $O(n^{3/5-1/45+\varepsilon}) = O(n^{0.5777+\varepsilon})$.*

4 Extremal Bounds for Reachability Preservers

Here, we formally prove the extremal results for reachability preservers overviewed in Section 2.3.

4.1 Lower Bounds in the $P \subseteq S \times V$ Setting

We now prove Theorem 2.3. We remark that we have only concerned ourselves in this proof with establishing the best possible lower bound of the form $\omega(n)$; we have not tried to optimize (or even compute) the quality of the lower bound at superlinear preserver sizes. This is because the lower bound is quite complicated in its current state, and these optimizations would introduce considerable additional complexity that we believe would distract more than it adds.

Geometric Setting and Definitions. Let R be a rectangle in $\mathbb{R}^2$ with width w and height h (we will set these parameters later, and have $w \ll h$). We do *not* align R with the axes of $\mathbb{R}^2$; rather, it is rotated such that the angle between the x-axis and the long side of the rectangle is a parameter ψ that we will choose later. Its shift in space is unimportant (i.e. it doesn't matter where the bottom-left point of the rectangle is placed). The *start zone* Z of R is defined as the nested sub-rectangle of R with width w and height $h_z \ll w$, positioned such that one of the w-length sides of Z is also a w-length side of R.

The following geometric definitions will be useful: $CH(s, r)$ is the convex hull of the set of points in R within distance r of lattice node s, and $\mathcal{C}_{s,\phi}$ denote the cone with apex s and internal line of symmetry positioned parallel to the h-length side of R and directed away from the start zone Z (we will interpret $\mathcal{C}_{s,\phi}$ to include its interior).

See Figure 2 for a picture of all of the above definitions (as well as a picture of a pair (s, t) included in the pair set of the construction; this process is described below).

Construction of the Lower Bound Instance.

- The nodes of G are precisely the points in the integer lattice $\mathbb{Z}^d$ in the interior of R. In an abuse of notation, throughout this construction we will use the names of nodes interchangeably with their vectors; for example, given nodes u, v we write $v - u$ to denote the vector in $\mathbb{R}^2$ between them. In particular, we will commonly write $\|u - v\|_2$ to denote the Euclidean distance between the vectors u, v; by contrast, we will exclusively use the notation $\text{dist}(u, v)$ to mean the shortest path distance in G from the node u to the node v. We will also try to make this distinction clear in context.

- The edges of G are defined: for each node s, we add a directed edge to each node $t \in CH(s, r) \cap \mathcal{C}_{s,\phi} \cap R$ (here ϕ is a new parameter of the construction that we will choose later).

- The pairs of P are defined as follows. For each $s \in Z$ and each edge (s, a) leaving s, we let $t = s + k(a - s)$ (interpreting the points here as vectors in $\mathbb{R}^2$), where k is the largest integer such that $t \in R$. We then include the pair (s, t) in P. By symmetry of the construction, note that there is a $s \rightsquigarrow t$ path of length k obtained by starting at s repeatedly stepping in the direction

$a - s$ until one reaches t. We shall call this the *canonical path* for the pair (s, t), and we denote it by $\pi(s, t)$.

One critical step in the construction remains: we will later add additional nodes S to the graph, connecting each $s \in S$ to the start point of many pairs $p \in P$, thus yielding the desired $P \subseteq S \times V$ property for the construction. However, many technical details need to be stated before this step can be appropriately de-mystified. Thus, for now, our goal is simply to argue that G, P requires an $\omega(n)$ size reachability preserver, and we will revisit this step later.

Density of the Construction. First, we analyze the choice of parameters necessary to ensure that our lower bound construction has a superlinear number of edges.

THEOREM 4.1. ([4]) $|CH(s, r)| = \Theta\left(r^{2/3}\right)$. *Moreover, all points $x \in CH(s, r)$ satisfy*

$$r - \Theta(r^{-1/3}) \leq \|(s, x)\|_2 \leq r$$

for some sufficiently large absolute constant hidden in the Θ.

LEMMA 4.1. *Suppose that the skew ψ is chosen uniformly at random. If $\phi = c_1 r^{-2/3}$, then in expectation, we have $CH(u, r) \cap C_{u,\phi} = c_2$ (note that this intersection includes nodes outside R) for any u and for some c_2 that can be made arbitrarily large by choice of c_1.*

Proof. Let $B(s, r)$ denote the r-ball in Euclidean space centered at the node s. Note that $B(s, r) \cap \mathcal{C}_{s,\phi}$ gives a shape whose area is $\Theta(\phi)$ times the area of $B(s, r)$. Since we have $\Theta(r^{2/3})$ points in $CH(s, r)$, which have essentially been randomly rotated by the skew ψ, the expected number of these points contained in $\mathcal{C}(s, \phi)$ is

$$\Theta\left(\phi \cdot r^{2/3}\right) = \Theta\left(c_1 r^{-2/3} \cdot r^{2/3}\right) = \Theta(c_1) =: c_2.$$

We now set ψ to any value such that obtains $\deg(u) = c_2$. This lemma yields our first parameter constraint: **we will ultimately set $\phi = \boldsymbol{\Theta}\left(\mathbf{r}^{-2/3}\right)$**. This parameter setting is assumed in the proofs that follow.

Analysis of Canonical Paths. Our next goal is to enforce that each canonical path is the *unique* path in G between its endpoints. Uniqueness of these paths is not immediate from the construction; rather, we take on some constraints on the construction parameters that must be satisfied in order for this desired uniqueness property to hold.

Let us fix attention on some canonical path with endpoints (s, z), and suppose this path contains k edges. The following notion will be useful in this part of the argument:

DEFINITION 3. (PROGRESS) *The* progress *of a node w is defined as*

$$progress(w) := \|\underset{z-s}{proj}\,(z - w)\|_2.$$

Less formally, $progress(w)$ is the distance along the $s \to z$ direction one has traveled so far in Euclidean space if one is currently sitting at the node w.

We have:

LEMMA 4.2. (SIMILAR TO AN ARGUMENT IN [21]) *The path $\pi(s, z)$ is the unique* shortest *(s, z) path in G.*

Proof. First, by construction, if (w, x) is any edge on the canonical path $\pi(s, z)$ then the number of edges the number of edges in $\pi(s, z)$ is

$$k = \frac{\|z - s\|_2}{\|x - w\|}.$$

On the other hand, we can compare vector projections

$$\|\underset{x-w}{proj}(x' - w)\|_2 < \|\underset{x-w}{proj}(x - w)\|_2$$

from the fact that $x, x' \in CH(w, r)$. Since $x - w = \lambda(z - s)$ for some scalar λ, and the operation of vector projection is sensitive only to the direction but not the magnitude of the base vector, we then have

$$\|\underset{z-s}{proj}(x' - w)\|_2 < \|\underset{z-s}{proj}(x - w)\|_2.$$

Thus, considering a non-canonical $s \rightsquigarrow z$ path including the edge (w, x'), the progress of its k^{th} (or less) node a is

$$progress(a) < k \cdot \|x - w\|_2 = \|z - s\|_2.$$

Thus $a \neq z$ (since $progress(z) = \|z - s\|_2$), so the length of the non-canonical path is strictly greater than k.

We have just shown a lower bound on the number of edges in any non-canonical $s \rightsquigarrow z$ path. We next show an upper bound on the same quantity:

LEMMA 4.3. *If $k = O(r^{4/3})$ with a sufficiently small implicit constant in the O, then any (s, z) path in G contains at most k edges.*

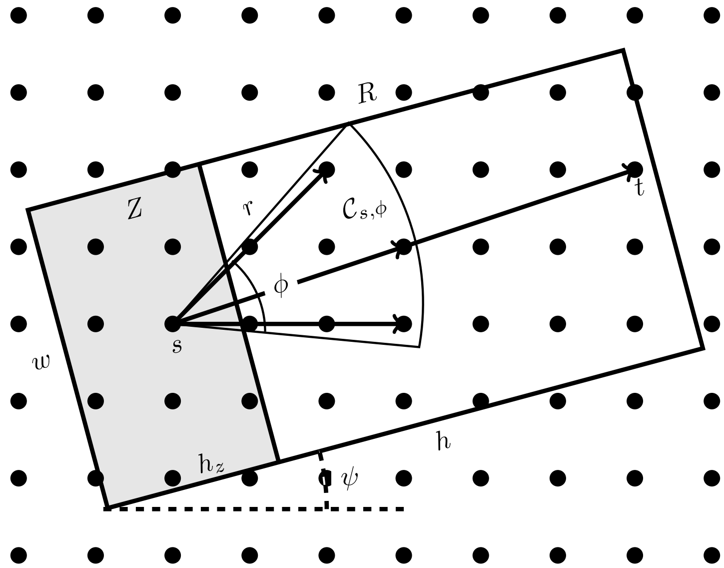

Figure 2: The (partial) lower bound construction. The nodes of G are the integer lattice points inside the entire rectangle R. All edges incident to s have been pictured; the edges leaving all other nodes are generated in the same way (the edge is omitted if the other endpoint leaves R). One of the pairs in P will be (s, t). The canonical path $\pi(s, t)$ is two edges long and is pictured here. The second step of the construction, in which source nodes are added, has not been described yet and is not pictured here.

Proof. Given any two edges $(u, v), (u, v')$, the length of the Euclidean projection of one onto the other is

$$\|\underset{v-u}{proj}(v' - u)\|_2 \geq \|v' - u\| \cdot \cos(\phi)$$

(using the standard formula for vector projection), since by construction the angle between these vectors is at most ϕ. Using the standard small-angle approximation $\cos(\phi) = 1 - \Theta(\phi^2)$, observing that $\|v' - u\| \geq r - \Theta(r^{-1/3})$ (from Theorem 4.1), and substituting in our previous parameter setting $\phi = \Theta(r^{-2/3})$ we can write

$$\|\underset{v-u}{proj}(v' - u)\|_2 \geq (r - \Theta(r^{-1/3})) \cdot (1 - r^{-4/3})$$
$$= r - \Theta(r^{-1/3}).$$

Thus

$$\|v - u\|_2 - \|\underset{v-u}{proj}(v' - u)\|_2 = O(r^{-1/3})$$

from the above inequality and the fact that $\|proj_{v-u}(v' - u)\|_2 < \|v - u\|_2 \leq r$.

Let us now consider the value of $progress(x)$ for any node x that is the endpoint of an $s \rightsquigarrow x$ path containing $k+1$ or more edges. We may lower bound:

$$progress(x) \geq (k + 1)(\|v - u\|_2 - O(r^{-1/3})).$$

Substituting in the parameter setting $k = O(r^{4/3})$, we may write

$$progress(x) \geq k\|v - u\|_2 + (\|v - u\|_2 - O(r)).$$

Since we also have (loosely) $\|v - u\|_2 > r - 1$, we may choose our implicit constants small enough that $(\|v - u\|_2 - O(r)$ is positive. Thus

$$progress(x) > k\|v - u\|_2 = progress(z)$$

and so $x \neq z$. Thus z is not the endpoint of any path starting at s containing $k + 1$ or more edges.

This gives our next parameter setting: **we will set** r **such that** $k = O(r^{4/3})$ **for all canonical paths** (with an implicit constant small enough to push Lemma 4.3 through). Combining the previous two lemmas, we have

Lemma 4.4. *The canonical* (s, z) *path is the unique* (s, z) *path in* G.

Proof. The canonical path has k edges, and since it is the unique shortest path by Lemma 4.2, any alternate path has $k+1$ edges or more. But by Lemma 4.3 and our setting of r, there is no $s \rightsquigarrow z$ path with $k + 1$ edges or more. Thus $\pi(s, z)$ is the unique $s \rightsquigarrow z$ path.

Pairwise Lower Bound Quality. We have:

Lemma 4.5. *If* $w = c_4 h r^{-2/3}$ *for a large enough constant* c_4*, then any reachability preserver of* G, P *contains at least a constant fraction of the edges in* G.

Proof. Fix a node u, and look at the set of edges (v, u) with endpoint u. Say that an edge is *left-leaning* if the angle of the vector $u - v$ with the x-axis exceeds ψ (i.e. it points more towards the "top" of the rectangle than the bottom), or *right-leaning* otherwise. By construction, a constant fraction of the edges incident to u are left- or right-leaning. Since no edge in a canonical path may be deleted in a reachability preserver, it then suffices to show that for each node u, either all left- or all right-leaning edges are contained in a canonical path.

Let us suppose that there exists a left-leaning edge (v, u) that is not part of any canonical path (otherwise we are done). It follows from the construction that the line $\ell_{u,v}$ in Euclidean space through v, u does not intersect any node in the start zone Z. More specifically, since (v, u) is left-leaning, we must have that $\ell_{u,v}$ misses the start zone to its right-hand side. Thus, by standard trigonometry, the Euclidean distance from u to its closest point on the long side of the rectangle R (of length h) is at most $h \tan \phi$. Substituting in the small angle approximation $\tan \phi = \Theta(\phi)$ and the parameter setting $\phi = \Theta(r^{-2/3})$, the distance is $\Theta(hr^{-2/3})$.

By a symmetric argument, if (v, u) is right-leaning then the distance from u to the closest point on the left-hand side of R is also $\Theta(hr^{-2/3})$. Note, however, that the sum of the distance from u to the closest points on the left- and right-hand sides of R is precisely the parameter w. Thus, if we have $w = \Omega(hr^{-2/3})$ with a sufficiently large implicit constant, then either every left-leaning edge ending at u is part of a canonical path, or every right-leaning edge ending at u is part of a canonical path. The lemma follows.

This gives our third parameter constraint: **we will set $\mathbf{w = \Theta\left(hr^{-2/3}\right)}$**.

Let us recap the progress made so far in the argument. We have proved that our construction is a good lower bound against $O(n)$-size *pairwise* reachability preservers: by setting the parameters of the construction within the constraints specified so far, we have that:

- $CH(u, r) \cap \mathcal{C}_{u,\phi} = c_2$ for some constant c_2 that we can make arbitrarily large,
- It is easy to see that for a constant fraction of the nodes u in G, a constant fraction of the nodes in $CH(u, r) \cap \mathcal{C}_{u,\phi}$; thus the density of G is nc'_2 for some constant c'_2 that we can make arbitrarily large, and
- A constant fraction of the edges in G belong to canonical paths and thus may not be removed in a reachability preserver of G, P. Therefore any reachability preserver has density nc''_2 for some constant c''_2 that we can make arbitrarily large.

Intuitively, this argument lets us refute the general possibility of reachability preserver constructions of $|P| = \omega(n^{2/3})$ pairs on $O(n)$ edges (because, as we shall see later, a setting of $|P| = O(n^{2/3})$ is compatible with the parameter restrictions we have made so far). Specifically, by pushing the implicit constant in the $|P| = O(n^{2/3})$ arbitrarily high, we may push the constant c''_2 in the density lower bound arbitrarily high until it exceeds whatever implicit constant is used in a claimed $O(n)$ sparsity upper bound.

What remains is to modify the construction so that the pair set P only uses a small set S of sources. Currently, every node in the start zone appears in a pair in P, and this is far too many.

Augmenting the Construction with Source Nodes. We will now add "source nodes" S to the graph by the following process. Choose a representative node $u \in Z$ for which all edges in $CH(u, r) \cap \mathcal{C}_{u,\phi}$ lie in R. For each edge (u, v), define the vector $a_v \in \mathbb{R}^2$ by the following process: let $(u, v_L), (u, v_R)$ be the nodes immediately to the left (counterclockwise) and right (clockwise), respectively, from (u, v) in the plane. Define $a_v := v_L - v_R$. For the two left- and right-extreme values of v with no suitable v_L, v_R (respectively), we define a_v by temporarily increasing ϕ so that $CH(u, r) \cap \mathcal{C}_{u,\phi}$ includes a few additional edges on either side; we use these edges to define a_v and then restore ϕ to its usual value and discard them.

For each possible a_v and any given node $x \in Z$, we define the line $\ell^x_{a_v}$ in Euclidean space in the direction a_v passing through x. Let L be the set of all such lines. For each $\ell \in L$, we add a node $s_\ell \in S$, and for each node $x \in Z$ on the line ℓ we add an edge (s_ℓ, x) to the graph. For each pair of the form $(x, z) \in P$, we replace it with the pair (s_ℓ, z). The set of all such nodes s_ℓ is denoted by S. One should not think of the nodes in S as being vectors in the plane; they are abstract.

This completes the construction. Two tasks remain in this part of the proof: first we will confirm that we still have unique pairs in P, and then we will count the size of S.

LEMMA 4.6. *Assuming $h_z = O(r)$ with a sufficiently small implicit constant, the unique path in G for a new pair $(s, z) \in P$ that replaced an old pair (x, z) is found by first walking the edge (s, x) and then walking the canonical path $\pi(x, z)$.*

Proof. The goal here is to generalize Lemmas 4.2 and

4.3 to apply the same upper- and lower-bounds on any path of the form $x' \rightsquigarrow z$, where $x \neq x'$ lie on the same $\ell \in L$. This will imply that no path $x' \rightsquigarrow z$ exists, and thus the unique $s_\ell \rightsquigarrow z$ path in G uses $\pi(x, z)$ as a subpath, and so the density analysis given above still applies. We once again let k be the number of edges in $\pi(x, z)$.

The upper bound argument, showing that any $x' \rightsquigarrow z$ path has length k or less, is nearly identical to the one given before, and we will not restate it fully. There is only one additional piece to the argument, which is the (only) place we use the restriction $h_z = O(r)$. Since we have $h_z = O(r)$, it is immediate that the progress of the node x' is at least (say) $r/2$, assuming the constant in the $O(r)$ is chosen sufficiently small. Thus, after walking any one edge from x' our progress (measured with respect to $z - x$) is $\Omega(r)$ (assuming h_z is a sufficiently small constant fraction of r). The rest of the proof can then continue as before: intuitively, the same calculations show that after taking $k + 1$ or more steps from x', the total progress made exceeds $\|z - x\|_2$ and so no path of length $k + 1$ or more ends at the node z.

The lower bound argument, needs significant new ideas. Observe that some of the nodes on ℓ *start* with positive progress, which means that arguments based on upper bounding total progress, as before, will break. Instead, we argue:

Let $k \cdot CH(z, r)$ denote the vectors in $\mathbb{Z}^2$ (not necessarily in R) of the form $z + k \cdot \alpha$ where $\alpha \in CH(z, r)$. It is clear that these vectors are a convex set in $\mathbb{Z}^2$, and that their convex hull encloses any node y for which there is a $y \rightsquigarrow z$ path in G of k steps or fewer. We observe that ℓ is a supporting line of this convex point set; that is, it intersects exactly one point (x) in $k \cdot CH(z, r)$ and all other points in $CH(z, r)$ lie strictly on one side of ℓ. It is immediate that $x \in \ell$ and it is demonstrated by the existence of $\pi(x, z)$ that $x \in CH(z, r)$. By standard structure of convex sets, it suffices to show that x_L, x_R lie (strictly) on the same side of ℓ, where x_L, x_R are the points in $CH(z, r)$ immediately to the left and right of x. Recall from the construction that the direction of ℓ is the vector a_v, and in fact we have $k \cdot a_v = x_L - x_R$. Thus the slope of a_v is strictly between the slopes of $x_L - x$ and $x - x_R$, and so ℓ is a supporting line of $k \cdot CH(z, r)$. From this, it is immediate that there is no $x' \rightsquigarrow z$ path containing k or fewer edges where $x' \in \ell$ (unless $x' = x$). Hence any $s \rightsquigarrow z$ path must first walk the edge (s, x), and from there the unique $x \rightsquigarrow z$ path is $\pi(x, z)$ by Lemma 4.4.

We now work towards counting the size of S.

Lemma 4.7. *For any line $\ell \in L$, if the Euclidean length of the line segment $\ell \cap Z$ is d, then the number of nodes in Z intersected by ℓ is $\Omega(dr^{-1/3})$.*

Proof. The line ℓ has some direction a_v. Recall that this direction was obtained as $a_v = v_L - v_R$, where v_L, v_R are both endpoints of edges starting at some node u. Since the angle between $v_L - u$ and $v_R - u$ is at most $\phi = \Theta(r^{-2/3})$, and their magnitudes are $\|v_L - u\|_2, \|v_R - u\|_2 \in [r - 1, r]$, some straightforward trigonometry gives that

$$\|v_L - v_R\|_2 = \|a_v\|_2 = O(r^{1/3}).$$

Since $v_L - u, v_R - u \in \mathbb{Z}^2$, it follows that given any point $x \in \mathbb{Z}^2 \cap \ell$, each time we add or subtract a_v we find another such point in $\mathbb{Z}^2 \cap \ell$. Since $\|a_v\|_2 = r^{1/3}$, it follows that any segment of ℓ of length d contains $\Omega(dr^{-1/3})$ points in $\mathbb{Z}^2$. Each point in $\ell \cap Z \cap \mathbb{Z}^2$ is a node in Z, and the lemma follows.

Lemma 4.8. *Assuming $\phi = O\left(\frac{h_z}{w}\right)$, we have*

$$|S| = O(r^{1/3} h_z).$$

Proof. Partition L into families $\{F_i\}$ of parallel lines; note that there are $O(1)$ such families since there are $O(1)$ edges on any node u and thus $O(1)$ different directions a_v that determine lines. It thus suffices to bound the size of each parallel family F_i as $|F_i| = O(r^{1/3} h_z)$.

For a given F_i, first note that each node in Z appears on exactly one line $\ell \in F_i$. Note that the angle between ℓ and the short w-length side of R is in the interval $[-\phi, \phi]$. Thus, by straightforward trigonometry, if $\tan(\phi) = O(h_z/w)$, then the average over $\ell \in F_i$ of the Euclidean length of the line segment $\ell \cap Z$ will be $\Omega(w)$. Thus, by Lemma 4.7, the average $\ell \in F_i$ holds $\Omega(w \cdot r^{-1/3})$ nodes. Since there are $O(wh_z)$ nodes in total and each node lies on one such line, we then have

$$|F_i| = O\left(\frac{wh_z}{wr^{-1/3}}\right) = O\left(r^{1/3} h_z\right)$$

as desired.

Balancing Parameters. If we choose our parameters within the constraints specified so far, we thus have that any reachability preserver $H = (V, E_H)$ of G, P satisfies $|E_H| = \Omega(n \cdot c_2)$ for some constant c_2 that can be made arbitrarily large by choice of other implicit constants. This refutes the possibility of a lower bound of type $O(n)$ for G, P, since the constant c_2 can be pushed high enough to violate the implicit constant in this O. It now remains only to see which values of $|P|, |S|$ can be obtained.

To recap, our constraints are (dropping the first constraint on ϕ, which is no longer used):

1. $\Omega(r) = h = O\left(r^{7/3}\right)$ (lower bound must hold for the graph to be nonempty)
2. $\frac{w}{h} = \Omega\left(r^{-2/3}\right)$ (slightly rearranged)
3. $\frac{h_z}{w} = \Omega\left(r^{-2/3}\right)$

and by construction we have

1. $n = \Theta(wh)$ nodes
2. $|P| = \Theta(|Z|) = \Theta(h_z w)$
3. $|S| = \Theta(h_z r^{1/3})$.

The parameter setting that proves Theorem 2.3 in the densest regime is given by

$$h = n^{7/12}, w = n^{5/12}, h_z = r = n^{1/4}.$$

Straightforward algebra then yields $|P| = \Theta\left(n^{2/3}\right)$ and $|S| = \Theta\left(n^{1/3}\right)$, as desired. Theorem 2.3 is given in its sparsest regime by the parameter setting

$$h = w = h_z = n^{1/2}$$

which gives $|S| = 1$ and $|P| = n$. Since all dependencies are linear, a linear interpolation between these parameter settings proves Theorem 2.3 in general.

4.2 Upper Bounds in the General Pairwise Setting Here we prove Theorem 2.4. As in Theorem 1.1, we may assume that G is a DAG. Let H be a sparsest reachability preserver of G, P. We then have

$$H = \bigcup_{(s,t)\in P} \pi(s,t)$$

where $\pi(s,t)$ is some canonical $s \rightsquigarrow t$ path. We again say that a pair (s,t) *requires* an edge (u,v) if every $s \rightsquigarrow t$ path in H includes the edge (u,v). Since H is minimal we may assign ownership of each edge to a pair that requires it.

As a preprocessing step, while there is a pair $p \in P$ that owns at most $\frac{n^{2/3}}{|P|^{1/3}}$ edges, we delete p from P and we delete all edges that are *uniquely* required by p from the graph. We lose at most $|P| \cdot \frac{n^{2/3}}{|P|^{1/3}} = (n|P|)^{2/3}$ edges in this way. If the number of remaining edges in the graph is $O(n + (n|P|)^{2/3})$ then we are done; otherwise, we proceed as follows.

Let D be the (new) average in-degree of H, and we say that an edge (u,v) is *heavy* if the in-degree of v is at least $D/2+1$, or (u,v) is *light* otherwise. We assume towards a contradiction that $D = \omega\left(1 + \frac{|P|^{2/3}}{n^{1/3}}\right)$. As before, we define

$$E^H_{(s,t)} := \{(u,v) \in \pi(s,t) \ \mid \ (u,v) \text{ is heavy and owned by } (s,t)\}$$

and

$$F_{(s,t)} := \left\{(a,b) \in E \ \mid \ \text{there is an edge } (a',b) \in E^H_{(s,t)}, a \neq a'\right\}.$$

Since at least half of all edges are heavy, there is be a pair (s,t) for which

$$\left|E^H_{(s,t)}\right| = \Theta\left(\frac{n^{2/3}}{|P|^{1/3}}\right)$$

and thus

$$\begin{aligned}\left|F_{(s,t)}\right| &= \Theta\left(D \cdot \frac{n^{2/3}}{|P|^{1/3}}\right)\\ &= \omega\left(\left(1 + \frac{|P|^{2/3}}{n^{1/3}}\right) \cdot \frac{n^{2/3}}{|P|^{1/3}}\right)\\ &= \omega\left(n^{1/3}|P|^{1/3}\right).\end{aligned}$$

It is clear that no two edges in $F_{(s,t)}$ can be required by the same path. Thus, letting $Q_{(s,t)}$ be the set of paths that own an edge in $F_{(s,t)}$, we have $\left|Q_{(s,t)}\right| = \left|F_{(s,t)}\right|$. Let $R_{(s,t)}$ be the set of (all) edges required by some path in $Q_{(s,t)}$. We then have

$$\begin{aligned}\left|R_{(s,t)}\right| &\geq \left|Q_{(s,t)}\right| \cdot \frac{n^{2/3}}{|P|^{1/3}}\\ &= \left|F_{(s,t)}\right| \cdot \frac{n^{2/3}}{|P|^{1/3}}\\ &= \omega\left(n^{1/3}|P|^{1/3}\right) \cdot \frac{n^{2/3}}{|P|^{1/3}}\\ &= \omega(n).\end{aligned}$$

However, following an identical argument to the one given in Theorem 1.1, no two of these edges may share an endpoint. By the pigeonhole principle we have a contradiction, and so $D = O\left(1 + \frac{|P|^{2/3}}{n^{1/3}}\right)$ and so $|E| = nD = O\left(n + (n|P|)^{2/3}\right)$.

4.3 Lower Bounds in the General Pairwise Setting To prove Theorem 2.6, we first need:

THEOREM 4.2. (PROVED IN [21]) *For any integer $d \geq 2$, for any $p = p(n)$, there exists an infinite family of undirected unweighted graphs $G = (V,E)$ on*

$$\Omega\left(n^{2d/(d^2+1)} p^{(d^2-d)/(d^2+1)}\right)$$

edges, as well as sets of node pairs $P \subseteq V \times V$ of size $|P| = O(p)$, such that

- *For each pair $(s,t) \in P$ there is a unique shortest path in G between s and t,*
- *These paths are all edge disjoint, and*
- *The edge set of G is precisely the union of these paths.*

The proof of Theorem 2.6 is by a natural layering transformation of the graphs drawn from Theorem 4.2, with an optimization over comparable techniques in prior work (e.g. [10]) that allows us to squeeze extra "costly" pairs into the lower bound that improve its density.

We construct our graph as follows. Start with an instance $G = (V, E), P$ drawn from Theorem 4.2, with d chosen the same as the desired d in Theorem 2.6. By standard tricks ([10]) we may assume that all pairs $(s,t) \in G$ have the same $\text{dist}(s,t)$; call this common distance L. We now take $2L$ copies of $\{G_1, \dots, G_{2L}\}$ of G, which will serve as layers for our new graph $G' = (V', E')$. For any node $x \in V$, let x_i denote the copy of x in the graph G_i. For each edge $(u,v) \in E$, we add edges (u_i, v_{i+1}) and (v_i, u_{i+1}) to G' (for each $1 \leq i \leq 2L-1$). Finally, for each pair $(s,t) \in P$, we add pairs (u_j, u_{j+L}) for all $1 \leq j \leq L$ to P'. This completes the construction. Note that G' has $n' = \Theta(nL)$ nodes and P' has $O(pL)$ pairs. We now show:

Proof. [Proof of Theorem 2.6] By construction, for each pair $(s_i, t_{i+L}) \in P'$, we observe that there is a unique $s \rightsquigarrow t$ path in G'. This holds because every (s_i, t_{i+L}) path in G' has length exactly L (since all edges are directed from lower-numbered layers to higher-numbered layers) so the corresponding (s,t) path in G has length L; by Theorem 4.2, there is a unique $s \rightsquigarrow t$ path of length L in G. Moreover, we observe that any two of these paths are edge disjoint, since they correspond to shortest paths in G for pairs in P which are edge disjoint. Thus, for each pair $p \in P'$, we may identify a set of $L-1$ unique edges in E' such that any reachability preserver of G', P' must keep all $L-1$ edges. Hence, any reachability preserver of G', P' has $\Omega(|P'|L) = \Omega(pL^2)$ edges.

With this, our lower bound follows from straightforward algebra. We compute:

$$L = \frac{|E(G)|}{p} = \Theta\left(n^{2d/(d^2+1)} p^{d(d-1)/(d^2+1)-1}\right)$$
$$= \Theta\left(n^{2d/(d^2+1)} p^{(-d-1)/(d^2+1)}\right)$$

and so the number of edges $|E'|$ in any reachability preserver of G', P' satisfies

$$\begin{aligned}|E'| &\geq pL^2\\ &= |E|L\\ &= \Omega\left(n^{2d/(d^2+1)} p^{(d^2-d)/(d^2+1)}\right) \cdot \Theta\left(n^{2d/(d^2+1)} p^{(-d-1)/(d^2+1)}\right)\\ &= \Omega\left(n^{4d/(d^2+1)} p^{(d^2-2d-1)/(d^2+1)}\right)\end{aligned}$$

We also have

$$nL = \Theta\left(n^{(d+1)^2/(d^2+1)} p^{(-d-1)/(d^2+1)}\right)$$

and

$$pL = \Theta\left(n^{2d/(d^2+1)} p^{(d^2-d)/(d^2+1)}\right)$$

and so

$$|E'| = \Omega\left((nL)^{2/(d+1)} (pL)^{(d-1)/d}\right)$$
$$|E'| = \Omega\left((n')^{2/(d+1)} (p')^{(d-1)/d}\right)$$

which completes the theorem.

Acknowledgement. We thank the anonymous reviewers for exceptionally helpful comments on the writing of the paper.

The authors were supported by the grants of Virginia Vassilevska Williams: NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338. Part of the work was performed while visiting the Simons Institute for the Theory of Computing, Berkeley, CA.

References

[1] Amir Abboud and Greg Bodwin. Error Amplification for Pairwise Spanner Lower Bounds. In *Proc. of the 27th SODA*, 2016.

[2] Amir Abboud and Greg Bodwin. The 4/3 Additive Spanner Exponent is Tight. In *Proc. of the 48th STOC*, 2016.

[3] Amir Abboud, Greg Bodwin, and Seth Pettie. A hierarchy of lower bounds for sublinear additive spanners. In *Proc. of the 28th SODA*, pages 568–576. SIAM, 2017.

[4] I. Barany and D. G. Larman. The convex hull of the integer points in a large ball. *Mathematische Annalen*, pages (312):167–181, 1998.

[5] Amitabh Basu and Anupam Gupta. Steiner point removal in graph metrics. *Unpublished manuscript,* `http://www.math.ucdavis.edu/~abasu/papers/SPR.pdf`, 2008.

[6] Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault tolerant subgraph for single source reachability: generic and optimal. In *STOC*, pages 509–518, 2016.

[7] Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences of the United States of America*, 32(12):331, 1946.

[8] Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and directed steiner forest. *Inf. Comput.*, 222:93–107, 2013.

[9] Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012.

[10] Greg Bodwin. Linear Size Distance Preservers. In *Proc. of the 28th SODA*, 2017.

[11] Greg Bodwin and Virginia Vassilevska Williams. Better Distance Preservers and Additive Spanners. In *Proc. of the 27th SODA*, 2016.

[12] Béla Bollobás, Don Coppersmith, and Michael Elkin. Sparse distance preservers and additive spanners. In *Proc. of the 14th SODA*, pages 414–423, 2003.

[13] Hubert T.-H. Chan, Donglin Xia, Goran Konjevod, and Andréa W. Richa. A tight lower bound for the steiner point removal problem on trees. In *9th APPROX*, pages 70–81, 2006.

[14] Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. In *SODA*, pages 192–200, 1998.

[15] Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Trans. Algorithms*, 7(2):18, 2011.

[16] Yun Kuen Cheung. Steiner point removal—distant terminals don't (really) bother. *arXiv preprint arXiv:1703.08790*, 2017.

[17] Yun Kuen Cheung, Gramoz Goranci, and Monika Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. In *43rd ICALP*, pages 131:1–131:14, 2016.

[18] Eden Chlamtác, Michael Dinitz, Guy Kortsarz, and Bundit Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. In *Proc. of the 28th SODA*, pages 534–553, 2017.

[19] Keerti Choudhary. An optimal dual fault tolerant reachability oracle. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 130:1–130:13, 2016.

[20] Michael B. Cohen, Jonathan A. 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 *STOC*, pages 410–419, 2017.

[21] Don Coppersmith and Michael Elkin. Sparse Sourcewise and Pairwise Distance Preservers. *SIAM Journal on Discrete Mathematics*, pages 463–501, 2006.

[22] L J Cowen and C G Wagner. Compact roundtrip routing in directed networks. *Journal of Algorithms*, pages 1,50,79–95, 2004.

[23] Marek Cygan, Fabrizio Grandoni, and Telikepalli Kavitha. On Pairwise Spanners. In *Proc. of the 30th STACS*, pages 209–220, 2013.

[24] Michael Dinitz and Zeyu Zhang. Approximating low-stretch spanners. In *SODA*, pages 821–840, 2016.

[25] Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *STOC*, pages 750–759, 1999.

[26] M Elkin. An improved construction of progression-free sets. *Israeli Journal of Math 184*, pages 98–128, 2011.

[27] Michael Elkin, Yuval Emek, Daniel A Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008.

[28] Michael Elkin, Arnold Filtser, and Ofer Neiman. Terminal embeddings. *Theor. Comput. Sci.*, 697:1–36, 2017.

[29] Matthias Englert, Anupam Gupta, Robert Krauthgamer, Harald Racke, Inbal Talgam-Cohen, and Kunal Talwar. Vertex sparsifiers: New results from old techniques. *SIAM Journal on Computing*, 43(4):1239–1262, 2014.

[30] Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for directed steiner forest. *J. Comput. Syst. Sci.*, 78(1):279–292, 2012.

[31] Arnold Filtser. Steiner point removal with distortion $o(\log k)$. *arXiv preprint arXiv:1706.08115*, 2017.

[32] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics 174*, pages 561–579, 2011.

[33] Gramoz Goranci, Monika Henzinger, and Pan Peng. Improved guarantees for vertex sparsification in planar graphs. *CoRR*, abs/1702.01136, 2017.

[34] Gramoz Goranci and Harald Räcke. Vertex sparsification in trees. In *14th WAOA*, pages 103–115, 2016.

[35] Anupam Gupta. Steiner points in tree metrics don't (really) help. In *12th SODA*, pages 220–227, 2001.

[36] Anupam Gupta, MohammadTaghi Hajiaghayi, Viswanath Nagarajan, and R Ravi. Dial a ride from k-forest. *ACM Transactions on Algorithms (TALG)*, 6(2):41, 2010.

[37] Giuseppe F Italiano. Finding paths and deleting edges in directed acyclic graphs. *Information Processing Letters*, 28(1):5–11, 1988.

[38] Telikepalli Kavitha. New Pairwise Spanners. In *Proc. of the 32nd STACS*, pages 513–526, 2015.

[39] Telikepalli Kavitha and Nithin Varma. Small Stretch Pairwise Spanners. In *Proc. of the 40th*

ICALP, pages 601–612, 2013.

[40] Guy Kortsarz and Zeev Nutov. Approximating minimum cost connectivity problems. In *Parameterized complexity and approximation algorithms, 13.12. - 17.12.2009*, 2009.

[41] Robert Krauthgamer, Huy L Nguyen, and Tamar Zondiner. Preserving terminal distances using minors. *SIAM Journal on Discrete Mathematics*, 28(1):127–141, 2014. A preliminary version in *39th ICALP*, 2012.

[42] Robert Krauthgamer and Inbal Rika. Refined vertex sparsifiers of planar graphs. *CoRR*, abs/1702.05951, 2017.

[43] J. Łacki. Improved deterministic algorithms for decremental reachability and strongly connected components. *ACM Transactions on Algorithms*, 9(3):27, 2013.

[44] Jakub Pachocki, Liam Roditty, Aaron Sidford, Roei Tov, and Virginia Vassilevska Williams. Approximating cycles in directed graphs: Fast algorithms for girth and roundtrip spanners. In *SODA, to appear*, 2018.

[45] Merav Parter. Bypassing Erdös' Girth Conjecture: Hybrid Stretch and Sourcewise Spanners. In *Proc. of the 41st ICALP*, pages 608–619, 2014.

[46] Merav Parter. Dual failure resilient bfs structure. In *PODC*, pages 481–490, 2015.

[47] Merav Parter and David Peleg. Sparse fault-tolerant bfs trees. In *ESA*, pages 779–790, 2013.

[48] Sofya Raskhodnikova. Transitive-closure spanners: A survey. In *Property Testing - Current Research and Surveys*, pages 167–196, 2010.

[49] Liam Roditty. Decremental maintenance of strongly connected components. In *SODA*, pages 1143–1150. Society for Industrial and Applied Mathematics, 2013.

[50] Liam Roditty, Mikkel Thorup, and Uri Zwick. Roundtrip spanners and roundtrip routing in directed graphs. *ACM Trans. Algorithms*, page 3(4): Article 29, 2008.

[51] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[52] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, 2004.

[53] David P. Woodruff. Lower bounds for additive spanners, emulators, and more. In *Proc. of the 47th FOCS*, pages 389–398, 2006.