Article:

Goranci, G., Henzinger, M. and Peng, P. orcid.org/0000-0003-2700-5699 (2020) Improved guarantees for vertex sparsification in planar graphs. SIAM Journal on Discrete Mathematics (SIDMA), 34 (1). pp. 130-162. ISSN 0895-4801

https://doi.org/10.1137/17M1163153

IMPROVED GUARANTEES FOR VERTEX SPARSIFICATION IN PLANAR GRAPHS[*]

GRAMOZ GORANCI[†], MONIKA HENZINGER[†], AND PAN PENG[‡]

Abstract. Graph Sparsification aims at compressing large graphs into smaller ones while preserving important characteristics of the input graph. In this work we study Vertex Sparsifiers, i.e., sparsifiers whose goal is to reduce the number of vertices. We focus on the following notions:

(1) Given a digraph $G = (V, E)$ and terminal vertices $K \subset V$ with $|K| = k$, a (vertex) reachability sparsifier of G is a digraph $H = (V_H, E_H)$, $K \subset V_H$ that preserves all reachability information among terminal pairs. Let $|V_H|$ denote the size of H. In this work we introduce the notion of reachability-preserving minors (RPMs) , i.e., we require H to be a minor of G. We show any directed graph G admits an RPM H of size $O(k^3)$, and if G is planar, then the size of H improves to $O(k^2 \log k)$. We complement our upper-bound by showing that there exists an infinite family of grids such that any RPM must have $\Omega(k^2)$ vertices.

(2) Given a weighted undirected graph $G = (V, E)$ and terminal vertices K with $|K| = k$, an exact (vertex) cut sparsifier of G is a graph H with $K \subset V_H$ that preserves the value of minimum-cuts separating any bipartition of K. We show that planar graphs with all the k terminals lying on the same face admit exact cut sparsifiers of size $O(k^2)$ that are also planar. Our result extends to flow and distance sparsifiers. It improves the previous best-known bound of $O(k^2 2^{2k})$ for cut and flow sparsifiers by an exponential factor, and matches an $\Omega(k^2)$ lower-bound for this class of graphs.

Key words. reachability-preserving minor, vertex sparsification, planar graphs, cut sparsifiers

AMS subject classifications. 05C10, 05C83, 05C85

1. Introduction.

Very large graphs or networks are ubiquitous nowadays, from social networks to information networks. One natural and effective way of processing and analyzing such graphs is to compress or sparsify the graph into a smaller one that well preserves certain properties of the original graph. Such a sparsification can be obtained by reducing the number of *edges*. Typical examples include cut sparsifiers [8], spectral sparsifiers [52], spanners [57] and transitive reductions [5], which are subgraphs defined on the same vertex set of the original graph G while having much smaller number of edges and still well preserving the cut structure, spectral properties, pairwise distances and transitive closure of G, respectively. Another way of performing sparsification is by reducing the number of *vertices*, which is most appealing when only the properties among a subset of vertices (which are called *terminals*) are of interest (see e.g., [50, 6, 40]). We call such small graphs *vertex sparsifiers* of the original graph. In this paper, we will particularly focus on vertex reachability sparsifiers for *directed* graphs and cut (and other related) sparsifiers for *undirected* graphs.

Vertex reachability sparsifiers in directed graphs is an important and fundamental notion in Graph Sparsification, which has been implicitly studied in the dynamic graph

[*]Submitted to the editors December 27, 2017. A preliminary version of this manuscript appeared in proceedings of the 25th Annual European Symposium on Algorithms (ESA) 2017 [30]. The current version contains all of the missing proofs and improves the size guarantees of RPMs over those in the conference version. The article is rearranged to highlight the main results.

Funding: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 340506.

[†]University of Vienna, Faculty of Computer Science, Währinger Straße 29, 1090, Vienna, Austria. (gramoz.goranci@univie.ac.at, monika.henzinger@univie.ac.at).

[‡]Department of Computer Science, University of Sheffield, UK. p.peng@sheffield.ac.uk. Work done in part while at the Faculty of Computer Science, University of Vienna, Austria.

algorithms community [53, 24], and explicitly in [37]. Specifically, given a digraph $G = (V, E)$, $K \subset V$, a digraph $H = (V_H, E_H)$, $K \subset V_H$ is a (*vertex*) *reachability sparsifier* of G if for any $x, x' \in K$, there is a directed path from x to x' in H iff there is a directed path from x to x' in G. If $|K| = k$, we call the digraph G a *k-terminal digraph*. Note that any k-terminal digraph G always admits a trivial reachability vertex sparsifier H, which corresponds to the transitive closure restricted to the terminals. In this work, we initiate the study of *reachability-preserving minors (RPMs)*, i.e., vertex reachability sparsifiers with H required to be a minor[1] of G. The restriction on H being a minor of G is desirable as it makes sure that H is structurally similar to G, e.g., any minor of a planar graph remains planar. We ask the question whether general graphs admit reachability-preserving minors whose size can be bounded independently of the input graph G, and study it from both the lower- and upper-bound perspective.

For the notion of cut (and other related) sparsifiers, we are given a capacitated undirected graph $G = (V, E, c)$, and a set of terminals K and our goal is to find a (capacitated undirected) graph $H = (V_H, E_H, c_H)$ with as few vertices as possible and $K \subseteq V_H$ such that the quantities like, cut value, multi-commodity flow and distance among terminal vertices in H are the same as or close to the corresponding quantities in G. If $|K| = k$, we call the graph G a *k-terminal graph*. We say H is a *quality-q* (*vertex*) *cut sparsifier* of G, if for every bipartition $(U, K \setminus U)$ of the terminal set K, the value of the minimum cut separating U from $K \setminus U$ in G is within a factor of q of the value of minimum cut separating U from $K \setminus U$ in H. If H is a quality-1 cut sparsifier, then it will be also called a *mimicking network* [33]. Similarly, we define flow and distance sparsifiers that (approximately) preserve multicommodity flows and distances among terminal pairs, respectively (see Section 6 for formal definitions). These type of sparsifiers have proven useful in approximation algorithms [50] and also find applications in network routing [20].

1.1. Our Results.

Reachability Sparsifiers. Our first main contribution is the study of reachability-preserving minors. Although reachability is a weaker requirement in comparison to shortest path distances, directed graphs are usually much more cumbersome to deal with from the perspective of graph sparsification. Surprisingly, we show that general digraphs admit reachability-preserving minors with $O(k^3)$ vertices, which is in contrast to the bound of $O(k^4)$ on the size of distance-preserving minors in undirected graphs by Krauthgamer et al. [40].

THEOREM 1.1. *Given a k-terminal digraph G, there is a reachability-preserving minor H of G with size $O(k^3)$.*

The above bound improves over the size of RPMs for general digraphs in the conference version [30] of this paper by a factor of k. We remark that the above minor H can be constructed in polynomial (in the size of graph G) time. It might be interesting to compare the above result with the lower bound for the construction of a relevant notion called *reachability preserver*. Given a directed graph G, and a terminal set K in G, a reachability preserver[2] of G with respect to K is defined to be a *subgraph* of G that preserves the reachability of all pairs in $K \times K$ [21, 9, 2].

[1] In this paper, a directed graph H is called a *minor* of another directed graph G if H can be formed from G by deleting edges and vertices and by contracting edges, as if they were undirected.

[2] In [21, 2], the reachability preserver is actually defined for any *vertex pair-set* P, while we are only considering the special case that $P = K \times K$.

Bodwin [9] (see Theorem 4.2 therein) implicitly showed that for any integer $d \geq 2$ and $k = k(n)$, there is a family of unweighted graphs $G = (V, E)$ with n vertices and sets K of k nodes in G such that any reachability preserver of G with respect to K has $\Omega(n^{2d/(d^2+1)} k^{(2d-1)(d-1)/(d^2+1)} 2^{-\Theta(\sqrt{\log n \log\log n})})$ edges.

Furthermore, by exploiting a tight integration of our techniques with the compact distance oracles for planar graphs by Thorup [56], we prove the following theorem regarding the size of reachability-preserving minors for planar digraphs[3].

THEOREM 1.2. *Given a k-terminal planar digraph G, there exists a reachability-preserving minor H of G with size $O(k^2 \log k)$.*

The above bound improves over the size of RPMs of planar digraphs in the conference version [30] of this paper by a factor of $\log k$. We complement the above result by showing that there exist instances where the above upper-bound is tight up to a $O(\log k)$ factor.

THEOREM 1.3. *For infinitely many $k \in \mathbb{N}$ there exists a k-terminal acyclic directed grid G such that any reachability-preserving minor of G must use $\Omega(k^2)$ non-terminals.*

Cut, Flow and Distance Sparsifiers. We provide new constructions for quality-1 (exact) cut, flow and distance sparsifiers for k-terminal planar graphs, where all the terminals are assumed to lie on the same face. We call such k-terminal planar graphs *Okamura-Seymour* (OS) instances. They are of particular interest in the algorithm design and optimization community, due to the classical Okamura-Seymour theorem that characterizes the existence of feasible concurrent flows in such graphs (see e.g., [51, 15, 16, 46]).

We show that the size of quality-1 sparsifiers can be as small as $O(k^2)$ for OS instances. Prior to our work, the best-known cut and flow sparisifiers for such instances had size exponential in k [41, 6]. Formally, we have the following theorem.

THEOREM 1.4. *For any k-terminal planar graph G in which all terminals lie on the same face, there exist quality-1 cut, flow and distance sparsifers of size $O(k^2)$. Furthermore, the resulting sparsifiers are also planar graphs (with all terminals on the same face).*

We remark that all the above sparsifiers can be constructed in polynomial time (in n and k), but we will not optimize the running time here. As we mentioned above, previously the only known upper bound on the size of quality-1 cut and flow sparsifiers for OS instances was $O(k^2 2^{2k})$, given by [41, 6]. Our upper bound for cut sparsifier also matches the lower bound of $\Omega(k^2)$ for an OS instance given by [41]. More specifically, in [41], an OS instance (that is a grid in which all terminals lie on the boundary) is constructed, and used to show that any mimicking network for this instance needs $\Omega(k^2)$ edges, which is thus a lower bound for planar graphs (see the table below for an overview). Note that that even though our distance sparsifier is not necessarily a minor of the original graph G, it still shares the nice property of being planar as G. Furthermore, Krauthgamer and Zondiner [43] proved that there exists a k-terminal planar graph G (not necessarily an OS instance), such that any quality-1 distance sparsifier of G that is planar requires at least $\Omega(k^2)$ vertices.

We further provide a lower bound on the size of any *data structure* (not necessarily a graph) that approximately preserves pairwise terminal distances of *general*

[3]A planar digraph is a directed graph such that the underlying undirected graph (i.e., ignoring edge orientations) is planar.

Type of sparsifier	Graph family	Upper Bound	Lower Bound
Cut	Planar	$O(k2^{2k})$ [41]	$\|E(G')\| \geq \Omega(2^k)$ [36]
Cut	Planar OS	$O(k^2)$ [**new**]	$\|E(G')\| \geq \Omega(k^2)$ [41]
Flow	Planar OS	$O(k^2 2^{2k})$ [6]	follows from cut
Flow	Planar OS	$O(k^2)$ [**new**]	follows from cut
Distance (minor)	Planar OS	$O(k^4)$ [40]	$\Omega(k^2)$ [40]
Distance (planar)	Planar OS	$O(k^2)$ [**new**]	

Table 1: Overview on the current best trade-offs for quality-1 vertex sparsifiers.

k-terminal graphs, which gives a trade-off between the distance stretch and the space complexity.

THEOREM 1.5. *For any $\varepsilon > 0$ and integer $t \geq 2$, there exists a family of k-terminal n-vertex graph such that $k = o(n)$, and any data structure that approximates pairwise terminal distances within a multiplicative factor of $t - \varepsilon$ or an additive error $2t - 3$ must use $\Omega(k^{1+1/(t-1)})$ bits of space.*

Abboud and Bodwin [1] recently gave lower bounds for additive spanners, and their constructions imply that there exists an infinite family of k-terminal n-vertex graphs G such that $k = o(n^{2/3})$, and any data structure that approximates pairwise terminal distances within an additive error t needs $\Omega(k^{2-\varepsilon})$ bits, for any $\varepsilon > 0, t = O(n^\delta)$ and $\delta = \delta(\varepsilon)$. Note that their lower bounds are stronger than ours in the setting with additive error $2t - 1$ for $t \geq 3$, though our constructions are different from theirs and also give bounds in the multiplicative setting. See Section 6.3 for more discussions on this result.

Remark. Recently and independently of our work, Krauthgamer and Rika [42] constructed quality-1 cut sparsifiers of size $O(\gamma 2^{2\gamma} k^4)$ for planar graphs whose terminals are incident to at most $\gamma = \gamma(G)$ faces. In comparison with our upper-bound which only considers the case $\gamma = 1$, the size of our sparsifiers from Theorem 1.4 is better by a $\Omega(k^2)$ factor. Subsequent to our work, Karpov et al. [36] proved that there exists edge-weighted k-terminal planar graphs that require $\Omega(2^k)$ edges in any exact cut sparsifier, which implies that it is necessary to have some additional assumption (e.g., $\gamma = O(1)$) to obtain an exact cut sparsifier of $k^{O(1)}$ size.

1.2. Our Techniques. Our results for reachability-preserving minors (RPMs) are obtained by exploiting a technique of counting "branching" events between shortest paths in the directed setting. This technique was introduced by Coppersmith and Elkin [21], and has also been recently leveraged by Bodwin [9] and Abboud and Bodwin [2] in the context of distance/reachability preservers. Using this and a consistent tie-breaking scheme for shortest paths, we can efficiently construct an RPM for general digraphs of size $O(k^4)$ and by using a more refined analysis of branching events (see [2]), we can further reduce the size to be $O(k^3)$. We then combine our construction with a decomposition for planar digraphs (see [56]), to show that it suffices to maintain the reachability information among $O(k \log k)$ terminal pairs, instead of the naive $O(k^2)$ pairs, and then construct an RPM for planar digraphs with $O(k^2 \log k)$ vertices. The lower-bound follows by constructing a special class of k-terminal directed grids and showing that any RPM for such grids must use $\Omega(k^2)$

vertices. Similar ideas for proving the lower bound on the size of distance-preserving minors for undirected graphs have been previously used by Krauthgamer et al. [40].

We construct our quality-1 cut and distance sparsifiers by repeatedly performing *Wye-Delta transformations*, which are local operations that preserve cut values and distances and have proven very powerful in analyzing electrical networks and in the theory of circular planar graphs (see e.g., [38, 22, 26]). Khan and Raghavendra [39] used Wye-Delta transformations to construct quality-1 cut sparsifiers of size $O(k)$ for trees, which improves upon the previous bound in [13] by a constant factor, while our case (i.e., the planar OS instances) is more general and complicated and previously it was not clear at all how to apply such transformations to a broader class of graphs. Our approach is as follows. Given a k-terminal planar graph with terminals lying on the same face, we first embed it into some large grid with terminals lying on the boundary of the grid. Next, we show how to embed this grid into a "more suitable" graph, which we will refer to as "half-grid". Finally, using the Wye-Delta operations, we reduce the "half-grid" into another graph whose number of vertices can be bounded by $O(k^2)$. Since we argue that the above graph reductions preserve exactly all terminal minimum cuts, our result follows. Gitler [29] proposed a similar approach for studying the reducibility of multi-terminal graphs with the goal to classify all Wye-Delta reducible graphs, which is very different from our motivation of constructing small vertex sparsifiers with good quality.

The distance sparsifiers can be constructed similarly by slightly modifying the Wye-Delta operation. Our flow sparsifiers follow from the construction of cut sparsifiers and the flow/cut gaps for OS instances (which has been initially observed by Andoni et al. [6]). Our lower bound on the space complexity of any compression function approximately preserving terminal pairwise distance is derived by combining extremal combinatorics construction of Steiner Triple System that was used to prove lower bounds on the size of distance approximating minors (see [18]) and the incompressibility technique from [49].

1.3. Related Work. There has been a long line of work on investigating the tradeoff between the quality of the vertex sparsifier and its size (see e.g., [25, 41, 6] and Section 1.2). (Throughout, cut, flow and distance sparsifiers will refer to their vertex versions.) Quality-1 *cut sparsifiers* (or equivalently, mimicking networks) were first introduced by Hagerup et al. [33], who proved that for any graph G, there always exists a mimicking network of size $O(2^{2^k})$. Krauthgamer and Rika [41] showed how to build a mimicking network of size $O(k^2 2^{2k})$ for any planar graph G that is a minor of the input graph. They also proved a lower bound of $\Omega(k^2)$ on the number of edges of the mimicking network of planar graphs, and a lower bound of $2^{\Omega(k)}$ on the number of vertices of the mimicking network for general graphs.

Quality-1 vertex flow sparsifiers have been studied in [6, 31], albeit only for restricted families of graphs like quasi-bipartite, series-parallel, etc. It is not known if any general undirected graph G admits a constant quality flow sparsifier with size independent of $|V(G)|$ and the edge capacities. For the quality-1 distance sparsifiers, Krauthgamer et al. [40] introduced the notion of *distance-preserving minors*, and showed an upper-bound of size $O(k^4)$ for general undirected graphs. They also gave a lower bound of $\Omega(k^2)$ on the size of such a minor for planar graphs. Recently, building upon the work [4], Chang et al. [11] gave an algorithm for constructing a (quality-1) distance sparsifier of size $O(\min\{k^2, \sqrt{kn}\log^3 n\})$ for a k-terminal n-vertex undirected,unweighted planar graph.

Over the last two decades, there has been a considerable amount of work on understanding the tradeoff between the sparsifier's quality q and its size for $q > 1$, i.e., when the sparsifiers only *approximately* preserve the corresponding properties [19, 6, 50, 47, 12, 25, 48, 32, 14, 10, 25, 35, 18, 17, 27, 28, 23].

2. Preliminaries. Let $G = (V, E)$ be a directed graph with terminal set $K \subset V$, $|K| = k$, which we will refer to as a *k-terminal digraph*. We say G is a *k-terminal* DAG if G has no directed cycles. The *in-degree* of a vertex v, denoted by $\deg^-_G(v)$, is the number of edges directed towards v in G. A digraph $H = (V_H, E_H)$, $K \subset V_H$ is a (*vertex*) *reachability sparsifier* of G if for any $x, x' \in K$, there is a directed path from x to x' in H iff there is a directed path from x to x' in G. In this paper, a *minor operation* in a directed graph refers to deleting an edge or a vertex, or contracting an edge in the underlying undirected graph[4]. If H is obtained by performing minor operations in G, then we say that H is a *reachability-preserving minor* of G. We define the *size* of H to be the number of vertices in H.

Given a digraph G with a terminal set K of size k and a pair-set $P \subseteq K \times K$, we say that H is a reachability-preserving minor (RPM) with respect to P, if H is a minor of G that preserves the reachability information only among the pairs in P. Note that in the definition of vertex reachability sparsifiers, the *trivial* pair-set P contains $k(k-1)$ terminal-pairs, i.e., for any pair $x, x' \in K$, both (x, x') and (x', x) belong to P. Whenever we omit P, we mean to preserve the reachability information among all possible terminal pairs.

Let $G = (V, E, c)$ be an undirected graph with terminal set $K \subset V$ of cardinality k, where $c : E \to \mathbb{R}_{\geq 0}$ assigns a non-negative capacity to each edge. We will refer to such a graph as a *k-terminal graph*. Let $U \subset V$ and $S \subset K$. We say that a cut $(U, V \setminus U)$ is S-separating if it separates the terminal subset S from its complement $K \setminus S$, i.e., $U \cap K$ is either S or $K \setminus S$. We will refer to such cut as a *terminal cut*. The cutset $\delta(U)$ of a cut $(U, V \setminus U)$ represents the edges that have one endpoint in U and the other one in $V \setminus U$. The cost $\mathrm{cap}_G(\delta(U))$ of a cut $(U, V \setminus U)$ is the sum over all capacities of the edges belonging to the cutset. We let $\mathrm{mincut}_G(S, K \setminus S)$ denote the minimum cost of any S-separating cut of G. A graph $H = (V_H, E_H, c_H)$, $K \subset V_H$ is a *quality-q* (*vertex*) *cut sparsifier* of G with $q \geq 1$ if for any $S \subset K$, $\mathrm{mincut}_G(S, K \setminus S) \leq \mathrm{mincut}_H(S, K \setminus S) \leq q \cdot \mathrm{mincut}_G(S, K \setminus S)$.

3. Reachability-Preserving Minors for General Digraphs. In this section, we construct reachability-preserving minors (RPMs) for general digraphs and prove Theorem 1.1.

High-level idea of our constructions. We first observe that in order to construct an RPM for k-terminal digraphs, it suffices to have a subroutine for constructing an RPM for any k-terminal directed acyclic graph (DAG) G. To see this, consider the following reduction. Given a general digraph, we can first find a decomposition of the graph into strongly connected components[5] (SCCs) [55]. We then contract each SCC into a single vertex to obtain a DAG, from which we can construct an RPM H' by the subroutine for handling DAGs. By appropriately expanding back in H' the contracted SCCs that contain terminals, we obtain an RPM for the original digraph.

Now we describe our ideas for constructing an RPM for a k-terminal directed acyclic graph (DAG) G. We provide two such constructions. Let P denote the set of all vertex pairs in K. In the first construction (Section 3.1), we first apply a

[4]In general, an arbitrary edge contraction in a directed graph might cause new reachability. However, in our construction, we will carefully choose specific edges whose contraction preserves the pairwise terminal reachability.

[5]Recall that a digraph is *strongly connected* if there is a directed path between all pairs of vertices.

well-known tie-breaking scheme on G to guarantee that for any vertex pair s, t, there is a unique shortest path from s to t. Then we delete all vertices and edges that do not participate in any shortest path among terminal-pairs in P and finally we appropriately contract edges on the remaining paths. The resulting graph can be shown to be a minor of G of small size. In the second construction (Section 3.2), we simply start with a *minimal* reachability preserver H of G and then appropriately contract edges on H. By adapting an analysis from [2], we can show that the resulting graph is an RPM of G. Though the first construction has a worse size guarantee, the underlying idea seems more intuitive and the analysis is slightly easier in comparison to the second construction.

By using these two different subroutines, we can obtain RPMs for a general digraph G of size $O(k^4)$ and $O(k^3)$, respectively. Both minors can be constructed in polynomial time.

3.1. A Warm-up: An Upper Bound of $O(k^4)$.

Basic tools. Let $P \subseteq K \times K$ be a pair-set. We first review a useful scheme for breaking ties between shortest paths connecting some vertex pair from P. This tie-breaking is usually achieved by slightly perturbing the edge lengths of the original graph such that no two paths have the same length (note that in our case, edge lengths are initially one). The perturbation gives a *consistent* scheme in the sense that whenever π is chosen as a shortest path, every sub-path of π is also chosen as a shortest path. Below we formalize these ideas using two definitions and a lemma from [9].

Definition 3.1 (Tie-breaking Scheme). *Given a k-terminal digraph G, a* shortest path tie breaking scheme *is a function π that maps every pair of vertices (s, t) to some shortest path between s and t in G. For any pair-set P, we let $\pi(P)$ denote the union over all shortest paths between pairs in P with respect to the scheme π.*

Definition 3.2 (Consistency). *A tie-breaking scheme is consistent if, for all vertices $y, x, x', y' \in V$, if $x, x' \in \pi(y, y')$ with $d(y, x) < d(y, x')$, then $\pi(x, x')$ is a sub-path of $\pi(y, y')$.*

Lemma 3.3 ([9]). *For any k-terminal digraph G, there is a consistent tie-breaking scheme in G.*

We remark that for any k-terminal digraph with n vertices, the consistent tie-breaking scheme can be constructed in polynomial (in n) time [21].

Constructing RPMs for DAGs. Let G be a k-terminal DAG. Given a tie-breaking scheme π, the first step to construct an RPM is to start with an empty graph H' and then for every pair $p \in P$, repeatedly add the shortest-path $\pi(p)$ to H'. We can alternatively think of this as deleting vertices and edges that do not participate in any shortest path among terminal-pairs in P with respect to the scheme π. Clearly, the DAG $H' = (V_{H'}, E_{H'})$, $E_{H'} := \pi(P)$, is a minor of G and preserves all reachability information among pairs in P. We next review the notion of a branching event, which will be useful to bound the size of H'.

Definition 3.4 (Branching Event). *A* branching event *is a set of two distinct directed edges $\{e_1 = (u_1, v), e_2 = (u_2, v)\}$ that enter the same node v.*

Lemma 3.5. *The* DAG *H' has at most $|P|(|P| - 1)/2$ branching events.*

Proof. First, note that by construction of H', we can associate each edge $e \in E_{H'}$ with some pair $p \in P$ such that $e \in \pi(p)$. To prove the lemma, it suffices to show that for any two terminal-pairs $p_1, p_2 \in P$, there is at most one branching event in the graph induced by $\pi(p_1) \cup \pi(p_2)$. Suppose towards contradiction that there exist two terminal pairs p_1, p_2 that have two branching events in $\pi(p_1) \cup \pi(p_2)$. More specifically, we assume there exist two branching events

$$b := \{e_1 = (u_1, v), e_2 = (u_2, v)\} \text{ and } b' := \{e_1 = (u'_1, v'), e_2 = (u'_2, v')\},$$

where e_i and e'_i lie on the dipath $\pi(p_i)$, for $i = 1, 2$.

Assume without loss of generality that the vertex v appears before v' in the dipath $\pi(p_1)$. We then claim that v must also appear before v' in the dipath $\pi(p_2)$, since otherwise we would have a directed cycle between v and v', thus contradicting the fact that H' is acyclic. Since the tie-breaking scheme π is consistent (Lemma 3.3), it follows that the dipaths $\pi(p_1)$ and $\pi(p_2)$ must share the subpath $\pi(v, v')$. Thus, $\pi(p_1)$ and $\pi(p_2)$ use the same edge that enters the node v', i.e., $e'_1 = e'_2$. However, by definition of a branching event, the edges that enter a node must be distinct, contradicting the fact that b' is a branching event. This implies that there cannot be two branching events for the terminal pairs p_1 and p_2, thus proving the lemma. ∎

We now present our algorithm for constructing an RPM for a DAG.

Algorithm 3.1 MinorSparsifyDag (k-terminal DAG G, pair-set P)

1: Set $H = \emptyset$.
2: Compute a consistent tie-breaking scheme π for shortest paths in G.
3: For each $p \in P$, add the shortest path $\pi(p)$ to H.
4: **while** there is an edge (u, v) such that v is non-terminal and $\deg^-_H(v) = 1$ **do**
5: Contract the edge (u, v).
6: **end while**
7: **return** H

Lemma 3.6. *Given a k-terminal* DAG *G with a pair-set P,* Algorithm 3.1 *outputs an RPM H for G with respect to P with $O(|P|^2)$ non-terminals.*

Proof. We first argue that H is an RPM with respect to the terminals. Indeed, after Line 2 of the algorithm, graph H can viewed as deleting vertices and edges from G that do not lie on any of the shortest path among terminal pairs in P, chosen according to the scheme π. Thus, at this point H is clearly a minor of G that preserves the reachability information among the pairs in P. The edge contractions we perform in the remaining part of the algorithm guarantee that the resulting H remains an RPM of G with respect to P.

To bound the number of non-terminals in H, note that every non-terminal $v \in V_H \setminus K$ has in-degree at least 2, and thus it corresponds to at least one branching event. Lemma 3.5 shows that the number of branching events is at most $O(|P|^2)$. Observing that edge contractions in Line 5 do not affect this number, we get that the number of non-terminals in H is $O(|P|^2)$. ∎

From DAG to general digraphs. We next show how the construction of RPMs can be reduced from general digraphs to DAGs, and prove the following theorem.

Theorem 3.7. *Given a k-terminal digraph G with a pair-set P, there exists a polynomial-time algorithm that outputs an RPM H for G with respect to P with $O(|P|^2)$ non-terminals.*

Taking P to be the trivial pair-set, i.e., P being the set of all possible $k(k-1)$ terminal pairs, we get an RPM of size $O(k^4)$.

Algorithm 3.2 MinorSparsify (k-terminal digraph G, pair-set P)

`// Preprocessing Step`
Compute a strongly connected component (SCC) decomposition of G. Let $\mathcal{D}$ and $\mathcal{D}_K$ denote the set of all SCCs, and the set of SCCs containing terminals in G, respectively.
Let f be some initially empty labelling that records the SCC of every vertex.
for all SCC $C \in \mathcal{D}$ **do**
    if C contains some terminal $x \in K$ **then**
        For all $v \in C$, set $f(v) = x$.
    else
        Choose some arbitrary $u \in C$, and set $f(v) = u$, for all $v \in C$.
    end if
end for
for all SCC $C \in \mathcal{D}_K$ **do**
    while C contains some non-terminal v **do**
        Choose some directed edge (v, u) inside C, and contract v into u.
    end while
end for
Let $\hat{G}$ denote the resulting graph. Let $\hat{\mathcal{D}}$ and $\hat{\mathcal{D}}_K$ denote the set of all SCCs, and the set of SCCs containing terminals in $\hat{G}$, respectively.

`// Main Procedure`
Contract each SCC in $\hat{\mathcal{D}}$ into a single vertex, producing the DAG $G' = (V', E')$.
Let $K' = \emptyset$ and $P' = \emptyset$ be the terminal set and pair-set of G', respectively.
For all $k \in K$, add $f(k)$ to K' and remove duplicates, if any.
For all $(s, t) \in P$, add $(f(s), f(t))$ to P' if $f(s) \neq f(t)$.
Set $H' =$MinorSparsifyDag(G', P').
Let H be the graph obtained by expanding back all contracted SCCs in $\hat{\mathcal{D}}_K$ in H'.
return H

Proof of Theorem 3.7. In order to construct an RPM for G, we first reduce G to be a DAG by contracting all the strongly connected components (SCCs) into a single vertex in G. However, since a SCC might contain more than one terminal, we will contract such SCCs to be cliques on the corresponding terminals. Then we apply Algorithm 3.1 on the resulting graph by viewing these terminal-cliques as a "super" vertex which we can expand back to restore all its terminals. We refer the reader to the overview at the beginning of Section 3 for more intuition. Our algorithm for constructing RPMs for general digraphs is formally described in Algorithm 3.2.

By construction, the algorithm runs in polynomial time. The main intuition behind the correctness of the algorithm lies on two important observations. First, vertices belonging to the same SCCs can always reach each other. Second, vertices belonging to different SCCs can reach each other if the corresponding vertices in the contracted graph can do so. We have the following useful observation.

Fact 3.8. *For any strongly connected digraph $G = (V, E)$, contracting any edge $e \in E$ results in another strongly connected digraph $G' = (V', E')$.*

Now we show that the graph H output by MinorSparsify is an RPM of G. It is easy to verify that the produced graph H is indeed a minor of G. To show the correctness, we will prove that H preserves the reachability information among all pairs from P in G. Before doing that, observe that the graph $\hat{G}$ obtained after the preprocessing step is a reachability preserving minor of G with respect to P. Indeed, this can be inferred by a repeated application of Fact 3.8 to each SCC containing terminal vertices.

Now, let $(s,t) \in P$ be any terminal-pair in G. Assume that t is reachable from s in G. We distinguish two cases:

1. If s and t belong to the same SCC in $\mathcal{D}$, they do also belong to the corresponding SCC in $\hat{\mathcal{D}}$. In Line 13, s and t are contracted into a single terminal. However, since the contracted SCC contains terminals, it is expanded back to its original form in $\hat{\mathcal{D}}$ in Line 24. Thus, it follows that t is reachable from s in the output graph H.
2. If s and t do not belong to the same SCC in $\mathcal{D}$, they must also not belong to the same SCC in $\hat{\mathcal{D}}$. Let $f(s)$ and $f(t)$ denote the terminals in the DAG G' obtained by contracting their corresponding components in $\hat{\mathcal{D}}$ (Line 13). Since t is reachable from s in $\hat{G}$, note that $f(t)$ must also be reachable from $f(s)$ in G'. By Lemma 3.6, it follows that $f(t)$ is reachable from $f(s)$ in the RPM H' of G'. Expanding back the SCCs that contain terminals in H' (Line 24), we can construct the directed path $s \rightsquigarrow f(s) \rightsquigarrow f(t) \rightsquigarrow t$ in H, which shows that t is also reachable from s in the output graph H.

When t is not reachable from s in G, we can similarly show that t is also not reachable from s in H, thus concluding the correctness proof.

We now bound the number of non-terminals in H. Since the DAG G' has $|P'| \leq |P|$ pairs, it follows by Lemma 3.6 that H' has $O(|P|^2)$ non-terminals. Further note that the algorithm in Line 24 only expands back terminals and does not increase the number of non-terminals. Therefore, the number of non-terminals in H is $O(|P|^2)$. ∎

3.2. An Improved Bound of $O(k^3)$.

Now we describe our improved construction. As mentioned earlier, the main idea of this improvement is to use a better construction of RPMs for DAGs.

A better construction of RPMs for DAGs. Given a k-terminal DAG $G = (V, E)$ with a pair-set P, a digraph $H = (V, E_H)$ with $E_H \subseteq E$ is a *reachability preserver (RP)* of G if for any $(s,t) \in P$, there is a directed path from s to t in H iff there is a directed path from s to t in G. We say that H is a *minimal reachability preserver* of G if (i) H is an RP of G, and (ii) no edge can be deleted from H such that the resulting digraph satisfies (i). The following lemma is implicit in [2], and we include it here for the sake of completeness.

LEMMA 3.9. *The* DAG $H = (V, E_H)$ *has at most* $k \cdot |P|$ *branching events.*

Proof. For each pair $(s,t) \in P$ such that t is reachable from s, we associate an arbitrary directed path $\tilde{\pi}(s,t)$ from s to t in H. Since H is a minimal reachability preserver, it holds that for every edge $e \in E_H$, there must be some pair $(s,t) \in P$ such that deleting e from H implies that s cannot reach t, i.e., $s \not\rightsquigarrow t$ in $H \setminus \{e\}$. This naturally leads to a relationship between edges in H and pairs in P. Specifically, we say that every edge $e \in E_H$ is *owned* by one such pair $(s,t) \in P$.

Next, for each $(s,t) \in P$ such that t is reachable from s, we let $B^H_{(s,t)}$ denote the set of all branching events $\{e_1, e_2\}$ in H such that either e_1 or e_2 is owned by (s,t). Note that for any branching event $\{e_1, e_2\}$ such that e_1 is owned by the pair $(s,t) \in P$, e_2 cannot be owned by (s,t). This is true as otherwise there would be two directed paths from s to t, where one path uses e_1 and the other uses e_2; then after deleting edge e_1, there is still another path from s to t, which contradicts the assumption that e_1 is owned by (s,t). This implies that for any event $\{e_1, e_2\} \in B^H_{(s,t)}$, *exactly one* of e_1 or e_2 is owned by (s,t).

Consider the set $\bigcup\{B^H_{(s,t)} \mid (s,t) \in P\}$ and note that it contains all the branching events. In order to prove the lemma, it suffices to show that $|B^H_{(s,t)}| \leq k$, for every $(s,t) \in P$. To this end, suppose towards contradiction that there exists a pair $(s,t) \in P$ such that $|B^H_{(s,t)}| \geq k+1$. Then by the pigeon-hole principle, there exist two branching events

$$\{(x_1,b_1),(x_2,b_1)\},\ \{(y_1,b_2),(y_2,b_2)\} \in B^H_{(s,t)}$$

entering the nodes b_1 and b_2, such that (s,t) owns (x_1,b_1) and (y_1,b_2), and the other edges are owned by pairs that share a common left terminal (as there are at most k distinct terminals), i.e.,

$$(x_2,b_1) \text{ is owned by } (u,v_1) \text{ and } (y_2,b_2) \text{ is owned by } (u,v_2),$$

for some $u \in K$ and $(u,v_1),(u,v_2) \in P$. Recall that by the definition of $B^H_{(s,t)}$, y_1 and y_2 are distinct vertices. We claim that $b_1 \neq b_2$. Suppose towards contraction that $b_1 = b_2$. Then it must be that either (i) $y_2 \neq x_2$ or (ii) $y_2 = x_2$ and $x_1 \neq y_1$. In case (i), there are two paths from u to v_1, one using the edge (x_2,b_1) and the other using (y_2,b_1), which contradicts the fact that (x_2,b_1) is owned by (u,v_1). In case (ii), there are two paths from s to t, one using the edge (x_1,b_1) and the other using (y_1,b_1), which contradicts the fact that (x_1,b_1) is owned by (s,t), and shows that our claim holds.

Next, assume without loss of generality that the node b_1 appears before b_2 in $\tilde{\pi}(s,t)$. Now, since the pair (u,v_2) owns the edge (y_2,b_2), every path $u \rightsquigarrow v_2$ must use the edge (y_2,b_2), which in turn implies that every path $u \rightsquigarrow b_2$ must use the edge (y_2,b_2). Furthermore, since H is a DAG, the edge (y_2,b_2) must be the last edge on every path from u to b_2.

Finally, we can form a path $u \rightsquigarrow b_2$ by first taking the path[6] $\tilde{\pi}(u,v_1)[u \rightsquigarrow b_1]$ and then extend it by concatenating it with the path $\tilde{\pi}(s,t)[b_1 \rightsquigarrow b_2]$. Note that since (y_2,b_2) is the last edge on this path and b_1 appeared before b_2, it must be the case that $(y_2,b_2) \in \tilde{\pi}(s,t)[b_1 \rightsquigarrow b_2]$. This further implies that $(y_2,b_2) \in \tilde{\pi}(s,t)$. Therefore, the path $\tilde{\pi}(s,t)$ contains both (y_1,b_2) and (y_2,b_2), which contradicts the fact that $\tilde{\pi}(s,t)$ is a simple path from s to t and completes the proof of the lemma. ∎

The above lemma leads to the following algorithm for constructing an RPM for a DAG.

Algorithm 3.3 MinorSparsifyDag2 (k-terminal DAG G, pair-set P)

1: Set $H = (V, E_H)$ to be the minimal reachability preserver with respect to P.
2: Remove isolated non-terminal vertices from H, if any.
3: **while** there is an edge (u,v) such that v is non-terminal and $\deg^-_H(v) = 1$ **do**
4: Contract the edge (u,v).
5: **end while**
6: **return** H

---

[6]Let $x, y, x', y' \in V$, $\tilde{\pi}(x,y)$ be a directed path from x to y, and suppose $x', y' \in \tilde{\pi}(x,y)$ with x' appearing before y'. Then $\tilde{\pi}(x,y)[x' \rightsquigarrow y']$ denotes the directed subpath from x' to y' in $\tilde{\pi}(x,y)$.

By using similar arguments as in the proof of Lemma 3.6, we have the following lemma.

LEMMA 3.10. *Given a k-terminal* DAG *G with a pair-set P,* Algorithm 3.3 *outputs an RPM H for G with respect to P with $O(k \cdot |P|)$ non-terminals.*

We remark that the above construction builds upon the minimal reachability preserver H (Line 1 in Algorithm 3.3), which can be constructed in polynomial time. This can be achieved by a simple greedy algorithm: if there exists an edge e in G whose removal does not change the reachability information among pairs in P, delete e from G; repeat until no such edge exists. Moreover, note that the non-terminal removals and the edge contractions in Lines 2-4 of Algorithm 3.3 can easily be implemented in polynomial time. Therefore, we get that for any DAG G, the RPM H of G from Lemma 3.10 can be constructed in polynomial time.

From DAGs to general digraphs. By using similar arguments as in the proof of Theorem 3.7, we have the following guarantee.

THEOREM 3.11. *Given a k-terminal digraph G with a pair-set P, there exists a polynomial-time algorithm that outputs an RPM H for G with respect to P with $O(k \cdot |P|)$ non-terminals.*

Taking P to be the trivial pair-set we get an RPM of size $O(k^3)$, which proves Theorem 1.1.

4. Reachability-Preserving Minors for Planar Digraphs. In this section we show that any k-terminal planar digraph G admits a reachability-preserving minor of size $O(k^2 \log k)$ and thus prove Theorem 1.2. This matches the lower-bound of Theorem 1.3 up to an $O(\log k)$ factor. The main idea is as follows. Given a k-terminal planar digraph G with the trivial pair-set P, $|P| = k(k-1)$, our approach is to slightly increase the number of terminals while considerably reducing the size of the pair-set P, under the condition that no reachability information is lost among the terminal-pairs in P.

Preprocessing Step. For any k-terminal n-vertex planar digraph G with terminal set K, we can first apply Theorem 1.1 to get a reachability-preserving minor G' with $O(k^3)$ vertices and then restrict our attention to finding an RPM for G'. To simplify the notation, throughout this section, we will use G instead of G', i.e., we assume that our terminal graph G has at most $n' := O(k^3)$ vertices. Furthermore, without loss of generality, we can assume that there is no isolated vertex in K. Otherwise, we can simply find an RPM with respect to the set of non-isolated terminal vertices, and then add all the isolated terminals back.

Decomposition into Path-Separable Digraphs and the Algorithm. Given a digraph $G = (V, E)$, a set $S \subset V$ is called an *α-separator* of G if the removal of S partitions G into connected components (when forgetting the orientation of edges), each of size at most $\alpha \cdot |V|$, where $1/2 \leq \alpha < 1$. If the vertices of S consist of the union over r directed paths of G, for some $r \geq 1$, we say that G is *(α, r)-path separable.* We now review the following reduction due to Thorup [56] and include its proof in Appendix A for the sake of completeness.

THEOREM 4.1 ([56]). *Given a planar digraph $G = (V, E)$ with $n' = O(k^3)$ vertices, we can construct a series of digraphs $G_0, \ldots, G_b$ for some $b = O(k^3)$ such that the total number of vertices and edges over all G_i's is linear in the number of vertices and edges in G, and*

1. *Each vertex and edge of G appears in at most two G_i's.*

2. *For all $u, v \in V$, if there is a directed path R from u to v in G, there is a G_i that contains R.*
3. *Each $G_i = (V_i, E_i)$ is $(1/2, 6)$-path separable. If we let S_i denote the set of 6 directed paths corresponding to the $1/2$-separator, then S_i induces a connected subgraph of the underlying undirected graph G_i.*
4. *For each $i \geq 0$, there exists a special vertex r_i in G_i such that all vertices in V_0 and $V_i \setminus \{r_i\}, i \geq 1$ belong to V. Furthermore, r_i can only be the endpoint of any path Q in S_i and the path $Q - \{r_i\}$ is also contained in G.*
5. *Each G_i is a minor of G.*

We now review how directed reachability can be represented by a separator that consists of directed paths. Let G be a k-terminal directed graph that contains some directed path Q. Assume that the vertices of Q are ordered in increasing order in the direction of the path. For each terminal $x \in K$, let $\text{to}_x[Q]$ be the first vertex in Q that can be reached by x, and let $\text{from}_x[Q]$ be the last vertex in Q that reaches x. If x does not reach Q, then $\text{to}_x[Q] = \emptyset$, and if Q does not reach x, then $\text{from}_x[Q] = \emptyset$. We say that x *connects to* Q via $\text{to}_x[Q]$ if $\text{to}_x[Q] \neq \emptyset$, and x *connects from* Q via $\text{from}_x[Q]$ if $\text{from}_x[Q] \neq \emptyset$.

The following fact immediately follows.

FACT 4.2. *For any terminal pair (s, t), there is a directed path from s to t intersecting Q if and only if s connects to Q via $\text{to}_s[Q]$ and t connects from Q via $\text{from}_t[Q]$, and $\text{to}_s[Q]$ equals or precedes $\text{from}_t[Q]$ in Q.*

We now combine the above tools to give our labelling algorithm Algorithm 4.1 aimed at reducing the size of the trivial pair-set $P = K \times K$. That is, we will mark some non-terminals in G as new terminals and find a terminal pair-set P' of smaller size that preserves reachability of pairs in $K \times K$. By Theorem 4.1, we restrict our attention only to the digraphs G_i. Let $K_i := V(G_i) \cap K$ be the set of terminals restricted to the graph G_i.

LEMMA 4.3. *Let $G = (V, E)$ be a k-terminal planar digraph with $n' = O(k^3)$ vertices such that there is no isolated vertex in the terminal set K. Let $P' := \bigcup_{i=0}^{b} P'_i$, where P'_i is the pair-set output by running* Algorithm 4.1 *on the digraph G_i. Then all the vertices involved in P' belong to V and the size of $|P'|$ is at most $O(k \log k)$. Moreover, if a digraph H is a reachability-preserving minor of G with respect to P', then H is a reachability-preserving minor of G with respect to all terminal pairs.*

Proof. Let $G_0, \ldots, G_b$ be the graphs obtained by the reduction in Theorem 4.1 and consider applying Algorithm 4.1 to each of them. By Item 2 of Theorem 4.1, each terminal appears in at most two G_i's. Thus at each level of the recursion (studied over all G_i's), there will be at most $O(k)$ active G_i's. Note that by construction, all the vertices involved in the pair-set P' belong to V, i.e., no special vertex r_i $(i \geq 1)$ will be marked as a new terminal. Also, note that the separator properties of planar graphs imply that the subgraph at each recursive level is $(1/2, 6)$-separable and there are $O(\log n') = O(\log k)$ recursive calls overall.

We next bound the size of the pair-set P'. Let q denote the total number of newly added terminals in Lines 9 and 10 per level of recursion. Since there are $O(k)$ terminals, each adding at most $O(1)$ new terminals, it follows that $q = O(k)$. First, we argue about the number of pairs added in Lines 9 and 10. Since this is bounded by q, it follows that there are $O(k \log k)$ pairs added in Lines 9 and 10 over all calls of REDUCEPAIRSET. Second, we bound the number of pairs added when sparsifying the separator paths, i.e., pair additions in Line 13. For all the separators in the same

Algorithm 4.1 ReducePairSet (planar digraph G_i, vertex $r_i \in V_i$, terminals K_i)

1: **if** $|V(G_i)| \leq 1$ or $K_i = \emptyset$ **then return** $\emptyset$.
2: Let $P'_i = \emptyset$ be the new pair-set.
3: Compute a 1/2-separator S_i of G_i consisting of 6 directed paths by Item 3 of Theorem 4.1.
4: **for** each directed path $Q \in S_i$ **do**
5:     `// Addition of terminal connections with` Q
6:     Let $Q' = Q \cap K_i$.
7:     **if** $r_i = r_0$, **then** let $z = \emptyset$; **otherwise** let $z = r_i$.
8:     **for** each terminal $x \in K_i$ **do**
9:         If x connects to $Q - \{z\}$ via $\text{to}_x[Q]$, then mark $\text{to}_x[Q]$ a terminal, add it to Q', and add $(x, \text{to}_x[Q])$ to P'_i.
10:         If x connects from $Q - \{z\}$ via $\text{from}_x[Q]$, then mark $\text{from}_x[Q]$ a terminal, add it to Q', and add $(\text{from}_x[Q], x)$ to P'_i.
11:     **end for**
12:     `// Sparsification of` Q `using` Q'
13:     Define directed pairs (s, t), where s and t are consecutive terminals of Q', according to the ordering of Q and add all these pairs to P'_i.
14: **end for**
15: Let $\{C_i^{(j)}\}_{j=1}^{\ell}$ be the resulting connected components of $G_i \setminus S_i$.
16: **for** $j = 1, \ldots, \ell$ **do**
17:     Let $K_i^{(j)} = C_i^{(j)} \cap K_i$.
18:     Let $G_i^{(j)}$ be the graph obtained by first taking the subgraph of G_i induced by $C_i^{(j)} \cup S_i$ and then contracting all vertices in S_i to the root r_{S_i}.
19: **end for**
20: `// Note that reachability information about terminals in` S_i `are taken care of.`
21: **return** $P'_i \cup \bigcup_{j=1}^{\ell} \text{ReducePairSet}(G_i^{(j)}, r_{S_i}, K_i^{(j)})$.

level of recursion, note that q equals $\sum_j |Q'_j|$, where Q'_j denotes the set of newly added terminals for a single separator path, and the sum is over all separators at the same recursive level. By Line 13, it follows that we need only $|Q'_j| - 1$ pairs to represent each such directed path. Thus, per recursive call, the total number of newly added pairs is at most $\sum_j (|Q'_j| - 1) = O(q) = O(k)$. Summing these over all $O(\log k)$ levels of recursion gives that $|P'| = O(k \log k)$.

Finally, we argue that P' is a pair-set that can recover reachability information among terminals. First, note that for any terminal $v \in K$, there exists at least one pair in P' that contains v. This is true as v is not isolated, and thus at least one pair (v, t) or (s, v) will be added in Lines 9 and 10.

Fix any terminal pair $(s, t) \in K \times K$. If t is not reachable from s, then in any RPM H of G with respect to P', there is also no path from s to t in H. Otherwise, assume that t is reachable from s in G. Let R be a directed path from s to t in G. By Item 2 of Theorem 4.1, there is some digraph G_i that contains R. Then, R must intersect with some separator path Q, at some level of the recursion of the above algorithm on G_i. Furthermore, this path entirely belongs to G and thus does not uses any special vertex r_i (for $i \geq 1$). The above argument gives that P' contains all the necessary information to give a (possibly) another directed path from s to t in G. ∎

Applying Theorem 3.11 on the digraph G with the pair-set P', as defined by the above lemma, we get Theorem 1.2.

4.1. Reachability-Preserving Minors: Lower-bound for Planar DAGs. In this section we prove that there exists an infinite family of k-terminal acyclic directed grids such that any RPM for such graphs needs $\Omega(k^2)$ non-terminals (i.e., prove Theorem 1.3). We achieve this by adapting the ideas of Krauthgamer et al. [40], from their lower-bound proof on distance-preserving minors for undirected graphs.

We start by defining of our lower-bound instance. Fix k such that $r = k/4$ is an integer. Initially, construct an undirected $(r+1)\times(r+1)$ grid, where all the k terminals lie on the boundary, except at the corners, and declare all non-boundary vertices non-terminals. Remove the four corner vertices, and then all boundary edges connecting the terminals. Now, make the graph directed by first directing each horizontal edge from left to right, and then directing each vertical edge from top to bottom. Let G denote the resulting k-terminal directed grid. It is easy to verify that G is acyclic.

THEOREM 4.4. *For infinitely many $k \in \mathbb{N}$ there exists a k-terminal acyclic directed grid G such that any RPM of G must use $\Omega(k^2)$ non-terminals.*

Proof. Let G be the k-terminal grid defined as above. Note that there are r terminals on each side of the grid. Let H be any RPM of G. Recall that H contains all terminal vertices from G. Furthermore, let $x_1, x_2, \ldots, x_r$ be the terminals on the left-hand side of the grid, ordered from top to bottom. Similarly, let $y_1, y_2, \ldots, y_r$ be the terminals on the right-hand side. Let $u_1, u_2, \ldots, u_r$ be the terminals on the top-side of the grid, ordered from left to right. Similarly, let $v_1, v_2, \ldots, u_r$ be the terminals on the bottom-side. By construction of G, for an index pair (i, j) with $i < j$, there is no directed path from x_j to y_i or u_j to v_i.

We first note that there is a *unique* directed path from x_i to y_i, and a unique path from u_i to v_i in G for any $1 \leq i \leq r$. We then note that we cannot perform any edge or vertex deletion in the process of constructing H. This is true as any edge deletion will irreversibly destroy the reachability of some terminal pair. We now show the following lemma.

LEMMA 4.5. *For any $i = 1, \ldots, r$, there is a* unique *directed path from x_i to y_i in H.*

Proof. Assume to the contrary that there are at least two directed paths from x_i to y_i in H. Since H is an RPM of G and there is a unique path from x_i to y_i in G, then an edge contraction must have been performed to get H from G. Suppose without loss of generality that a vertical edge from row j to row $j + 1$ has been contracted. Then after such a contraction, the vertex y_j will be reachable from x_{j+1} in H, which will contradict the fact that y_j is not reachable from x_{j+1} in G and that H is an RPM of G. Thus, there is unique path from x_i to y_i in H. ◻

We will let P_H^i be the unique directed path from x_i to y_i in H, for $i = 1, \ldots, r$. Throughout we will refer to such paths as *horizontal*.

CLAIM 4.6. *The horizontal directed paths $P_H^1, P_H^2, \ldots, P_H^r$ are vertex disjoint in H.*

Proof. Suppose towards contradiction that there exist some i and j with $i < j$ such that P_H^i and P_H^j intersect at some vertex z in H. This implies that there are directed paths from x_i and x_j to z, and from z to y_i and y_j. The latter implies that there is a directed path from x_j to y_i in H. However, by construction of G, we know that x_j cannot reach y_i for $i < j$, contradicting the fact that H is an RPM of G. ◻

We can apply a symmetric argument to the *vertical* paths in H. More specifically, define Q^i_H to be the *unique* directed path from u_i to v_i in H, for $i = 1, \ldots, r$. (The uniqueness of such paths can be shown similarly to the proof of Lemma 4.5.) Then we get the following symmetric claim.

CLAIM 4.7. *The vertical directed paths $Q^1_H, Q^2_H, \ldots, Q^r_H$ are vertex disjoint in H.*

We next argue that all the horizontal and the vertical paths must intersect with each other.

CLAIM 4.8. *Any pair of horizontal and vertical paths P^i_H and Q^j_H intersect in H.*

Proof. Since H is a minor of G, any directed path that connects two terminals in H can be mapped back to a directed path connecting two terminals in G. Let P_i and Q_j be the corresponding directed paths in G that are obtained by expanding back the directed paths P^i_H and Q^j_H in H. By construction of G, the horizontal and the vertical directed paths between terminals are unique, implying that P_i and Q_j must intersect at some vertex of G. By performing the backtracked minor-operations on this vertex yields an intersection vertex between P^i_H and Q^j_H in H. ∎

The last claim we need shows that no pair of horizontal and the vertical paths intersects intersect at a terminal vertex.

CLAIM 4.9. *No pair of horizontal and vertical paths P^i_H and Q^j_H intersects at a terminal vertex in G.*

Proof. Consider the terminal pairs (x_i, y_i) and (u_j, v_j) corresponding to the paths P^i_H and Q^j_H. Note that by construction of G, the set of terminals reachable from both x_i and u_j in G is $\{y_i, y_{i+1}, \ldots, y_r\} \cup \{v_j, v_{j+1}, \ldots, v_r\}$. Since H is an RPM of G, x_i and u_j must also be able to reach this terminal-set in H and also P^i_H and Q^j_H cannot intersect at any terminal in $\{y_1, \ldots, y_{i-1}\} \cup \{v_1, \ldots, v_{j-1}\}$. Now, suppose towards contradiction that P^i_H and Q^j_H intersect at some terminal y_k, for $k \in \{i+1, \ldots, r\}$. This implies that in the path P^i_H, there is a directed path from y_k to y_i, for $k > i$, giving a contradiction by construction of G. Furthermore, observe that P^i_H and Q^j_H cannot intersect at y_i, as otherwise we would have a directed path from y_i to v_j, which is a contradiction by construction of G. Applying a similar argument to the case when paths intersect at some terminal v_ℓ, for $k \in \{j+1, \ldots, r\}$, gives the claim. ∎

We know have all the necessary tools to prove the theorem. Claim 4.8 shows that the paths P^i_H and Q^j_H intersect in H and let $z^{i,j}_H$ denote one of the intersection vertices. Now, we must show that all these vertices are distinct. To this end, assume that $z^{i_1,j_1}_H = z^{i_2,j_2}_H$. Since these vertices belong to both $P^{i_1}_H$ and $P^{i_2}_H$, by Claim 4.6 we get that $i_1 = i_2$. Similarly, by Claim 4.7 we get that $j_1 = j_2$. Thus, we have that all vertices $z^{i,j}_H$, for $i, j = 1, 2, \ldots, r$ are distinct. Since Claim 4.9 implies that none of this intersection vertices is a terminal, we conclude that H must contain at least $r^2 = (k/4)^2$ non-terminals. ∎

5. An Exact Cut Sparsifier of Size $O(k^2)$.

In this section we show that given a k-terminal planar graph, where all terminals lie on the same face, one can construct a quality-1 cut sparsifier of size $O(k^2)$. Note that it suffices to consider the case when all terminals lie on the *outer* face. We first present some basic tools.

5.1. Basic Tools.

Wye-Delta Transformations. In this section we investigate the applicability of some graph reduction techniques that aim at reducing the number of non-terminals

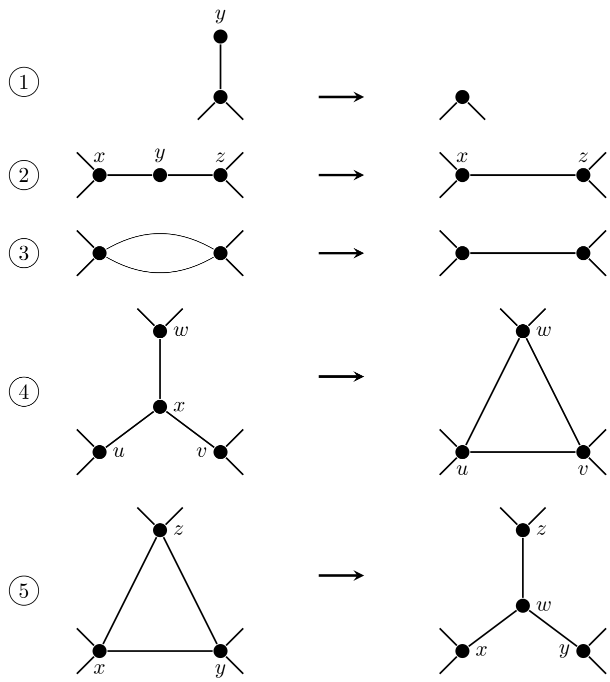

Fig. 1: Wye-Delta operations: 1. Degree-one reduction; 2. Series reduction; 3. Parallel reduction; 4. Wye-Delta transformation; 5. Delta-Wye transformation.

in a k-terminal graph. We start by reviewing the so-called *Wye-Delta* operations in graph reductions. These operations consist of five basic rules, which we describe below. (See Fig. 1 for illustrations.)

1. *Degree-one reduction:* Delete a degree-one non-terminal and its incident edge.
2. *Series reduction:* Delete a degree-two non-terminal y and its incident edges (x, y) and (y, z), and add a new edge (x, z) of capacity $\min\{c(x, y), c(y, z)\}$.
3. *Parallel reduction:* Replace all parallel edges by a single edge whose capacity is the sum of the capacities over all parallel edges.
4. *Wye-Delta transformation:* Let x be a degree-three non-terminal with neighbour set $\Gamma(x) = \{u, v, w\}$. Assume without loss of generality[7] that for any pair $u, v \in \Gamma(x)$, $c(u, x) + c(v, x) \geq c(w, x)$, where $w \in \Gamma(x) \setminus \{u, v\}$. Then we can delete x (along with all its incident edges) and add edges $(u, v), (v, w)$ and (w, u) with capacities $(c(u, x) + c(v, x) - c(w, x))/2$, $(c(v, x) + c(w, x) - c(u, x))/2$ and $(c(u, x) + c(w, x) - c(v, x))/2$, respectively.
5. *Delta-Wye transformation:* Delete the edges of a triangle connecting x, y and z, introduce a new non-terminal vertex w and add new edges (w, x), (w, y) and (w, z) with edge capacities $c(x, y) + c(x, z)$, $c(x, y) + c(y, z)$ and $c(x, z) + c(y, z)$ respectively.

By definition, it holds that performing the above rules on a terminal graph pre-

[7] Suppose there exist a pair $u, v \in \Gamma(x)$ with $c(u, x) + c(v, x) < c(w, x)$, where $w \in \Gamma(x) \setminus \{u, v\}$. Then we can simply set $c(w, x) = c(u, x) + c(v, x)$, since any terminal minimum cut would cut the edges (u, x) and (v, x) instead of the edge (w, x).

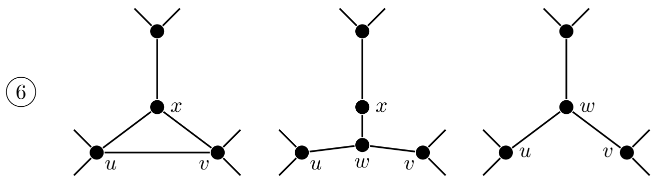

Fig. 2: Edge deletion transformation. Edge capacities are omitted.

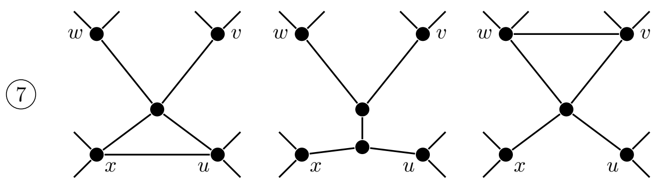

Fig. 3: Edge replacement transformation. Edge capacities are omitted.

serves exactly all terminal minimum cuts. That is, we have the following lemma.

Lemma 5.1. *Let G be a k-terminal graph and G' be a k-terminal graph obtained from G by applying one of the rules $1-5$. Then G' is a quality-1 cut sparsifier of G.*

For our application, it will be useful to enrich the set of rules by introducing two new operations. These operations can be realized as series of the operations 1-5. (See Fig. 2 and 3 for illustrations.)

6. *Edge deletion:* For a degree-three non-terminal with neighbours u, v, the edge (u, v) can be deleted, if it exists. To achieve this, we use a Delta-Wye transformation followed by a series reduction.
7. *Edge replacement:* For a degree-four non-terminal vertex with neighbours x, u, v, w, if the edge (x, u) exists, then it can be replaced by the edge (v, w). To achieve this, we use a Delta-Wye transformation followed by a Wye-Delta transformation.

A k-terminal graph G is *Wye-Delta* reducible to another k-terminal graph H, if G is reduced to H by repeatedly applying one of the operations 1-7.

Lemma 5.2. *Let G and H be k-terminal graphs. Moreover, let G be Wye-Delta reducible to H. Then H is a quality-1 cut sparsifier of G.*

Proof. Observe that the rules 1-7 do not affect any terminal vertex and each rule preserves exactly all terminal minimum cuts by Lemma 5.1. An induction on the number of rules needed to reduce G to H proves the claim. ∎

Grid Graphs. A *grid* graph is a graph with $n \times n$ vertices $\{(u, v) : u, v = 1, \ldots, n\}$, where (u, v) and (u', v') are adjacent if $|u' - u| + |v' - v| = 1$. For $k < n$, a *half-grid* graph with a set K of k terminals is a graph $T_k^n = (V, E)$ with $K \subset V$ and $n(n+1)/2$ vertices $\{(i, j) : i \leq j \text{ and } i, j = 1, \ldots, n\}$, where (i, j) and (i', j') are connected by an edge if $|i' - i| + |j' - j| = 1$, and additional diagonal edges between (i, i) and $(i + 1, i + 1)$ for $i = 1, \ldots, n - 1$. Moreover, each terminal vertex in T_k^n must be one of its diagonal vertices, i.e., for any terminal vertex $x \in K$, it is of the form (m, m) for some $m \in \{1, \ldots, n\}$. Let $\hat{T}_k^n$ be the same graph as T_k^n but excluding the diagonal edges.

Graph Embeddings. Throughout this paper, we will be dealing with the embedding of a planar graph into a square *grid* graph. One way of drawing graphs in the plane are *orthogonal grid-embeddings* [58]. In this setting, the vertices correspond to distinct points and edges consist of alternating sequences of vertical and horizontal segments. Equivalently, one can view this as drawing our input graph as a subgraph of some grid. Formally, a *node-embedding* ρ of $G_1 = (V_1, E_1)$ into $G_2 = (V_2, E_2)$ is an injective mapping that maps V_1 into V_2, and E_1 into paths in G_2, i.e., (u, v) maps to a path from $\rho(u)$ to $\rho(v)$, such that every pair of paths that correspond to two different edges in G_1 is vertex-disjoint (except possibly at the endpoints). Note that if G_2 is a planar graph, then $\rho(G_1)$ and G_1 are also planar. We call ρ an *orthogonal embedding* if G_1 is planar and G_2 is a grid. Moreover, given a planar graph G_1 drawn in the plane, the embedding ρ is called *region-preserving* if $\rho(G_1)$ and G_1 have the same planar topological embedding.

Let $G_1 = (V, E)$ be a k-terminal graph with terminal set K. For any $v \in K$, we will mark $\rho(v)$ as the corresponding terminal in $\rho(G_1)$. Note that a non-terminal vertex in G_1 will not be mapped to a terminal in $\rho(G_1)$ as ρ is injective. That is, there is a one-to-one mapping from K to the terminal set in $\rho(G_1)$. Although the embedding does not consider the edge capacities in G_1, we can still guarantee that such an embedding preserves all terminal minimum cuts, for which we make use of the following operation:

1. *Edge subdivision:* Let (u, v) be an edge of capacity $c(u, v)$. Delete (u, v), introduce a new vertex w and add edges (u, w) and (w, v), each of capacity $c(u, v)$.

The following lemma shows that a node-embedding is a cut preserving mapping.

LEMMA 5.3. *Let G_1 be a k-terminal graph. Let ρ be a node-embedding from G_1 to some grid and $\rho(G_1)$ be a k-terminal graph defined as above. Then $\rho(G_1)$ preserves exactly all terminal minimum cuts of G.*

Proof. We can view each path obtained from the embedding as taking the edge corresponding to the path endpoints in G_1 and performing edge subdivisions finitely many times. We claim that such subdivisions preserve all terminal cuts.

Indeed, let us consider a single edge subdivision for (u, v) (the general claim then follows by induction on the number of edge subdivisions). Fix $S \subset K$ and consider some S-separating minimum cut $(U, V \setminus U)$ in G_1 cutting (u, v). Then, in the transformed graph $\rho(G_1)$, we can simply cut either the edge (u, w) or (w, v). Since by construction, the new edge has the same capacity as the subdivided edge, we get that $\text{cap}_{\rho(G_1)}(\delta_{\rho(G_1)}(\rho(U))) = \text{cap}_{G_1}(\delta_{G_1}(U))$, and in particular $\text{mincut}_{\rho(G_1)}(\rho(S), \rho(K \setminus S)) \leq \text{mincut}_{G_1}(S, K \setminus S)$.

Furthermore, since G_1 is obtained by contracting two edges of the same capacity of $\rho(G_1)$, for any $S \subset K$ and the corresponding $\rho(S)$-separating minimum cut $(U', V \setminus U')$ in $\rho(G_1)$, we have $\text{cap}_{\rho(G_1)}(\delta_{\rho(G_1)}(U')) \geq \text{cap}_{G_1}(\delta_{G_1}(\rho^{-1}(U')))$. This implies that $\text{mincut}_{\rho(G_1)}(\rho(S), \rho(K \setminus S)) \geq \text{mincut}_{G_1}(S, K \setminus S)$. Combining the above gives the lemma. ∎

5.2. Our Construction.

In this section we construct our exact cut sparsifier and prove that any planar k-terminal graph with all terminals lying on the same face admits a cut sparsifier of size $O(k^2)$ that is also planar.

5.2.1. Embedding into Grids.

It is well-known that one can obtain an orthogonal embedding of a planar graph with maximum-degree at most three into a grid (see Valiant [58]). However, our input planar graph can have arbitrarily large maximum degree. In order to be able to make use of such an embedding, we need to first reduce our input graph to a bounded-degree graph while preserving planarity and all terminal minimum cuts. We achieve this by making use of a *vertex splitting* technique, which we describe below.

Given a k-terminal planar graph $G' = (V', E', c')$ with $K \subset V'$ lying on the outer face, vertex splitting produces a k-terminal planar graph $G = (V, E, c)$ with $K \subset V$ such that the maximum degree of G is at most three. Specifically, for each vertex v of degree $d > 3$ with neighboring vertices $u_1, \dots, u_d$, we delete v and introduce new vertices $v_1, \dots, v_d$ along with edges $\{(v_i, v_{i+1}) : i = 1, \dots, d-1\}$, each of capacity $C + 1$, where $C = \sum_{e \in E'} c'(e)$. Further, we replace the edges $\{(u_i, v) : i = 1, \dots, d\}$ with $\{(u_i, v_i) : i = 1, \dots, d\}$, each of corresponding capacity. If v is a terminal vertex, we set one of the v_i's to be a terminal vertex. It follows that the resulting graph G is planar and terminals can be still embedded on the outer face. Note that while the degree of every vertex v_i is at most 3, the degree of any other vertex is not affected.

CLAIM 5.4. *Let G' and G be k-terminal graphs defined as above. Then G preserves exactly all minimum terminal cuts of G', i.e., G is a quality-1 cut sparsifier of G'.*

Proof. It suffices to prove the case where G is obtained from G' by a single vertex splitting. Then the claim follows by induction on the number of vertex splittings required to transform G' to G.

Let $S \subset K$ and $(U, V \setminus U)$ be an S-separating cut in G of size $\text{mincut}_G(S, K \setminus S)$. Suppose towards contradiction that $\delta(U)$ contains an edge of the form (v_j, v_{j+1}), for some j, which in turn gives that $\text{cap}(\delta(U)) \geq C + 1$. Then we can move all the points v_i to one of the sides of the cut $(U, V \setminus S)$ and obtain a new S-separating cut in G of cost at most C, contradicting the fact that $(U, V \setminus U)$ is a minimum terminal cut. Hence, it follows that $\delta(U)$ uses either edges that are in both G and G' or edges of the form (u_i, v_i), which by construction have the same capacity as the edges (u_i, v) in G'. Thus, an S-separating minimum cut in G corresponds to an S-separating minimum cut in G' of the same cost. Since S is chosen arbitrarily, the claim follows. ∎

Let $G = (V, E)$ be a k-terminal graph obtained by vertex splitting of all vertices of degree larger than 3 of $G' = (V', E')$. Further, let $n' = |V'|$, $m' = |E'|$, $n = |V|$ and $m = |E|$. Then it is easy to show that $n \leq 2m'$ and $m \leq m' + n \leq 3m'$. Since G' is planar, we have that $n = O(n')$ and $m = O(n')$. Thus, by just a linear blow-up on the size of vertex and edge sets, we may assume without loss of generality that our input graph is a planar graph of degree at most three.

Valiant [58] and Tamassia et al. [54] showed that a k-terminal planar graph G with n vertices and degree at most three admits an orthogonal region-preserving embedding into some square grid of size $O(n) \times O(n)$. By Lemma 5.3, we know that the resulting graph (with appropriate edge capacities) exactly preserves all terminal minimum cuts of G. We remark that since the embedding is region-preserving, the outer face of the input graph is embedded to the outer face of the grid. Therefore, all terminals in the embedded graph lie on the outer face of the grid. Performing appropriate edge subdivisions, we can make all the terminals lie on the boundary of some possibly larger grid. Further, we can add dummy non-terminals and zero edge capacities to transform our graph into a full-grid H. We observe that the latter does not affect any terminal min-cut. The above leads to the following:

LEMMA 5.5. *Given a k-terminal planar graph G with n vertices, where all termi-*

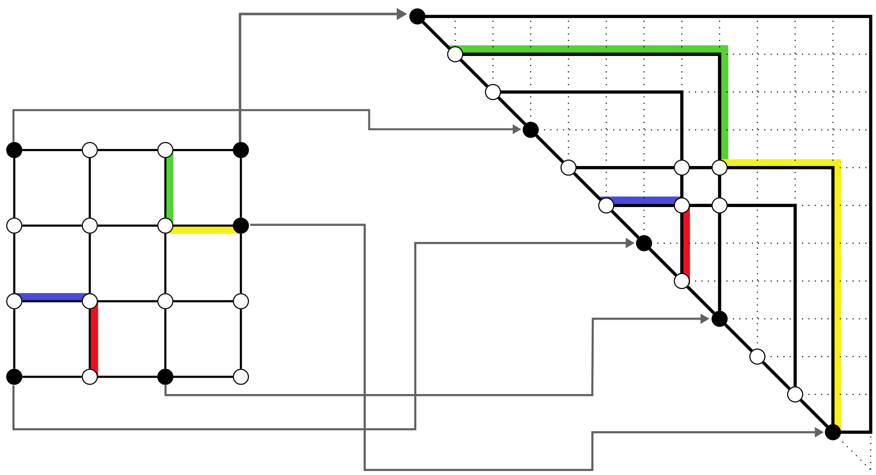

Fig. 4: Embedding grid into half-grid. Black vertices represent terminals while white vertices represent non-terminals. The counter-clockwise ordering starts at the top right terminal. Coloured edges and paths correspond to the mapping of the respective edges: blue for edges $((i, 1), (i, 2))$, red for edges $((n-1, j), (n, j))$, green for edges $((1, j), (2, j))$ and yellow for edges $((i, n-1), (i, n))$, where $i, j = 2, \ldots, n-1$.

nals lie on the outer face, there exists a k-terminal grid graph H, where all terminals lie on the boundary such that H preserves exactly all terminal minimum cuts of G. The resulting graph has $O(n^2)$ vertices and edges.

5.2.2. Embedding Grids into Half-Grids. Next, we show how to embed square grids into half-grid graphs (see Section 2), which will facilitate the application of Wye-Delta transformations. The existence of such an embedding was claimed in the thesis of Gitler [29], but no details on its construction were given.

Let G be a k-terminal square grid on $n \times n$ vertices where terminals lie on the boundary of the grid. We obtain the following:

Lemma 5.6. *There exists a node embedding of the grid G into T_k^ℓ, where $\ell = 4n-3$.*

Proof. Our construction works as follows. We first fix an ordering on the vertices lying on the boundary of the grid in the order induced by the grid. Then we embed each vertex according to that order into the diagonal vertices of the half-grid, along with the edges that form the boundary of the grid. The sub-grid obtained by removing all boundary vertices is embedded appropriately into the upper-part of the half-grid. Finally, we show how to embed edges between the boundary and the sub-grid vertices and argue that such an embedding is indeed vertex-disjoint for any pair of paths. See Fig. 4 for an illustration.

We start with the embedding of the vertices of G. Let us first consider the boundary vertices. The ordering imposed on these vertices can be viewed as starting with the upper-right vertex $(1, n)$ and visiting the rest of vertices in a counter-clockwise direction until reaching the vertex $(2, n)$. We map the vertices on the boundary as follows.

1. For $j = 2, \ldots, n$, the vertex $(1, j)$ is mapped to the vertex $(n-j+1, n-j+1)$,
2. For $i = 1, \ldots, n-1$, the vertex $(i, 1)$ is mapped to the vertex $(n+i-1, n+i-1)$,

3. For $j = 1, \ldots, n-1$, the vertex (n, j) is mapped to the vertex $(2n+j-2, 2n+j-2)$,
4. For $i = 2, \ldots, n$, the vertex (i, n) is mapped to the vertex $(4n-i-2, 4n-i-2)$.

Now we consider the vertices that belong to the induced sub-grid S of G of size $(n-2)^2$ when removing the boundary vertices of our input grid. We map the vertex (i, j) to the vertex $(n+i-1, 2n+j-2)$ for $i, j = 2, \ldots, n-1$. In other words, for every vertex of S we make a vertical shift by $n-1$ units and an horizontal shift by $2n-2$ units. By construction, it is not hard to check that every vertex of G is mapped to a different vertex of T_k^ℓ and all terminal vertices lie on the diagonal of T_k^ℓ.

We continue with the embedding of the edges of G. First, every edge between two boundary vertices in G is embedded to the edge between the corresponding mapped diagonal vertices of T_k^ℓ, except the edge between $(1, n)$ and $(2, n)$. For this edge, we define an edge embedding between the corresponding vertices $(1, 1)$ and $(4n-4, 4n-4)$ of T_k^ℓ by using the path:

$$\begin{aligned}(1,1) &\to (1,2) \to \ldots \to (1, 4n-3) \to (2, 4n-3) \\ &\to \ldots \to (4n-4, 4n-3) \to (4n-4, 4n-4).\end{aligned}$$

Next, every edge of the sub-grid S is embedded in to the edge connecting the mapped endpoints of that edge in T_k^ℓ. In other words, if (i, j) and (i', j') were connected by an edge e in S, then $(n+i-1, 2n+j-2)$ and $(n+i'-1, 2n+j'-2)$ are connected by an edge e' in T_k^ℓ and e is mapped to e'. Finally, the only edges that remain are those connecting a boundary vertex of G with a boundary vertex of S. We distinguish four cases depending on the edge position.

1. For $i = 2, \ldots, n-1$, the edge $((i, 2), (i, 1))$ is mapped to the horizontal path given by:
$$(n+i-1, 2n) \to (n+i-1, 2n-1) \to \ldots \to (n+i-1, n+i-1).$$
2. For $j = 2, \ldots, n-1$, the edge $((n-1, j), (n, j))$ is mapped to the vertical path given by:
$$(2n-2, 2n+j-2) \to (2n-1, 2n+j-2) \to \ldots \to (2n+j-2, 2n+j-2).$$
3. For $j = 2, \ldots, n-1$, the edge $((2, j), (1, j))$ is mapped to the L-shaped path:
$$\begin{aligned}(n+1, 2n+j-2) &\to (n, 2n+j-2) \to \ldots \to (n-j+1, 2n+j-2) \\ &\to (n-j+1, 2n+j-3) \to \ldots \to (n-j+1, n-j+1).\end{aligned}$$
4. For $i = 2, \ldots, n-1$, the edge $((i, n-1), (i, n))$ is mapped to the L-shaped path:
$$\begin{aligned}(n+i-1, 3n-3) &\to (n+i-1, 3n-2) \to \ldots \to (n+i-1, 4n-i-2) \\ &\to (n+i, 4n-i-2) \to \ldots \to (4n-i-2, 4n-i-2).\end{aligned}$$

By construction, it follows that the paths in our edge embedding are vertex disjoint. ∎

5.2.3. Reducing Half-Grids and Bringing the Pieces Together. We now review the construction[8] of Gitler [29], which shows how to reduce half-grids to much

[8] The main motivation of Gitler's study in [29] is to classify graphs that are Wye-Delta reducible. In particular, he used the reductions in this section to prove that any 2-connected plane graph with k terminals on a common face is Wye-Delta reducible to some sub-grid in a triangular shape.

smaller half-grids (excluding diagonal edges) whose size depends only on k. For the sake of completeness, we provide a full proof here. Recall that $\hat{T}_k^n$ is the graph T_k^n without the diagonal edges.

LEMMA 5.7 ([29]). *For any positive k, n with $k < n$, the graph T_k^n with the four vertices $(1,1)$, $(2,2)$, $(n-1,n-1)$ and (n,n) being terminals is Wye-Delta reducible to $\hat{T}_k^k$.*

Proof. We say that two terminals (i,i) and (j,j) are *adjacent* iff $i < j$ and there is no terminal (ℓ,ℓ) such that $i < \ell < j$.

We next describe the reduction procedure. See also Fig. 5 for an illustration. The reduction procedure starts by removing the diagonal edges of T_k^n, thus producing the graph $\hat{T}_k^n$. Specifically, the two edges $((1,1),(2,2))$ and $((n-1,n-1),(n,n))$ are removed using an edge deletion operation. For each remaining diagonal edge of the form $((i,i),(i+1,i+1))$, $i = 2,\dots,n-2$ we repeatedly apply an edge replacement operation until the edge is incident to a boundary vertex $(1,j)$ or (j,n) of the grid, where an edge deletion operation with one of the neighbours of $(1,j)$ resp. (j,n) as vertex x is applied. See Fig. 5(a).

Now, we know that all non-terminals of the form (i,i) are degree-two vertices, thus a series reduction is applied on each of them. This produces new diagonal edges, which are effectively reduced by the above procedure. We keep removing the newly-created degree-two non-terminal vertices and the newly-created edges until no further removals are possible. At this point, all the degree-2 vertices except the top right conner vertices are terminal vertices. See Fig. 5(b).

The resulting graph has a staircase structure, where for every pair of adjacent terminals (i,i) and (j,j), there is a non-terminal (i,j) of degree three or four, namely, the intersection vertex, and a (possibly empty) sequence of degree-three non-terminals that lie on the boundary path from (i,i) to (j,j). For $k = i+1,\dots,j-1$, let (i,k) and (k,j) be the degree-three non-terminals lying on the row and the column subpath, respectively. Additionally, for $k = i+1,\dots,j-1$, let $C_k^i = \{(i',k) : i' = i,\dots,1\}$, resp. $R_k^j = \{(k,j') : j' = j,\dots,n\}$ be the vertices sharing the same column, resp. row with (i,k), resp. (k,j). We next show that the vertices belonging to C_k^i and R_k^j can be removed.

The removal process works as follows. For $k = i+1,\dots,j-1$, we start by choosing a degree 3 vertex (i,k) and its corresponding column C_k^i. Then we apply a Wye-Delta transformation on (i,k), thus creating two new diagonal edges. See Fig. 5(c). Similarly as above, we remove such edges by repeatedly applying an edge replacement operation until they have been pushed to the boundary of the grid, where an edge deletion operation is applied. See Fig. 5(d). In the resulting graph, the vertex $(i-1,k) \in C_k^i$ is now a degree-three non-terminal. We apply the same procedure to this vertex. Applying such a procedure to all remaining vertices of C_k^i, we eliminate a column of the grid. See Fig. 5(e). Symmetrically, the same process applies to the case when we want to remove the row R_k^j corresponding to the vertex (k,j). See Fig. 5(f) - (h).

Applying the above removal process for every adjacent terminal pair and the corresponding degree-three non-terminals, we end up with the graph $\hat{T}_k^k$, where every diagonal vertex is a terminal. See Fig. 5(i). By definition, it follows that $\hat{T}_k^k$ has at most $O(k^2)$ vertices. ∎

Combining the above reductions leads to the following theorem:

THEOREM 5.8. *Let G be a k-terminal planar graph where all terminals lie on the*

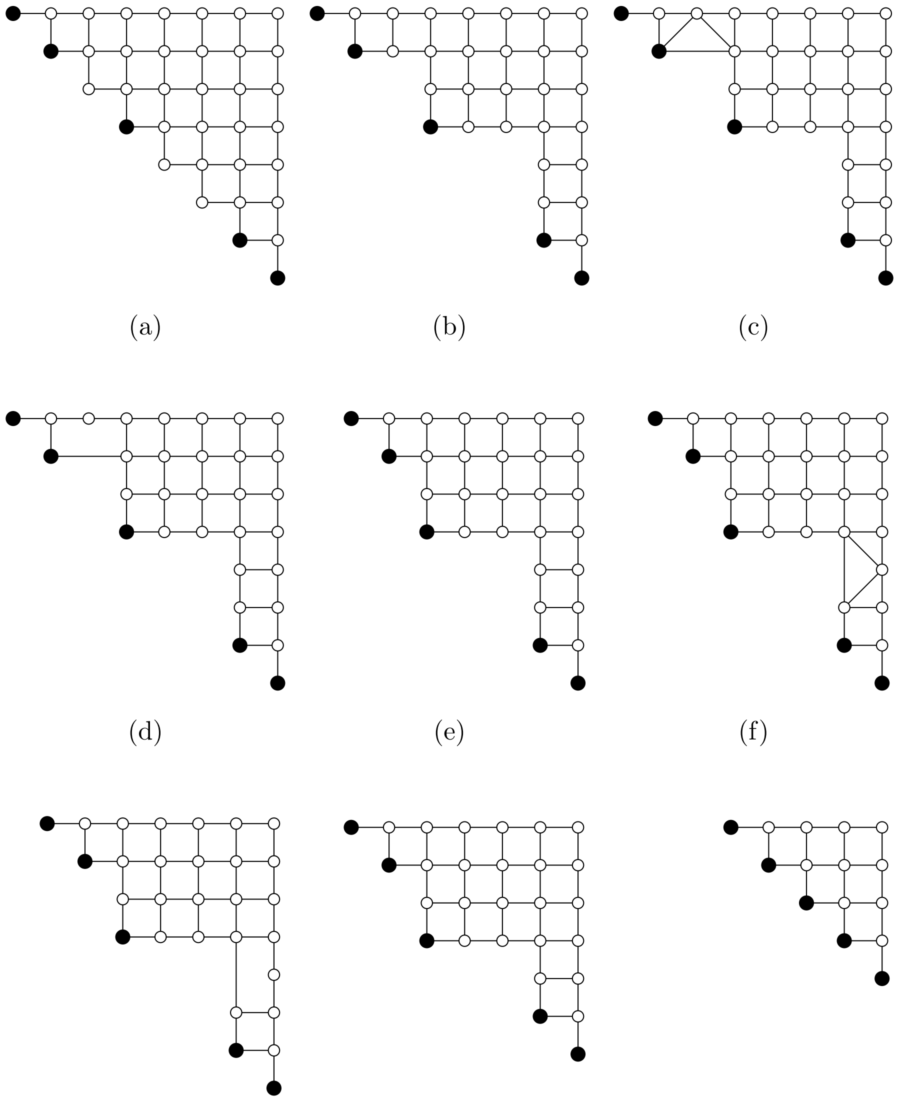

Fig. 5: Half-Grid Reduction.

outer face. Then G admits a quality-1 cut sparsifier of size $O(k^2)$, which is also a planar graph.

Proof. Let n denote the number of vertices in G. First, we apply Lemma 5.5 on G to obtain a grid graph H with $O(n^2)$ vertices, which preserves exactly all terminal minimum cuts of G. We then apply Lemma 5.6 on H to obtain a node embedding ρ into the half-grid T_k^{ℓ}, where $\ell = 4n' - 3$ and $n' = O(n)$ is the width of the grid H. By Lemma 5.3, $\rho(H)$ preserves exactly all terminal minimum cuts of H. We can further extend $\rho(H)$ to the full half-grid T_k^{ℓ}, if dummy non-terminals and zero edge capacities are added. We then mark all the four vertices $(1,1)$, $(2,2)$, $(n-1,n-1)$ and (n,n) in the half grid T_k^ℓ as terminals, if any of them was not. Let the resulting half grid be $T_{k'}^\ell$. Note that $k \leq k' \leq k+4$. Finally, we apply Lemma 5.7 on $T_{k'}^\ell$ to obtain a Wye-Delta reduction to the reduced half-grid graph $\hat{T}_{k'}^{k'}$. It follows by Lemma 5.2 that $\hat{T}_{k'}^{k'}$ is a quality-1 cut sparsifier of $T_{k'}^\ell$, where the size guarantee is immediate from the definition of $\hat{T}_{k'}^{k'}$ and that $k' = \Theta(k)$. ∎

6. Extensions to Planar Flow and Distance Sparsifiers. In this section we show how to extend our result for cut sparsifiers to flow and distance sparsifiers.

6.1. An Upper Bound for Flow Sparsifiers. We first review the notion of Flow Sparsifiers. Let d be a function (called a *demand* function) over terminal pairs in G such that $d(x,x') = d(x',x) \geq 0$ and $d(x,x) = 0$ for all $x, x' \in K$. We denote by $P_{xx'}$ the set of all paths between terminals x and x'. Further, let P_e be the set of all paths using edge e, for all $e \in E$. A *concurrent* (*multi-commodity*) flow f of *throughput* λ is a function over paths among terminal pairs in G such that (1) $f(p) \geq 0$ for any path p, (2) $\sum_{p \in P_{xx'}} f(p) \geq \lambda d(x,x')$, for all distinct terminal pairs $x, x' \in K$, and (3) $\sum_{p \in P_e} f(p) \leq c(e)$, for all $e \in E$. We let $\lambda_G(d)$ denote the *throughput of the concurrent flow* in G that attains the largest throughput and we call a flow achieving this throughput the *maximum concurrent flow.* A graph $H = (V_H, E_H, c_H)$, $K \subset V_H$ is a *quality-q (vertex) flow sparsifier* of G with $q \geq 1$ if for every demand function d, $\lambda_G(d) \leq \lambda_H(d) \leq q \cdot \lambda_H(d)$.

Next we show that given a k-terminal planar graph, where all terminals lie on the outer face, one can construct a quality-1 flow sparsifier of size $O(k^2)$. Our result follows from combining the observation of Andoni et al. [6] for constructing flow-sparsifiers using flow/cut gaps and the flow/cut gap result of Okamura and Seymour [51].

Given a k-terminal graph and a demand function d, recall that $\lambda_G(d)$ is the maximum fraction of d that can be routed in G, and that $\text{cap}(\delta(U))$ is the sum of all capacities of the edges belonging to the cutset $(U, V \setminus U)$. We define the *sparsity* of a cut $(U, V \setminus U)$ to be

$$\Phi_G(U,d) := \frac{\text{cap}(\delta(U))}{\sum_{i,j:|\{i,j\}\cap U|=1} d_{ij}}$$

and the *sparsest cut* as $\Phi_G(d) := \min_{U \subset V} \Phi_G(U,d)$. Then the *flow-cut* gap is given by

$$\gamma(G) := \max\{\Phi_G(d)/\lambda_G(d) : d \in \mathbb{R}_+^{\binom{k}{2}}\}.$$

We will make use of the following theorem:

THEOREM 6.1 ([6]). *Given a k-terminal graph G with terminals K, let G' be a quality-β cut sparsifier for G with $\beta \geq 1$. Then for every demand function $d \in \mathbb{R}_+^{\binom{k}{2}}$,*

$$\frac{1}{\gamma(G')} \leq \frac{\lambda_{G'}(d)}{\lambda_G(d)} \leq \beta \cdot \gamma(G).$$

Therefore, the graph G' with edge capacities scaled up by $\gamma(G')$ is a quality-$\beta \cdot \gamma(G) \cdot \gamma(G')$ flow sparsifier of size $|V(G')|$ for G.

This leads to the following corollary.

COROLLARY 6.2. *Let G be a k-terminal planar graph where all terminals lie on the outer face. Then G admits a quality-1 flow sparsifier of size $O(k^2)$.*

Proof. Given a k-terminal planar graph where all terminals lie on the outer face, Theorem 5.8 shows how to construct a cut sparsifier G' with quality $\beta = 1$ and size $O(k^2)$, which is also a planar graph with all the k terminals lying on the outer face. Okamura and Seymour [51] showed that for every k-terminal planar graph G with terminals lying on the outer face the flow-cut gap is 1. This implies that $\gamma(G) = 1$ and $\gamma(G') = 1$. Invoking Theorem 6.1 we get that G' is a quality-1 flow sparsifier of size $O(k^2)$ for G. ◻

6.2. An Upper Bound for Distance Sparsifiers. We first review the notion of Vertex Distance Sparsifiers. Let $G = (V, E, \ell)$ with $K \subset V$ be a k-terminal graph, where we replace the capacity function c with a length function $\ell : E \to \mathbb{R}_{\geq 0}$. For a terminal pair $(x, x') \in K$, let $d_G(x, x')$ denote the shortest path with respect to the edge lengths ℓ in G. A graph $H = (V', E', \ell')$ is a *quality-q (vertex) distance sparsifier* of G with $q \geq 1$ if for any $x, x' \in K$, $d_G(x, x') \leq d_H(x, x') \leq q \cdot d_G(x, x')$.

Next we argue that a symmetric approach applies to the construction of vertex sparsifiers that preserve distances. Concretely, we prove that given a k-terminal planar graph, where all terminals lie on the outer face, one can construct a quality-1 distance sparsifier of size $O(k^2)$, which is also a planar graph. It is not hard to see that almost all arguments that we used about cut sparsifiers go through, except some adaptations regarding edge lengths in the Wye-Delta rules, edge subdivision operation and vertex splitting operation.

We start adapting the Wye-Delta operations.

1. *Degree-one reduction:* Delete a degree-one non-terminal and its incident edge.
2. *Series reduction:* Delete a degree-two non-terminal y and its incident edges (x, y) and (y, z), and add a new edge (x, z) of length $\ell(x, y) + \ell(y, z)$.
3. *Parallel reduction:* Replace all parallel edges by a single edge whose length is the minimum over all lengths of parallel edges.
4. *Wye-Delta transformation:* Let x be a degree-three non-terminal with neighbours $\Gamma(x) = \{u, v, w\}$. Delete x (along with all its incident edges) and add edges $(u, v), (v, w)$ and (w, u) with lengths $\ell(u, x) + \ell(v, x)$, $\ell(v, x) + \ell(w, x)$ and $\ell(w, x) + \ell(u, x)$, respectively.
5. *Delta-Wye transformation:* Let x, y and z be the vertices of the triangle connecting them. Assume without loss of generality[9] that for any triangle edge (x, y), $\ell(x, y) \leq \ell(x, z) + \ell(y, z)$, where z is the other triangle vertex. Delete the edges of the triangle, introduce a new vertex w and add new edges (w, x), (w, y) and (w, z) with edge lengths $(\ell(x, y) + \ell(x, z) - \ell(y, z))/2$, $(\ell(x, z) + \ell(y, z) - \ell(x, y))/2$ and $(\ell(x, y) + \ell(y, z) - \ell(x, z))/2$, respectively.

The following lemma shows that the above rules preserve exactly all shortest path distances between terminal pairs.

LEMMA 6.3. *Let G be a k-terminal graph and G' be a k-terminal graph obtained from G by applying one of the rules 1-5. Then G' is a quality-*1 *distance sparsifier of G.*

We remark that there is no need to re-define the Edge deletion and replacement operations, since they are just a combination of the above rules. An analogue of Lemma 5.2 can also be shown for distances. We now modify the Edge subdivision operation, which is used when dealing with graph embeddings (see Section 5.1).

[9]Suppose there exists a triangle edge (x, y) with $\ell(x, y) > \ell(x, z) + \ell(y, z)$, where z is the other triangle vertex. Then we can simply set $\ell(x, y) = \ell(x, z) + \ell(y, z)$, since any shortest path between terminal pairs would use the edges (x, z) and (y, z) instead of the edge (x, y).

1. *Edge subdivision*: Let (u, v) be an edge of length $\ell(u, v)$. Delete (u, v), introduce a new vertex w and add edges (u, w) and (w, v), each of length $\ell(u, v)/2$.

We now prove an analogue to Lemma 5.3.

LEMMA 6.4. *Let ρ be a node embedding and let G_1 and $\rho(G_1)$ be k-terminal graphs as defined in Section 5.1. Then $\rho(G_1)$ preserves exactly all shortest path distances between terminal pairs.*

Proof. We can view each path obtained from the embedding as taking the edge corresponding to that path endpoints in G_1 and performing edge subdivisions finitely many times. We claim that such subdivisions preserve all terminal shortest paths.

Indeed, let us consider a single edge subdivison for (u, v) (the general claim then follows by induction on the number of edge subdivions). Fix $x, x' \in K$ and consider some shortest path $p(x, x')$ in G_1 that uses (u, v). We can construct in $\rho(G_1)$ a path $q(x, x')$ of the same length as follows: traverse the subpath $p(x, u)$, traverse the edges (u, w) and (w, v) and finally traverse the subpath $p(v, x')$. It follows that $\sum_{e \in p(x,x')} \ell(e) = \sum_{e \in q(x,x')} \ell(e)$, and thus $d_{\rho(G_1)}(s, t) \leq d_{G_1}(s, t)$.

On the other hand, fix $x, x' \in K$ and consider some shortest path $p'(x, x')$ in $\rho(G_1)$ that uses the two subdivided edges (u, w) and (w, v) (note that it cannot use only one of them). We can construct in G_1 a path $q'(x, x')$ of the same length as follows: traverse the subpath $p'(x, u)$, traverse the edge (u, v) and finally traverse the subpath $p'(v, x')$. It follows that $\sum_{e \in p'(x,x')} \ell(e) = \sum_{e \in q'(x,x')} \ell(e)$ and thus $d_{G_1}(s, t) \leq d_{\rho(G_1)}(s, t)$. Combining the above gives the lemma. ∎

We next consider vertex splitting for graphs whose maximum degree is larger than three. For each vertex v of degree $d > 3$ with $u_1, \ldots, u_d$ adjacent to v, we delete v and introduce new vertices $v_1, \ldots, v_d$ along with edges $\{(v_i, v_{i+1}) : i = 1, \ldots, d-1\}$, each of length 0. Furthermore, we replace the edges $\{(u_i, v) : i = 1, \ldots, d\}$ with $\{(u_i, v_i) : i = 1, \ldots, d\}$, each of corresponding length. If v is a terminal vertex, we make one of the v_i's be a terminal vertex. An analogue to Claim 5.4 gives that the resulting graph preserves all terminal shortest path distances.

We finally note that whenever we add dummy edges of capacity 0 in the cut setting, we replace them by edges of length $D + 1$ in the distance setting, where D is the sum over all edge lengths in the graph we consider. Since any shortest path in the graph does not use the added edges, the terminal shortest path remain unaffected. The above discussion leads to the following theorem.

THEOREM 6.5. *Let G be a k-terminal planar graph where all terminals lie on the outer face. Then G admits a quality-1 distance sparsifier of size $O(k^2)$, which is also a planar graph.*

6.3. Incompressibility of Distances in k-Terminal Graphs. In this section we prove the following incompressibility result (i.e., Theorem 1.5) concerning the trade-off between quality and size of any compression function when estimating terminal distances in k-terminal graphs: for every $\varepsilon > 0$ and $t \geq 2$, there exists a family of (sparse) k-terminal n-vertex graphs such that $k = o(n)$, and that any data structure that approximates pairwise terminal distances within a factor of $t - \varepsilon$ or an additive error $2t - 3$ must use $\Omega(k^{1+1/(t-1)})$ bits of space. Our lower bound is inspired by the work of Matoušek [49], which has also been utilized in the context of distance oracles [57]. Our arguments rely on the recent extremal combinatorics construction (see [18]) that was used to prove lower bounds on the size of distance approximating minors.

Discussion on our result. Note that for any k-terminal graph G, if we do not have any restriction on the structure of the distance sparsifier, then G always admits a trivial quality 1 distance sparsifier H which is the complete weighted graph on k terminals with each edge weight being equal to the distance between the two endpoints in G. Furthermore, by the well-known result of Awerbuch [7], such a graph H in turn admits a multiplicative $(2t-1)$-*spanner* H' with $O(k^{1+1/t})$ edges, that is, all the distances in H are preserved up to a multiplicative factor of $2t-1$ in H', for any $t \geq 1$. This directly implies that the k-terminal graph G has a quality $2t-1$ distance sparsifier with k vertices and $O(k^{1+1/t})$ edges.

We note that *unconditional* lower bounds similar to our result are known for the *number of edges* of spanners, preservers and emulators [44, 45, 60]. Furthermore, as we mentioned, the constructions from [2] imply a stronger lower bound than ours in the setting with additive error $2t-1$ for $t \geq 3$: for a k-terminal n-vertex graph G with $k = o(n^{2/3})$, any data structure that approximates pairwise terminal distances of G within an additive error t needs $\Omega(k^{2-\varepsilon})$ bits, for any $\varepsilon > 0, t = O(n^{\delta})$ and $\delta = \delta(\varepsilon)$. Our constructions are different from [2] and also give lower bounds for multiplicate setting. There are also implicit lower bounds from [3, 34] on the size of data structures for preserving distances of k-terminal graphs with different approximation guarantees.

We start by reviewing a classical notion in combinatorial design.

Definition 6.6 (Steiner Triple System). *Given a ground set $T = [k]$, a $(3,2)$-Steiner system (abbr. $(3,2)$-SS) of T is a collection of 3-subsets of T, denoted by $\mathcal{S} = \{S_1, \dots, S_r\}$, where $r = \binom{k}{2}/3$, such that every 2-subset of T is contained in* exactly *one of the 3-subsets.*

Lemma 6.7 ([59]). *For infinity many k, the set $T = [k]$ admits a $(3,2)$-SS.*

Roughly speaking, our proof proceeds by forming a k-terminal bipartite graph, where terminals lie on one side and non-terminals on the other. The set of non-terminals will correspond to some subset of a Steiner Triple System $\mathcal{S}$, which will satisfy some *certain* property. One can equivalently view such a graph as taking union over *star* graphs. Before delving into details, we need to review a couple of other useful definitions and the construction from [18].

Detour Graph and Cycle. Let k be an integer such that $T = [k]$ admits a $(3,2)$-SS. Let $\mathcal{S}$ be such a $(3,2)$-SS. We define a *detouring graph* $G_{\mathcal{S}}$ with vertex set $\mathcal{S} = \{S_1, \dots, S_r\}$ as follows. By the definition of Steiner system, it follows that $|S_i \cap S_j|$ is either zero or one. Then two vertices S_i and S_j in $G_{\mathcal{S}}$ are adjacent iff $|S_i \cap S_j| = 1$. It is also useful to label each edge (S_i, S_j) with the vertex in $S_i \cap S_j$. We remark that $G_{\mathcal{S}}$ is only an auxiliary graph and has no terminals. A *detouring cycle* is a cycle in the detouring graph such that no two neighbouring edges in the cycle have the same label. Observe that the detouring graph has other cycles which are not detouring cycles.

We have the following lemma which shows that there exists a large induced subgraph in a detouring graph with no short detouring cycles.

Lemma 6.8 ([18]). *For any integer $t \geq 3$, given a detouring graph with vertex set $\mathcal{S}$, there exists a subset $\mathcal{S}' \subset \mathcal{S}$ of cardinality $\Omega(k^{1+1/(t-1)})$ such that the induced graph on $\mathcal{S}'$ has no detouring cycles of size t or less.*

Now we are ready to prove our incompressibility result regarding approximately preserving terminal pairwise distances.

Proof of Theorem 1.5. Let k be an integer such that $T = [k]$ admits a $(3,2)$-SS $\mathcal{S}$. Fix some integer $t \geq 3$, some positive constant c and use Lemma 6.8 on the detouring graph with vertex set $\mathcal{S}$ to construct a subset $\mathcal{S}'$ of $\mathcal{S}$ of size $\Omega(k^{1+1/(t-1)})$ such that the induced graph on $\mathcal{S}'$ has no detouring cycles of size t or less. We may assume without loss of generality that $\ell = |\mathcal{S}'| = c \cdot k^{1+1/(t-1)}$, for some constant $c > 0$ (this can be achieved by removing some elements from $\mathcal{S}'$, as the property concerning the detouring cycles is not destroyed). For each 3-subset S_i in $\mathcal{S}'$, we let $x_1^i, x_2^i, x_3^i \in T$ denote the 3 different numbers in S_i.

We define the k-terminal graph G as follows:

- For each $S_i \in \mathcal{S}'$ create a non-terminal vertex v_i. Let $V_{\mathcal{S}'}$ denote the set of such vertices. The vertex set of G is $T \cup V_{\mathcal{S}'}$, where $T = [k]$ denotes the set of terminals.
- For each $S_i \in \mathcal{S}'$, connect v_i to the three terminals $\{x_1^i, x_2^i, x_3^i\}$ belonging to S_i, i.e., add edges (v_i, x_j^i), $j = 1, 2, 3$.

Note that both the number of vertices and edges of G are $\Theta(\ell + k) = \Theta(k^{1+1/(t-1)})$, and it also holds that $k = \Theta(|V(G)|^{(t-1)/t}) = o(|V(G)|)$.

For any subset $R \subseteq \mathcal{S}'$, we define the subgraph $G_R = (V(G), E_R)$ of G as follows. For each $S_i \in \mathcal{S}'$, if $S_i \in R$, perform no changes. If $S_i \notin R$, delete the edge (v_i, x_1^i). Note that there are 2^ℓ subgraphs G_R. We let $\mathcal{G}$ denote the family of all such subgraphs.

We say a terminal pair (x, x') *respects* $\mathcal{S}'$ if in the $(3,2)$-SS $\mathcal{S}$, the unique 3-subset S that contains x and x' belongs to $\mathcal{S}'$. Given $R \subseteq \mathcal{S}'$ and some terminal pair (x, x'), we say that R *covers* (x, x') if both x and x' are connected to some non-terminal v in G_R.

CLAIM 6.9. *For all $R \subseteq \mathcal{S}'$ and terminal pairs (x, x') covered by R we have that $d_{G_R}(x, x') = 2$.*

Proof. By the definition of Steiner system and the construction of G_R, the shortest path between x and x' is simply a 2-hop path, i.e., $d_{G_R}(x, x') = 2$. ∎

CLAIM 6.10. *For all $R \subseteq \mathcal{S}'$ and any terminal pair (x, x') that respects $\mathcal{S}'$ and is* not *covered by R, we have that $d_{G_R}(x, x') \geq 2t$.*

Proof. Since (x, x') respects $\mathcal{S}'$, there exists $S_i = (x_1^i, x_2^i, x_3^i) \in \mathcal{S}'$ that contains both x and x'. By construction of G_R and the fact that (x, x') is not covered by R, it follows that $S_i \in \mathcal{S}' \setminus R$, and one of x, x' corresponds to x_1^i and the other corresponds to x_2^i or x_3^i. Without loss of generality, we assume $x = x_1^i$ and $x' = x_2^i$. Note that there is no edge in G_R connecting x_1^i with the non-terminal v_i that corresponds to S_i. Since any simple path p between x_1^i and x_2^i in G will use visit each terminal at most once, it corresponds to paths in the detouring graph $G_{\mathcal{S}}$ such that no two neighbouring edges have the same label. Now by Lemma 6.8, the detouring graph induced on $\mathcal{S}'$ has no detouring cycles of size t or less, which implies that any simple path between x_1^i and x_2^i in G must pass through at least $t-1$ other terminals. Let $w_1, \ldots, w_{t-1}$ be such terminals and let $P := x_1^i \rightarrow w_1, \ldots, w_{t-1} \rightarrow x_2^i$ denote the corresponding path, ignoring the non-terminals along the path. Between any consecutive terminal pairs in P, the shortest path is at least 2. Thus, the length of P is at least $2t$, i.e., $d_{G_R}(x_1^i, x_2^i) \geq 2t$. ∎

Fix any two subsets $R_1, R_2 \subseteq \mathcal{S}'$ with $R_1 \neq R_2$. It follows that there exists a 3-subset $S_i = (x_1^i, x_2^i, x_3^i) \in \mathcal{S}'$ such that either $S_i \in R_1 \setminus R_2$ or $S_i \in R_2 \setminus R_1$. Assume without loss of generality that $S_i \in R_2 \setminus R_1$, i.e., (x_1^i, x_2^i) respects $\mathcal{S}'$ and it is covered by R_2 but not by R_1. By Claim 6.9 and 6.10, it holds that $d_{G_{R_2}}(x_1^i, x_2^i) = 2$ and $d_{G_{R_1}}(x_1^i, x_2^i) \geq 2t$.

Since R_1, R_2 are two arbitrary subsets of $\mathcal{S}'$, it holds that there exists a set $\mathcal{G}$ of 2^ℓ different subgraphs on the same set of nodes $V(G)$ satisfying the following property:

for any $G_1, G_2 \in \mathcal{G}$, there exists a terminal pair (x, x') such that the distances between x and x' in G_1 and G_2 differ by at least a t factor as well as by at least $2t-2$.

Assume on the contrary that there exists a compression function that approximates terminal path that preserves terminal distances within a $t-\varepsilon$ factor or an additive error $2t-3$ and uses less than ℓ bits of space. Since there are 2^ℓ graphs in $\mathcal{G}$, two different graphs $G_1, G_2 \in \mathcal{G}$ will map to the same bit string. However, since there exists a pair x, x' such that the distances between them in G_1 and G_2 differ by at least a t factor and by at least $2t-2$, G_1 and G_2 should be mapped to two different bit strings. This is a contradiction. Therefore, any such compression must use at least $\Omega(\ell) = \Omega(k^{1+1/(t-1)})$ bits of space.

To complete the proof of Theorem 1.5, we need to show the claim for quality $t = 2$. The only significant modification we need is the usage of a $(3,2)$-SS in the construction of graph G (instead of using a subset of it). The remaining details are similar to the above proof and we omit them here.

Appendix A. Proof of Theorem 4.1.

Throughout, given a directed graph G, we say that G is *disoriented* if we forget the orientation of edges in G and treat G as an undirected graph. We next give the definition of "2-layered" graphs and "2-layered" spanning trees. These definitions allow us to reduce reachability in G to reachability in some digraphs with special properties.

DEFINITION A.1. *Given a digraph H, and an integer parameter $t \geq 1$, a t-*layered *spanning tree T in H is a disoriented rooted spanning tree such that any path in T from the root is a concatenation of at most t directed paths in H. If H has such a t-layered spanning tree, then we say that H is a t-*layered *digraph.*

Proof of Theorem 4.1. Assume without loss of generality that G is connected in the undirected sense; otherwise we can apply the construction we are about to describe separately to each connected component.

Our construction starts by partitioning the vertices of G into layers $L_0, \ldots, L_b$, where $b = O(k^3)$, as follows: L_0 is the set of vertices reachable from an arbitrary vertex v_0, and layer L_i consists of all vertices reaching or reachable from the previous layers, depending on whether the index i is even or odd. Formally, for $i > 0$, we have

$$L_i = \begin{cases} \{v \in V \setminus L_{<i} \mid v \rightsquigarrow L_{<i}\} & \text{if } i \text{ is odd} \\ \{v \in V \setminus L_{<i} \mid L_{<i} \rightsquigarrow v\} & \text{if } i \text{ is even,} \end{cases}$$

where $L_{<i} := \bigcup_{j<i} L_j$. Similarly, let $L_{\leq i} := \bigcup_{j \leq i} L_j$ and define k to be the first index such that $L_{\leq k} = V$. For each vertex v, we also defined an index $\iota(v)$ with $\iota(v) = i$, if $v \in L_i$.

We construct the digraph G_i by taking two consecutive layers and contracting all preceding layers into a single vertex, i.e., G_i is constructed by first taking the subgraph of G induced by $L_{\leq i+1}$, and for $i > 0$, contracting all vertices in $L_{<i}$ to the single *root* vertex r_i. For G_0, we set $r_0 = v_0$.

We next discuss the properties of $G_i's$. By construction, G_i's satisfy Item 5. Moreover, since the layering forms a partitioning, each vertex occurs as a non-root vertex at most twice over all G_i's. Similarly, every edge occurs at most twice, thus proving Item 1. The claimed bound on the number of vertices and edges over all G_i's follows since (i) there are at most $b \leq n' = O(k^3)$ root vertices and (i) there can be at most $2n'$ edges incident to the roots.

Consider Item 2, and let R be any directed path from a vertex s to a vertex t. Let i be the smallest index of a layer that intersects R, and let x be a vertex in the intersection. By definition, if $j \geq i$ is even, then $L_{\leq j}$ contains the part of R after x, and if $j \geq i$ is odd, then $L_{\leq j}$ contains the part of R before x. Thus R is contained in $L_i \cup L_{i+1}$. By construction of G_i's, it follows that R is contained in G_i. Note that $s \in R$ is either contained in $G_{\iota(s)-1}$ or $G_{\iota(s)}$, so the path R from s to t is either contained in one of these two digraphs.

To see that Item 3 is satisfied, we first need to show that each G_i is a 2-layered digraph, i.e., it admits a 2-layered spanning tree with root r_i. To this end, assume without loss of generality that i is odd. By definition, r_i reaches every vertex in L_i, so a spanning tree U_i of $\{r_i\} \cup L_i$ can be constructed with edges oriented away from r_i. Moreover, since $\{r_i\} \cup L_i$ is reached by all vertices in L_{i+1}, we can extend U_i to a spanning tree T_i of $\{r_i\} \cup L_i \cup L_{i+1} = V(G_i)$ with the new edges oriented towards $\{r_i\} \cup L_i$. Note that any path in T_i from r_i has a first part oriented away from r_i and the other part oriented towards r_i, so T_i is 2-layered.

Now we make use of the following result from [56]. Given a rooted tree T in an undirected graph and a vertex v, we let $T(v)$ denote the path between the root of T and v.

Lemma A.2 (Lemma 2.3. [56]). *Given an undirected planar graph H with a rooted spanning tree T and non-negative vertex weights, we can find three vertices u, v and w such that each component of $H \setminus V(T(u) \cup T(v) \cup T(w))$ has at most half the weight of H.*

The above lemma shows that an undirected planar graph H with a rooted spanning tree T admits a vertex separator, which consists of three paths starting at the root in T, whose removal separates H into components of at most half its size.

Applying Lemma A.2 to each digraph G_i (when forgetting about the orientation of its edges) with the 2-layered spanning tree T_i rooted at r_i, we have that there are at most six directed paths in the digraph G_i whose removal separates G_i into components of at most half its size. Note that if S_i is the set of 6 directed paths corresponding the 1/2-separator, then S_i induces a connected subgraph of the underlying undirected graph G_i. This finishes the proof of Item 3.

Finally, Item 4 follows by construction and this finishes the proof of Theorem 4.1. ∎

REFERENCES

[1] A. Abboud and G. Bodwin, *The 4/3 additive spanner exponent is tight*, in Proc. of the 48th STOC, 2016, pp. 351–361.

[2] A. Abboud and G. Bodwin, *Reachability preservers: New extremal bounds and approximation algorithms*, in Proc. of the 29th SODA, 2018. available at Arxiv: CoRR abs/1710.11250.

[3] A. Abboud, G. Bodwin, and S. Pettie, *A hierarchy of lower bounds for sublinear additive spanners*, in Proc. of the 28th SODA, 2017, pp. 568–576.

[4] A. Abboud, P. Gawrychowski, S. Mozes, and O. Weimann, *Near-optimal compression for the planar graph metric*, in Proc. of the 29th SODA, 2018. available at arXiv: CoRR abs/1703.04814.

[5] A. V. Aho, M. R. Garey, and J. D. Ullman, *The transitive reduction of a directed graph*, SIAM J. Comput., 1 (1972), pp. 131–137.

[6] A. Andoni, A. Gupta, and R. Krauthgamer, *Towards $(1+\varepsilon)$-approximate flow sparsifiers*, in Proc. of the 25th SODA, 2014, pp. 279–293.

[7] B. Awerbuch, *Complexity of network synchronization*, J. ACM, 32 (1985), pp. 804–823.

[8] A. A. Benczúr and D. R. Karger, *Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time*, in Proc. of the 28th STOC, 1996, pp. 47–55.

[9] G. Bodwin, *Linear size distance preservers*, in Proc. of the 28th SODA, 2017, pp. 600–615.

[10] T.-H. H. Chan, D. Xia, G. Konjevod, and A. Richa, *A tight lower bound for the steiner point removal problem on trees*, in Proc. of the 9th APPROX/RANDOM, 2006, pp. 70–81.
[11] H.-C. Chang, P. Gawrychowski, S. Mozes, and O. Weimann, *Near-optimal distance emulator for planar graphs*, in Proc. of the 26th ESA, 2018.
[12] M. Charikar, T. Leighton, S. Li, and A. Moitra, *Vertex sparsifiers and abstract rounding algorithms*, in Proc. of the 51th FOCS, 2010, pp. 265–274.
[13] S. Chaudhuri, K. Subrahmanyam, F. Wagner, and C. D. Zaroliagis, *Computing mimicking networks*, Algorithmica, 26 (2000), pp. 31–49.
[14] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair, *Embedding k-outerplanar graphs into l1*, SIAM J. Discrete Math., 20 (2006), pp. 119–136.
[15] C. Chekuri, S. Khanna, and F. B. Shepherd, *Edge-disjoint paths in planar graphs with constant congestion*, SIAM J. Comput., 39 (2009), pp. 281–301.
[16] C. Chekuri, F. B. Shepherd, and C. Weibel, *Flow-cut gaps for integer and fractional multiflows*, in Proc. of the 21st SODA, 2010, pp. 1198–1208.
[17] Y. K. Cheung, *Steiner point removal - distant terminals don't (really) bother*, in Proc. of the 29th SODA, 2018. available at Arxiv: CoRR abs/1703.08790.
[18] Y. K. Cheung, G. Goranci, and M. Henzinger, *Graph minors for preserving terminal distances approximately - Lower and Upper Bounds*, in Proc. of the 43rd ICALP, 2016, pp. 131:1–131:14.
[19] J. Chuzhoy, *On vertex sparsifiers with steiner nodes*, in Proc. of the 44th STOC, 2012, pp. 673–688.
[20] J. Chuzhoy, *Routing in undirected graphs with constant congestion*, in Proc. of the 44th STOC, 2012, pp. 855–874.
[21] D. Coppersmith and M. Elkin, *Sparse sourcewise and pairwise distance preservers*, SIAM J. Discrete Math., 20 (2006), pp. 463–501.
[22] E. B. Curtis, D. Ingerman, and J. A. Morrow, *Circular planar graphs and resistor networks*, Linear algebra and its applications, 283 (1998), pp. 115–150.
[23] K. Däubel, Y. Disser, M. Klimm, T. Mütze, and F. Smolny, *Distance-preserving graph contractions*, CoRR, abs/1705.04544 (2017).
[24] K. Diks and P. Sankowski, *Dynamic plane transitive closure*, in Proc. of the 15th ESA, 2007, pp. 594–604.
[25] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar, *Vertex sparsifiers: New results from old techniques*, SIAM J. Comput., 43 (2014), pp. 1239–1262.
[26] T. A. Feo and J. S. Provan, *Delta-wye transformations and the efficient reduction of two-terminal planar graphs*, Operations Research, 41 (1993), pp. 572–582.
[27] A. Filtser, *Steiner point removal with distortion o(log k)*, in Proc. of the 29th SODA, 2018. available at Arxiv: CoRR abs/1706.08115.
[28] K. Gajjar and J. Radhakrishnan, *Distance-preserving subgraphs of interval graphs*, in Proc. of the 25th ESA, 2017, pp. 39:1–39:13.
[29] I. Gitler, *Delta-Wye-Delta Transformations: Algorithms and Applications*, PhD thesis, Department of Combinatorics and Optimization, University of Waterloo, 1991.
[30] G. Goranci, M. Henzinger, and P. Peng, *Improved guarantees for vertex sparsification in planar graphs*, in Proc. of the 25th ESA, 2017, pp. 45:1–45:14.
[31] G. Goranci and H. Räcke, *Vertex sparsification in trees*, in Proc. of the 14th WAOA, 2016, pp. 103–115.
[32] A. Gupta, *Steiner points in tree metrics don't (really) help*, in Proc. of the 12th SODA, 2001, pp. 220–227.
[33] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde, *Characterizing multiterminal flow networks and computing flows in networks of small treewidth*, J. Comput. Syst. Sci., 57 (1998), pp. 366–375.
[34] S.-E. Huang and S. Pettie, *Lower Bounds on Sparse Spanners, Emulators, and Diameter-reducing shortcuts*, in 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018), vol. 101, 2018, pp. 26:1–26:12.
[35] L. Kamma, R. Krauthgamer, and H. L. Nguyen, *Cutting corners cheaply, or how to remove steiner points*, SIAM J. Comput., 44 (2015), pp. 975–995.
[36] N. Karpov, M. Pilipczuk, and A. Zych-Pawlewicz, *An exponential lower bound for cut sparsifiers in planar graphs*, The 12th International Symposium on Parameterized and Exact Computation (IPEC), (2017).
[37] I. Katriel, M. Kutz, and M. Skutella, *Reachability substitutes for planar digraphs*, in Technical Report MPI-I-2005-1-002, Max-Planck-Institut für Informatik, 2005.
[38] A. E. Kennelly, *The equivalence of triangles and three-pointed stars in conducting networks*,

Electrical world and engineer, 34 (1899), pp. 413–414.

[39] A. Khan and P. Raghavendra, *On mimicking networks representing minimum terminal cuts*, Inf. Process. Lett., 114 (2014), pp. 365–371.

[40] R. Krauthgamer, H. L. Nguyen, and T. Zondiner, *Preserving terminal distances using minors*, SIAM J. Discrete Math., 28 (2014), pp. 127–141.

[41] R. Krauthgamer and I. Rika, *Mimicking networks and succinct representations of terminal cuts*, in Proc. of the 24th SODA, 2013, pp. 1789–1799.

[42] R. Krauthgamer and I. Rika, *Refined vertex sparsifiers of planar graphs*, CoRR, abs/1702.05951 (2017).

[43] R. Krauthgamer and T. Zondiner, *Preserving terminal distances using minors*, in Proc. of the 39th ICALP, 2012, pp. 594–605.

[44] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar, *A new series of dense graphs of high girth*, Bulletin of the American mathematical society, 32 (1995), pp. 73–79.

[45] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar, *A characterization of the components of the graphs d (k, q)*, Discrete Mathematics, 157 (1996), pp. 271–283.

[46] J. R. Lee, M. Mendel, and M. Moharrami, *A node-capacitated okamura-seymour theorem*, in Proc. of the 45th STOC, 2013, pp. 495–504.

[47] F. T. Leighton and A. Moitra, *Extensions and limits to vertex sparsification*, in Proc. of the 42nd STOC, 2010, pp. 47–56.

[48] K. Makarychev and Y. Makarychev, *Metric extension operators, vertex sparsifiers and lipschitz extendability*, in Proc. of the 51th FOCS, 2010, pp. 255–264.

[49] J. Matoušek, *On the distortion required for embedding finite metric spaces into normed spaces*, Israel Journal of Mathematics, 93 (1996), pp. 333–344.

[50] A. Moitra, *Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size*, in Proc. of the 50th FOCS, 2009.

[51] H. Okamura and P. D. Seymour, *Multicommodity flows in planar graphs*, J. Comb. Theory, Ser. B, 31 (1981), pp. 75 – 81.

[52] D. A. Spielman and S. Teng, *Spectral sparsification of graphs*, SIAM J. Comput., 40 (2011), pp. 981–1025.

[53] S. Subramanian, *A fully dynamic data structure for reachability in planar digraphs*, in Proc. of the 1st ESA, 1993, pp. 372–383.

[54] R. Tamassia and I. G. Tollis, *Planar grid embedding in linear time*, IEEE Trans. Circuits Syst., 36 (1989), pp. 1230–1234.

[55] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

[56] M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*, J. ACM, 51 (2004), pp. 993–1024.

[57] M. Thorup and U. Zwick, *Approximate distance oracles*, J. ACM, 52 (2005), pp. 1–24.

[58] L. G. Valiant, *Universality considerations in VLSI circuits*, IEEE Trans. Computers, 30 (1981), pp. 135–140.

[59] R. M. Wilson, *An existence theory for pairwise balanced designs, III: proof of the existence conjectures*, J. Comb. Theory, Ser. A, 18 (1975), pp. 71–79.

[60] D. P. Woodruff, *Lower bounds for additive spanners, emulators, and more*, in Proc. of the 47th FOCS, 2006, pp. 389–398.