Cut-Preserving Vertex Sparsifiers for Planar and Quasi-bipartite Graphs

Yu Chen*   Zihan Tan†

September 4, 2025

Abstract

We study vertex sparsification for preserving cuts. Given a graph G with a subset $|T| = k$ of its vertices called terminals, a *quality-q cut sparsifier* is a graph G' that contains T, such that, for any partition (T_1, T_2) of T into non-empty subsets, the value of the min-cut in G' separating T_1 from T_2 is within factor q from the value of the min-cut in G separating T_1 from T_2. The construction of cut sparsifiers with good (small) quality and size has been a central problem in graph compression for years.

Planar graphs and quasi-bipartite graphs are two important special families studied in this research direction. The main results in this paper are new cut sparsifier constructions for them in the high-quality regime (where $q = 1$ or $1 + \varepsilon$ for small $\varepsilon > 0$).

We first show that every planar graph admits a planar quality-$(1 + \varepsilon)$ cut sparsifier of size $\tilde{O}(k/\mathsf{poly}(\varepsilon))$, which is in sharp contrast with the lower bound of $2^{\Omega(k)}$ for the quality-1 case.

We then show that every quasi-bipartite graph admits a quality-1 cut sparsifier of size $2^{\tilde{O}(k^2)}$. This is the second to improve over the doubly-exponential bound for general graphs (previously only planar graphs have been shown to have single-exponential size quality-1 cut sparsifiers).

Lastly, we show that contraction, a common approach for constructing cut sparsifiers adopted in most previous works, does not always give optimal bounds for cut sparsifiers. We demonstrate this by showing that the optimal size bound for quality-$(1 + \varepsilon)$ contraction-based cut sparsifiers for quasi-bipartite graphs lies in the range $[k^{\tilde{\Omega}(1/\varepsilon)}, k^{O(1/\varepsilon^2)}]$, while in previous work an upper bound of $\tilde{O}(k/\varepsilon^2)$ was achieved via a non-contraction approach.

Contents

1   Introduction

Given a graph G and a set T of its vertices called *terminals*, a *cut sparsifier* of G with respect to T is a graph G' with $T \subseteq V(G')$, such that for every partition (T_1, T_2) of T into non-empty subsets, the size of the minimum cut separating T_1 from T_2 in G and the size of the minimum cut separating T_1 from T_2 in G' are within some small multiplicative factor $q \geq 1$, which is also called the *quality* of the sparsifier.

Karger [Kar93, Kar99] first considered the special case where $T = V(G) = V(G')$ and G' is required to be a subgraph of G, and he used sampling approaches to construct quality-$(1 + \varepsilon)$ cut sparsifiers with $O(n \log n / \varepsilon^2)$ edges. Such sparsifiers are called *edge sparsifiers*, since we are only allowed to sparsify the edges (not the vertices) and the goal is to minimize $|E(G')|$. In this paper, we allow G' to have a different set of vertices, and such graphs G' are known as *vertex sparsifiers*. Constructing cut-preserving vertex sparsifiers with small quality and size (measured by the number of vertices in G') has been a central problem in graph compression.

Cut-preserving vertex sparsifiers were first studied in the special case where G' is only allowed to contain terminals (that is, $V(G') = T$). Moitra [Moi09] first showed that every k-terminal graph has a cut sparsifier with quality $O(\log k / \log \log k)$. Then Charikar, Leighton, Li and Moitra [CLLM10] showed that such sparsifiers can be computed efficiently. The best quality lower bound in this case is $\Omega(\sqrt{\log k} / \log \log k)$ [MM10, CLLM10], leaving a small gap to be closed. Therefore, the natural next question, which has been a central question on cut/flow sparsifiers over the past decade, is:

> *Can better quality sparsifiers be achieved by allowing a small number of Steiner vertices?*

For exact cut sparsifiers ($q = 1$), it has been shown that every k-terminal graph admits an exact cut sparsifier of size at most 2^{2^k} [HKNR98, KR14], while the strongest lower bound is $2^{\Omega(k)}$ [KR14, KPZP17], leaving an exponential gap yet to be closed. Since then the research on cut sparsifiers was diverted into two directions. The first direction focuses on proving better size upper bounds for special families of graphs. For example, for planar graphs, Krauthgamer and Rika [KR13, KR17] showed that $2^{O(k)}$ vertices are sufficient for an exact cut sparsifier, and then Karpov, Philipzcuk and Zych-Pawlewicz [KPZP17] proved a lower bound of $2^{\Omega(k)}$, showing that the single exponential upper bound is almost tight. Another example is graphs where each terminal has degree exactly 1. For these graphs, Chuzhoy [Chu12] showed the construction for $O(1)$-quality sparsifiers of size $O(k^3)$ by contracting expanders, and Kratsch and Wahlstrom [KW12] constructed quality-1 sparsifiers of size $O(k^3)$ via a matroid-based approach. The second direction focuses on constructing cut sparsifiers with a slightly worse quality of $(1 + \varepsilon)$ for small $\varepsilon > 0$. Andoni, Gupta, and Krauthgamer [AGK14] initiated the study of quality-$(1+\varepsilon)$ flow sparsifiers [1] (which are stronger versions of cut sparsifiers), established a framework of constructing flow sparsifiers via contractions with a doubly exponential upper bound. However, the only family that they managed to obtain a better-than-2^{2^k} upper bound for is the family of quasi-bipartite graphs, graphs where every edge is incident to some terminal. They showed an upper bound of $\mathsf{poly}(k/\varepsilon)$, via a sampling-based approach. For quality-$(1 + \varepsilon)$ cut sparsifiers on quasi-bipartite graphs, the bound was improved to $\tilde{O}(k/\varepsilon^2)$ by Jambulapati, Lee, Liu and Sidford [JLLS23], also via a sampling-based approach.

---

[1] Flow sparsifiers are concretely related to cut sparsifiers. We review the previous work and discuss the connection between them in Section 1.3.

Among the above algorithmic results, the most common approach for constructing cut sparsifiers is contraction. Specifically, we compute a partition $\mathcal{F}$ of the vertices of G into disjoint subsets, and then contract each subset of $\mathcal{F}$ into a supernode. The resulting graph is called a *contraction-based cut sparsifier*. Contraction-based cut sparsifiers may only increase the value of every terminal min-cut, which partly explains why contraction is a commonly adopted approach, as to analyze its quality it suffices to show that the min-cuts are not increased by too much. For general graphs, a lower bound of $2^{2^{\Omega(k)}}$ (almost matching the upper bound of 2^{2^k}) was proved [KPZP17] for contraction-based cut sparsifiers. It is therefore natural to ask whether or not better bounds can be achieved for special graphs. Moreover, all the previous constructions of cut sparsifiers with Steiner nodes, except for the ones using matroid-based/sampling-based approaches, are contraction-based. Therefore, it is natural to explore if contraction can give optimal constructions for cut sparsifiers.

1.1 Our Results

In this paper, we make progress on the above questions. Our first results, summarized as Theorem 1 and Theorem 2, are new size upper bounds on exact or quality-$(1+\varepsilon)$ sparsifiers for special families of graphs.

Theorem 1. *For every integer $k \geq 1$ and real number $\varepsilon > 0$, every planar graph with k terminals admits a quality-$(1+\varepsilon)$ cut sparsifier on $O(k \cdot \mathsf{poly}(\log k/\varepsilon))$ vertices, which is also a planar graph.*

For planar graphs, our near-linear size bound is in sharp contrast with the single-exponential bound $2^{\Theta(k)}$ [KR13, KR17, KPZP17] for exact cut sparsifiers. In other words, Theorem 1 shows that, for preserving cuts in planar graphs, we can trade a small loss in quality for a significant improvement in size.

Theorem 2. *For every integer $k \geq 1$, every quasi-bipartite graph with k terminals admits an exact (quality-1) contraction-based cut sparsifier of size $2^{O(k^2 \log k)}$, and there exists a quasi-bipartite graph with k terminals whose exact contraction-based cut sparsifier must contain $\Omega(2^k)$ vertices.*

Theorem 2 shows that, similar to planar graphs, the family of quasi-bipartite graphs also admit a single-exponential upper bound (even achievable by contraction-based sparsifiers), better than the doubly-exponential bound for general graphs [HKNR98, KR14, KPZP17].

Next we turn to study contraction-based sparsifiers. Our next result shows that contraction-based sparsifiers are not optimal constructions, as they give worse bounds for quasi-bipartite graphs.

Theorem 3. *For every integer $k \geq 1$ and real number $\varepsilon > 0$, there exists a quasi-bipartite graph with k terminals, whose quality-$(1+\varepsilon)$ contraction-based cut sparsifier must contain $k^{\tilde{\Omega}(1/\varepsilon)}$ vertices.*

Compared with the previous bounds $\mathsf{poly}(k/\varepsilon)$ [AGK14] and $\tilde{O}(k/\varepsilon^2)$ [JLLS23] (both obtained by sampling-based and therefore not contraction-based approaches), Theorem 3 shows that contraction-based sparsifiers are sometimes provably suboptimal.

We then proceed to study the size upper bound for quality-$(1+\varepsilon)$ contraction-based cut sparsifiers of quasi-bipartite graphs. Our last result, summarized in Theorem 4, shows that when ε is a constant, quasi-bipartite graphs have $\mathsf{poly}(k)$-sized contraction-based cut sparsifiers, complementing the lower bound in Theorem 3, in contrast with the $\Omega(2^k)$ lower bound for exact sparsifiers in Theorem 2.

Theorem 4. *For every integer $k \geq 1$ and real number $\varepsilon > 0$, every quasi-bipartite graph with k terminals admits a quality-$(1 + \varepsilon)$ contraction-based cut sparsifier of size $k^{O(1/\varepsilon^2)} \cdot f(\varepsilon)$, where f is a function that only depends on ε.*

We summarize our results and provide some comparison with previous work in Table 1.

Graph Type	Quality	Size	Contraction-based?	Citation
	1	2^{2^k}	Yes	[HKNR98, KR14]
General	1	$2^{\Omega(k)}$	No	[KPZP17, KR14]
	1	$2^{2^{\Omega(k)}}$	Yes	[KPZP17]
	1	$2^{O(k)}$	Yes	[KR13, KR17]
Planar	1	$2^{\Omega(k)}$	No	[KPZP17]
	$1 + \varepsilon$	$O(k \cdot \mathsf{poly}(\log k/\varepsilon))$	No	Theorem 1
	1	$2^{O(k^2 \log k)}, \Omega(2^k)$	Yes	Theorem 2
Quasi-bipartite	$1 + \varepsilon$	$\tilde{O}(k/\varepsilon^2)$	No	[JLLS23]
	$1 + \varepsilon$	$k^{\tilde{\Omega}(1/\varepsilon)}, k^{O(1/\varepsilon^2)} f(\varepsilon)$	Yes	Theorems 3 and 4

Table 1: Results on exact or quality-$(1 + \varepsilon)$ cut sparsifiers for planar and quasi-bipartite graphs

1.2 Technical Overview

Planar cut sparsifiers. Our construction of planar cut sparsifiers follows a similar framework as the planar distance emulator construction in [CKT22], based on the intuition that, if we take the dual graph, then min-cuts become min-length partitioning cycles and therefore can be approximated by distance-based approaches. The algorithm consists of three steps:

1. reduce general planar graphs to $O(1)$-face instances (where terminals lie on $O(1)$ faces in the planar embedding of the graph);

2. reduce $O(1)$-face instances to 1-face instances (where all terminals lie on the boundary of a single face; such graphs are also known as *Okamura-Seymour instances*); and

3. construct quality-$(1 + \varepsilon)$ planar cut sparsifiers for 1-face instances.

Step 1 and Step 3 are implemented in a similar way as [CKT22]. Step 2 requires several new ideas. The goal is to preserve the min-cuts for all terminal partitions. As terminals may lie on $O(1)$ faces, for different terminal partitions, the dual partitioning cycle corresponding to the min-cuts may go around these $O(1)$ faces in different ways, and therefore it is not enough to just preserve pairwise shortest paths in the dual graph. Indeed, as observed by [KR17], we need to preserve, for each pair of terminals, and for each pattern (how a path going around the $O(1)$ faces, e.g., on the left side of face 1, on the right side of face 2, etc), the shortest path following the pattern and connecting the terminal pair. We call such a sparsifier a *pattern distance emulator*, which a generalization of the

4

standard distance emulators. As there are $O(1)$ faces, there are $2^{O(1)}$ different patterns (left/right for each face). We manage to construct a near-linear size pattern emulator, incurring an unharmful factor $2^{O(1)}$ loss in its size.

We believe that the notion of pattern emulators is of independent interest, and should play a role in solving other algorithmic problems on planar graphs.

Upper bounds for quasi-bipartite graphs. For exact (quality-1) contraction-based cut sparsifiers, a central notion in previous constructions and lower bound proofs is the *profiles* of vertices. Specifically, the profile of a vertex v specifies, for each terminal partition $(S, T \setminus S)$, on which side of the cut $\mathsf{mincut}_G(S, T \setminus S)$ lies the vertex v. Vertices of the same profile can be safely contracted together, and the number of different profiles is therefore an upper bound on the size of exact cut sparsifiers. For general graphs, a doubly-exponential upper bound of 2^{2^k} [HKNR98] on the number of distinct profiles was established, and this was later improved to $2^{\binom{k}{k/2}}$ [KR14]. For quasi-bipartite graphs, we will utilize its structure to proved a better (single-exponential) upper bound.

By the definition of quasi-bipartite graphs, there are no edge between non-terminal vertices, so non-terminal vertices form separate stars with terminals, and therefore contribute to terminal min-cuts independently. For example, in a star centered at v with 6 edges $e_1, \ldots, e_6$ connecting to terminal $t_1, \ldots, t_6$, respectively. The profile of vertex v only depends on the weights of the edges $e_1, \ldots, e_6$: for the terminal partition $(S = \{t_1, t_2, t_3\}, T \setminus S = \{t_4, t_5, t_6\})$,

v lies on the S side of $\mathsf{mincut}_G(S, T \setminus S)$ iff $w(e_1) + w(e_2) + w(e_3) > w(e_4) + w(e_5) + w(e_6)$.

This gives us some power to reveal properties of profiles in quasi-bipartite graphs. For example, if we consider subsets $S \subseteq T$ being $\{1, 2\}, \{3, 4\}, \{1, 3\}, \{2, 4\}$, then v cannot lie on the S side for all these terminal min-cuts. Since otherwise, letting $w = \sum_{1 \le i \le 6} w(e_i)$, we get $w(e_1) + w(e_2) > w/2$, $w(e_3) + w(e_4) > w/2$, $w(e_1) + w(e_3) \le w/2$, and $w(e_2) + w(e_4) \le w/2$, a contradiction. This means that there are some "configurations" in the family of terminal subsets that are simply not realizable by any vertex profiles in quasi-bipartite graphs. We implement this idea through the notion of VC-dimension, realizing "configurations" by "set systems" and "not realizable" by "non-shatterable", and obtain a bound of $k^{O(k^2)} = 2^{O(k^2 \log k)}$ on the number of profiles in quasi-bipartite graphs.

For quality-$(1 + \varepsilon)$ cut sparsifiers, since we allow a small multiplicative error in accuracy, we are not restricted to contracting vertices with the same profile, but are allowed to moderately manipulate vertex profiles so as to reduce their number. We implement this idea by "projections onto $O(1/\varepsilon^2)$-size stars". On the one hand, consider for example a full star centered at v (that is, a star containing edges (v, t) for all $t \in T$ with uniform edge weights, and a subset $S \subseteq T$ with $|S| = |T|/2 - 1$, so v lies on the $T \setminus S$ side of $\mathsf{mincut}_G(S, T \setminus S)$. If we uniformly at random sample $c = O(1/\varepsilon^2)$ edges from it and obtain a "mimicking substar" H_v, then by central limit theorem, with high probability H_v contains at most $c/2 + O(1/\varepsilon)$ edges to S and at least $c/2 - O(1/\varepsilon)$ edges to $T \setminus S$. Therefore, even if H_v fails in mimicking the behavior of the full star on the cut $(S, T \setminus S)$, in that it mistakenly selects more edges to S than to $T \setminus S$ and therefore place v on the S side of $\mathsf{mincut}_G(S, T \setminus S)$ rather than the $T \setminus S$ side, it only causes an error of $\frac{c/2 + O(1/\varepsilon)}{c/2 - O(1/\varepsilon)} = 1 + O(\varepsilon)$ in the min-cut size, which is allowed.

On the other hand, we have shown in the exact case that the number of profiles for stars on a fixed $O(1/\varepsilon^2)$-size terminal set is $O(1/\varepsilon^2)^{O(1/\varepsilon^4)}$ (replacing k with $O(1/\varepsilon^2)$ in the $k^{O(k^2)}$ bound). Since

the number of $O(1/\varepsilon^2)$-size terminal subsets is $k^{O(1/\varepsilon^2)}$, we can bound the total number of profiles produced by $O(1/\varepsilon^2)$-size starts by $k^{O(1/\varepsilon^2)} \cdot O(1/\varepsilon^2)^{O(1/\varepsilon^4)}$, obtaining a size bound of $k^{O(1/\varepsilon^2)} f(\varepsilon)$ for quality-$(1+\varepsilon)$ contraction-based cut sparsifiers for quasi-bipartite graphs.

Lower bounds for quasi-bipartite graphs. As shown in [CT24b, CT24a], contraction-based sparsifiers are closely related to the Steiner node version of the classic 0-Extension problem [Kar98]. Specifically, [CT24b] showed that the best quality achievable by contraction-based flow sparsifiers is bounded by the integrality gap of the semi-metric LP relaxation. For cut sparsifiers, we observe that the best achievable quality is controlled by an even more restricted case of the 0-Extension with Steiner Nodes problem, where the underlying graph is a boolean hypercube. We focus on this special case, construct a hard hypercube-instance which is also a quasi-bipartite graph, and prove a size lower bound for its $(1+\varepsilon)$-approximation, leading to a same size lower bound for quality-$(1+\varepsilon)$ contraction-based cut sparsifiers for quasi-bipartite graphs.

Concurrent Work. Independent of work, Das, Kumar, and Vaz, showed in their work [DKV24] that quasi-bipartite graphs admit exact non-contraction-based cut sparsifiers of size 2^{k^2} and exact contraction-based cut sparsifiers of size 2^{k^3}. Our Theorem 2 gives a $2^{O(k^2 \log k)}$ size bound for exact contraction-based cut sparsifiers, which is slightly stronger than their 2^{k^3} bound, and slightly weaker than their 2^{k^2} bound. They also have some results on flow sparsifiers. For example, quasi-bipartite graphs admit exact flow sparsifiers of size 3^{k^3}, and treewidth-w graphs admit quality-$O(\frac{\log w}{\log \log w})$ flow sparsifiers of size $O(kw)$.

1.3 Related Work

Edge sparsifiers. After Karger's result [Kar99] on cut-preserving edge sparsifiers, there are other work using sampling-based approaches. Benzcur and Karger [Ben96] sampled edges based on inverse edge-strengths, and obtained a sparsifier of $O(n \log n/\varepsilon^2)$ edges. Fung and Harvey [FH10] sampled edges according to their inverse edge-connectivity, and also obtained the bound of $O(n \log^2 n/\varepsilon^2)$. Spielman and Srivastava [SS11] sampled edges based on their inverse effective-resistance to obtain a spectral sparsifier (a generalization of cut sparsifiers) with size $O(n \log n/\varepsilon^2)$. This bound was later improved by Batson, Spielman, and Srivastava [BSS12] to $O(n/\varepsilon^2)$.

Other work on cut sparsifiers. There are some work on (i) constructing better cut sparsifiers for special families of graphs, for example trees [GR17] and planar graphs with all terminals lying on the same face [GHP17], and graphs with bounded treewidth [AGK14]; (ii) preserving terminal min-cut values up to some threshold value [CDK+21, Liu20]; and (iii) dynamic cut/flow sparsifiers and their utilization in dynamic graph algorithms [DGGP19, CGH+20, GRST21].

Flow sparsifiers. Flow sparsifiers are highly correlated with cut sparsifiers. Given a graph G and a set T of k terminals, a graph G' is a *flow sparsifier* of G with respect to T with *quality q*, iff every G-feasible multicommodity flow on T can be routed in G', and every G'-feasible multicommodity flow on T can be routed in G if the capacities of edges in G are increased by factor q. Flow sparsifiers are

stronger than cut sparsifiers, in the sense that quality-q flow sparsifiers are automatically quality-q cut sparsifiers, but the other direction is not true in general.

When no Steiner nodes are allowed, Leighton and Moitra [LM10] showed the existence of quality-$O(\log k / \log \log k)$ flow sparsifiers. On the lower bound side, the first quality lower bound in [LM10] is $\Omega(\log \log k)$, and this was later improved to $\Omega(\sqrt{\log k / \log \log k})$ by Makarychev and Makarychev [MM10].

In the setting where Steiner nodes are allowed, Chuzhoy [Chu12] showed $O(1)$-quality contraction-based flow sparsifiers with size $C^{O(\log \log C)}$ exist, where C is the total capacity of all edges incident to terminals (assuming that all edges have capacity at least 1). On the lower bound side, Krauthgamer and Mosenzon [KM23] showed that there exist 6-terminal graphs whose quality-1 flow sparsifiers must have an arbitrarily large size (i.e., the size bound cannot depend only on k and ε). Chen and Tan [CT24b] showed that there exists 6-terminal graphs whose quality-$(1+10^{-18})$ contraction-based flow sparsifiers must have an arbitrarily large size.

1.4 Organization

The rest of the paper is organized as follows. We start with some preliminaries and formal definitions in Section 2. We first provide the construction of planar cut sparsifier in Section 3, proving Theorem 1. We then show the construction of contraction-based cut sparsifiers for quasi-bipartite graphs in Section 4 and Section 5, proving Theorem 2 and Theorem 4. Finally, we show the lower bound for quasi-bipartite graphs, giving the proof of Theorem 3 in Section 6.

2 Preliminaries

By default, all logarithms are to the base of 2. All graphs are undirected and simple.

Let $G = (V, E, w)$ be an edge-weighted graph, where each edge $e \in E$ has weight w_e. In this paper, weights can be either representing capacities (in the context of cut values) or lengths (in the context of shortest-path distances). For a vertex $v \in V$, we denote by $\deg_G(v)$ the degree of v in G. For a pair v, v' of vertices in G, we use $\mathsf{dist}_G(v, v')$ (or $\mathsf{dist}_w(v, v')$) to denote the shortest-path distance between v and v' in G, viewing edge weights as lengths. We may omit the subscript G in the above notations when the graph is clear from the context.

Cut sparsifiers, contraction-based sparsifiers. Let G be a graph and let T be a subset of its vertices called *terminals*. Let H be a graph with $T \subseteq V(H)$. We say that H is a *cut sparsifier* of G with respect to T with *quality* $q \geq 1$, iff for any partition of T into two non-empty subsets T_1, T_2,

$$\mathsf{mincut}_H(T_1, T_2) \leq \mathsf{mincut}_G(T_1, T_2) \leq q \cdot \mathsf{mincut}_H(T_1, T_2).$$

We will use the following observation, which is immediate from the definition of cut sparsifiers.

Observation 5. *If (G_1, T) is a quality-q_1 cut sparsifier of (G, T), and (G_2, T) is a quality-q_2 cut sparsifier of (G_1, T), then (G_2, T) is a quality-$q_1 q_2$ cut sparsifier of (G, T).*

We say that a graph H is a *contraction-based sparsifier of G with respect to T*, iff there exists a partition $\mathcal{L}$ of vertices in G into subsets where different terminals in T lie in different subsets in $\mathcal{L}$, and H is obtained from G by contracting, for each $L \in \mathcal{L}$, vertices in L into a supernode u_L, keeping parallel edges and discarding self-loops.

3   Proof of Theorem 1

In this section we provide the proof of Theorem 1. We will use a similar algorithmic framework as [CKT22], by first considering the case where all terminals lie on the boundary of one face and then generalizing the arguments to the $O(1)$-face case and eventually to the general case.

Let G, G' be planar graphs with $T \subseteq V(G), V(G')$. Let $\mathcal{F}$ be the set of faces in the planar drawing of G that contain all terminals, and we define $\mathcal{F}'$ similarly for G'. We say instances $(G, T), (G', T)$ are *aligned*, iff there is a one-to-one correspondence between sets $\mathcal{F}$ and $\mathcal{F}'$, such that for each face $F \in \mathcal{F}$, the set of terminals in G lying on F is exactly the set of terminals in G' lying on F', its corresponding face in $\mathcal{F}'$, and the circular ordering in which these terminals appear on F and F' are also identical. We say (G, T) is a $|\mathcal{F}|$-face instance. Throughout this section, all the cut sparsifiers that we will ever construct for the input instance or any subinstance generated from it are aligned.

3.1   Preparation: removing separator terminals, and dual graphs

We start with the following divide and conquer lemma, whose proof is deferred to Appendix A.1.

Lemma 6. *Let (G, T) be a planar instance and let $\mathcal{G}$ be a collection of subgraphs of G such that the edge sets $\{E(G') \mid G' \in \mathcal{G}\}$ partition $E(G)$. For each graph $G' \in \mathcal{G}$, we denote by $T_{G'}$ the set that contains all vertices in $T \cap V(G')$ and all vertices of G' that appear in some other graph in $\mathcal{G}$. If for each $(G', T_{G'})$ we are given a quality-q aligned cut sparsifier $(H_{G'}, T_{G'})$, then we can efficiently obtain a quality-q planar cut sparsifier H' of G with $|V(H')| \leq \sum_{G' \in \mathcal{G}} |V(H_{G'})|$.*

We say a vertex $v \in V(G)$ is a *separator* iff $G \setminus v$ is disconnected. Denote its connected components by $G'_1, \ldots, G'_r$. For each $1 \leq i \leq r$, let G_i be the subgraph of G induced by $V[G'_i] \cup \{v\}$. We say graphs $G_1, \ldots, G_r$ are obtained by *splitting G at v*. We use the following lemma from [CKT22].

Claim 7 (Claim 4.3 in [CKT22]). *Let (G, T) be an instance. Let U be the set of separators in T. Let $\mathcal{G}$ be the set of graphs obtained by splitting G at all vertices in U (that is, we first split G at a vertex in U, and then repeatedly split each of the obtained graph at one of its separator terminal, until no resulting graph contains any separator terminal). Then $\sum_{G' \in \mathcal{G}} |T \cap V(G')| \leq O(|T|)$.*

With Lemma 6, we will assume that no terminal in T is a separator in G, and so if we go around the boundary of a face in $\mathcal{F}$, we will visit each terminal that lies on this face exactly once. This is since we can split G at all its separator terminals, compute quality-$(1 + \varepsilon)$ cut sparsifiers for them separately, and then combine them in the way of Lemma 6, paying only an $O(1)$ factor in the total size, as guaranteed by Claim 7.

Dual graphs. Cuts in planar graphs are cycles in their dual graphs. We will exploit this property in a similar way as [KR17] and transform the cut-preserving tasks to distance-preserving tasks. For technical reasons, in order to handle graphs with terminals, our definition of dual graphs are slightly different from the standard planar dual graphs.

Let (G, T) be the input instance and let $\mathcal{F}$ be the faces that contains all terminals. Assume each face is a simple cycle. For each $F \in \mathcal{F}$, add an auxiliary vertex x_F, and connect it to every terminal via a new edge, such that these edges are drawn in an internally disjoint way within face F, so that they separates face F into subfaces, which we call special subfaces. Then we take the standard planar dual of the modified graph. Here each special face correspond to a node, which we call *dual terminals*. We then remove all edges in the dual graph connecting a pair of dual terminals. We denote the resulting graph by G^*, and denote by T^* the set of dual terminals. See Figure 1 for an illustration. Note that when graph G does not have the property that each face is a simple cycle, as long as we are guaranteed that no terminal is a separator, we can still define dual graphs similarly. It is also easy to verify that the dual of the dual graph G^* is the original graph G itself. Sometimes we say that we *reverse* G^* to obtain G. Finally, as both G and G^* are planar graphs, and each non-terminal vertex in G^* corresponds to a face in G, the number of vertices in G and G^* are within $O(1)$ factor from each other.

Observation 8. (G, T) *is a f-face instance iff* (G^*, T^*) *is a f-face instance.*

We denote by $\mathcal{F}^*$ the faces in G^* that contain all dual terminals. It is easy to see that there exists a natural one-to-one correspondence between $\mathcal{F}$ and $\mathcal{F}^*$.
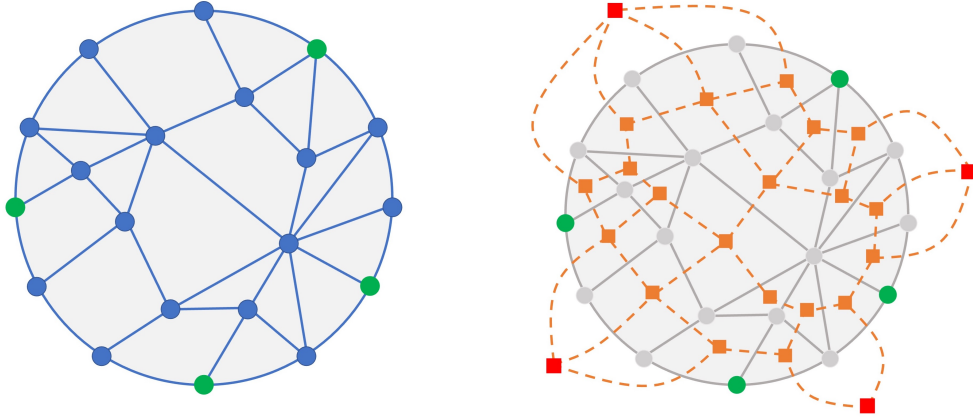
Figure 1: An illustration of a one-face instance (G, T) (left) and its dual (G^*, T^*) defined in our way (right). The dual terminals in T^* are marked in red. Taking the dual of (or reverse) G^* we get G back.

3.2 Warm-up: the one-face case

In this subsection we prove Theorem 1 for the special case where all terminals lie on the boundary of the outer face in the planar embedding of G.

Lemma 9. *Let (G, T) be a one-face instance. Then for any partition (T_1, T_2) of the terminals into non-empty subsets, for any min-cut $\hat{E}$ in G separating T_1 from T_2, the corresponding edge set $\hat{E}^*$ in the dual graph G^* is the union of several edge-disjoint shortest paths connecting dual terminals.*

Proof. TOPROVE 0 □

We are now ready to prove Theorem 1 for one-face instances using Theorem 1. We first construct the dual graph G^*, and use the result in [CKT22] to construct a quality-$(1 + \varepsilon)$ planar distance emulator $\hat{G}^*$ with size $O(k \cdot \mathsf{poly}(\log k/\varepsilon))$, and then reverse it to get $\hat{G}$. From our definition of dual graphs and their reverse, $\hat{G}$ contains all terminals and $|V(\hat{G})| = O(k \cdot \mathsf{poly}(\log k/\varepsilon))$. It remains to show that $(\hat{G}, T)$ is a quality-$(1 + \varepsilon)$ cut sparsifier for (G, T).

Let (T_1, T_2) be a partition of T. On the one hand, the min-cut in G separating T_1 from T_2 consists of several edge-disjoint dual terminal shortest paths in G^*, and as $\hat{G}^*$ is a $(1 + \varepsilon)$ aligned distance emulator for G^*, we can find, for each such dual terminal shortest path in G^*, a shortest path in $\hat{G}^*$ connecting its corresponding dual terminals, with at most $(1 + \varepsilon)$ times the length. By taking the union of the edges in all these paths in $\hat{G}^*$ (note however that these paths are not necessarily edge-disjoint in $\hat{G}^*$), we obtain an edge set in $\hat{G}$ separating T_1 from T_2 whose value is at most $(1+\varepsilon)$ times the value of $\mathsf{mincut}_G(T_1, T_2)$. On the other hand, we apply the same analysis, starting with a min-cut in $\hat{G}$ separating T_1 from T_2. As G^* is also a $(1 + \varepsilon)$ aligned distance emulator for $\hat{G}^*$, we can derive similarly that $\mathsf{mincut}_{\hat{G}}(T_1, T_2) \leq (1 + \varepsilon) \cdot \mathsf{mincut}_G(T_1, T_2)$.

3.3   The $O(1)$-face case

In this subsection we prove the following lemma.

Lemma 10. *There exists a universal constant $c > 0$, such that for any real number $0 < \varepsilon < 1$, every f-face instance (H, U) has an aligned f-face instance (H', U) as its quality-$(1 + \varepsilon)$ cut sparsifier, with $|V(H')| \leq |U| \cdot (cf \log |U|/\varepsilon)^{cf^2}$.*

Terminal-separating min-cuts in G dual-terminal shortest paths in our graph G^*. Therefore, from now on we focus on preserving dual-terminal shortest paths in G^*. Recall that $\mathcal{F}^*$ is the collection of faces in G^* that contains all dual terminals. The following notion is central in our algorithm.

Pattern-shortest paths, pattern distances, and pattern emulators.   For a pair t, t' of dual terminals in graph G^*, there are more than one way for them to be connected by a path, depending on how the path goes around other faces in $\mathcal{F}^*$, called *patterns*, A rigorous way of defining patterns was given in [KR17], which we describe below.

Fix a point ν^* in the plane. Let γ be a closed curve (not necessarily simple). A vertex $v \notin \gamma$ is said to be *inside* γ iff any simple curve connecting ν^* to v crosses γ an odd number of times, otherwise it is said to be *outside* γ. For a pair t, t' of dual terminals in G^*, we denote by $\Pi_{t,t'}$ the shortest path connecting them in G^*. For every face $F \in \mathcal{F}^*$, we place a point z_F inside its interior. Consider now a path P in G^* connecting t, t'. Note that the union of P and $\Pi_{t,t'}$ is a closed curve. Now the *pattern* of P is defined to be a $|\mathcal{F}^*|$-dimensional vector $\Phi(P) = (\phi_F)_{F \in \mathcal{F}^*}$, where $\phi_F = 1$ iff z_F is inside the closed curve formed by the union of P and $\Pi_{t,t'}$ and $\phi_F = -1$ iff z_F is outside.

Observation 11. *Between every pair of terminals there are 2^f different patterns.*

Proof. TOPROVE 1 $\hspace{1cm}$ □

Let P, P' be simple paths with same endpoints, the above definition of pattern implies that paths P, P' *have the same pattern* iff all points $\{z_F\}_{F \in \mathcal{F}^*}$ lie outside the closed curve $P \cup P'$. We will also consider paths P, P' with one common endpoint t. Let R be a path such that the other-endpoints of P and P' both lie on R. In this case we say that paths P, P' have the *same pattern with respect to R*, iff all points $\{z_F\}_{F \in \mathcal{F}^*}$ lie ourside the closed curve formed by the union of P, P', and the subpath of R between the other-endpoint of P and the other-endpoint of P'.

For a pattern Φ, the t-t' Φ-*shortest path* is defined to be the shortest path among all t-t' paths with pattern Φ. Its length is defined to be the Φ-*distance* between t, t', which we denote by $\mathsf{dist}^\Phi(t_1, t_2)$. Let $(G, T), (H, T)$ be aligned instances. We say that (H, T) is a *quality-q pattern emulator* of (G, T), iff for every pair t_1, t_2 of terminals and every pattern Φ, the Φ-distance between t_1, t_2 in G is within factor q from the Φ-distance between t_1, t_2 in H. We use the following result from [KR17].

Theorem 12 (Adaptation of Theorem 4.6 in [KR17])**.** *Let $(G, T), (H, T)$ be aligned instances. If the dual instance (H^*, T^*) is a quality-q pattern emulator of the dual instance (G^*, T^*), then (H, T) is a quality-q cut sparsifier of (G, T).*

3.4 Constructing pattern emulators

We will now prove the following lemma, which, combined with Theorem 12, implies Lemma 10.

Lemma 13. *There exists a universal constant $c > 0$, such that for any $0 < \varepsilon < 1$, every f-face instance (G, T) admits a quality-$(1+\varepsilon)$ pattern emulator (H, T) with $|V(H)| \leq |T| \cdot (cf \log |T|/\varepsilon)^{cf^2}$.*

We prove Lemma 13 by induction on f. The base case where $f = 1$ has been proved in Section 3.2. Assume that we are given a f-face instance (G, T). We will convert it into an $(f-1)$-face instance and apply the inductive hypothesis. Throughout this subsection, we use the parameters $\varepsilon' = \varepsilon/2f^2$ and $\varepsilon'' = (1 - 1/f^2) \cdot \varepsilon$.

We first pick a pair t, t' of terminals that lie on distinct faces, denoted by α, α', respectively, and compute the shortest path $\Pi_{t,t'}$ in G connecting t to t', such that $\Pi_{t,t'}$ does not contain any terminal as inner vertices. Denote $\Pi = \Pi_{t,t'}$. Then we compute a set of vertices on Π that we call *portals*.

Portals on Π. We use the notion of ε-covers defined in [KS98, Tho04]. Let R be a shortest path and let v be a vertex that does not lie in R. An ε-*cover* of v in R is a set $C(v, R)$ of vertices, such that for any vertex $x \in R$, there exists some $y \in C(v, R)$, such that $\mathsf{dist}(v, x) \leq \mathsf{dist}(v, y) + \mathsf{dist}(y, x) \leq (1+\varepsilon) \cdot \mathsf{dist}(v, x)$. It has been shown [KS98, Tho04] that for every $\varepsilon > 0$, shortest path R and vertex $v \notin R$, there exists an ε-cover of v in R of size $O(1/\varepsilon)$. As we want to construct pattern emulators, we will need to modify the definition of ε-covers into a "pattern version" accordingly, as follows.

Let G be an f-face instance, R be a shortest path in G that does not intersect any face internally, and v be a vertex that does not lie in R. Let Q be a path connecting v to some vertex in R, and let Φ be its pattern. A Φ-*respecting ε-cover* of v in R is a set $C(v, R, \Phi)$ of vertices, such that for

11

any vertex $x \in R$, there exists some $y \in C(v, R, \Phi)$, such that the shortest v-y path with the same pattern as Φ, concatenated with the y-x subpath of R, is within factor $(1 + \varepsilon)$ in length with the shortest v-x path with the same pattern as Φ. In other words (slightly abusing the notations),

$$\mathsf{dist}^{\Phi}(v, x) \leq \mathsf{dist}^{\Phi}(v, y) + \mathsf{dist}(y, x) \leq (1 + \varepsilon) \cdot \mathsf{dist}^{\Phi}(v, x).$$

We use the following lemma, whose proof is similar to the previous result of ε-cover and is deferred to Appendix A.2.

Lemma 14. *For any planar instance G, shortest path R in G, vertex $v \notin R$, and pattern Φ, there exists a Φ-respecting ε-cover of v in R of size $O(1/\varepsilon)$.*

We now define a set Y of vertices on Π as follows. For each terminal t and for each pattern Φ, let $Y_{t,\Phi}$ be the pattern-respecting ε'-cover of v in R of size $O(1/\varepsilon')$ given by Lemma 14. We then let set $Y = \bigcup_{t,\Phi} Y_{t,\Phi}$. From Observation 11 and Lemma 14, $|Y| \leq O(2^f \cdot |T|/\varepsilon')$.

We then slice the graph G open along the path Π connecting faces α, α'. Specifically, we duplicate path Π into two copies Π_1, Π_2 and place them closely at two sides (which we call 1-side and 2-side, respectively) of its original image, such that the thin strip between them connect faces α, α' into a single face β. All vertices x on P into their copies x_1, x_2 (x_1 lies on Π_1 and x_2 lies on Π_2), including the terminals t, t' that now have copies t_1, t_2 and t'_1, t'_2. All weights on edges remain the same. It is easy to verify that the obtained graph, which we denote by G', is a $(f - 1)$-face instance. Let T' be the set that contains all copies of terminals and portals in Y. We then construct a quality-$(1 + \varepsilon'')$ pattern emulator H' of G' with respect to T', and then glue, for each portal $y \in Y$, its copies y_1, y_2 back to their original vertex y. The resulting graph is denoted by H and is returned as the pattern emulator for G. For a complete description of the cut and glue operations, please refer to Appendix B of [CKT22]. See Figure 2(b) for an illustration.

It remains to complete the proof by induction. We first bound the size of graph H.

Size of H. Since graph H is obtained from graph H' by gluing the portals on paths Π_1, Π_2 back to their original vertices, $|V(H)| \leq |V(H')|$, so it suffices to bound the number of vertices in H'. Recall that T' is the set that contains all portals and terminals in G', then

$$
\begin{aligned}
|V(H')| &\leq |T'| \cdot \left(\frac{cf \cdot \log |T'|}{\varepsilon''} \right)^{c(f-1)^2} \leq \frac{c \cdot 2^f \cdot |T|}{\varepsilon/2f^2} \cdot \left(\frac{cf \cdot \log(c \cdot 2^f \cdot |T|/\varepsilon')}{\varepsilon \cdot (1 - 1/f^2)} \right)^{c(f-1)^2} \\
&\leq |T| \cdot \left(\frac{cf}{\varepsilon} \right)^{c(f-1)^2 + f} \cdot \left(\frac{\log(cf^2 \cdot 2^f \cdot |T|/\varepsilon)}{(1 - 1/f^2)} \right)^{c(f-1)^2} \\
&\leq |T| \cdot \left(\frac{cf}{\varepsilon} \right)^{c(f-1)^2 + f} \cdot \left(\log |T| + \log(cf^2 \cdot 2^f/\varepsilon) \right)^{c(f-1)^2} \\
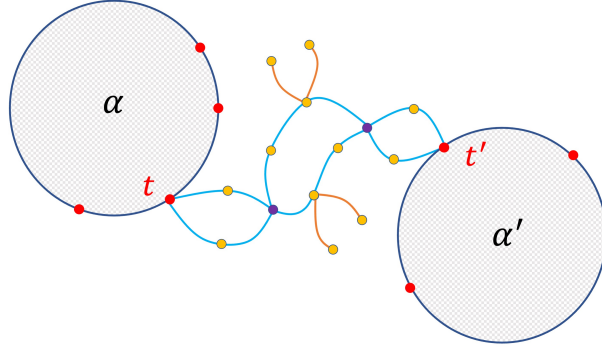&\leq |T| \cdot \left(\frac{cf^2 \log |T|}{\varepsilon} \right)^{cf^2},
\end{aligned}
$$

where we have used the previously set parameters $\varepsilon' = \varepsilon/2f^2$ and $\varepsilon'' = (1 - 1/f^2) \cdot \varepsilon$, and the fact that $(1 - \frac{1}{f^2})^{-c(f-1)^2} \leq e^c < c^c$, as c is large enough.
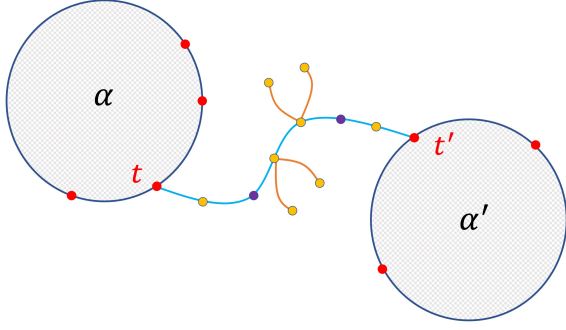
(a) Graph G: faces α, α' (shaded gray), path Π (blue), and portals (purple).
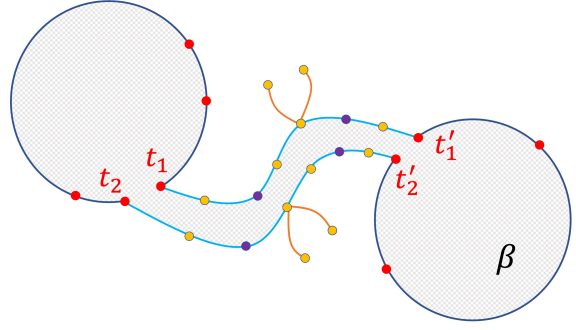
(b) Graph G': the new hole β (shaded gray), and paths Π_1, Π_2 (blue).

(c) Graph H: holes α, α' and portals are restored.

Figure 2: An illustration of cutting open and gluing an f-face instance along path Π.

Quality of H. Recall that we sliced G open along Π to obtain G', computed a $(1 + \varepsilon'')$ pattern emulator H' for G', and then glue H' back to obtain H. For the purpose of analysis, image that we have also glued G' back to obtain G''. The analysis is complete by the following claims.

Claim 15. (H, T) is a quality-$(1 + \varepsilon'')$ pattern emulator for (G'', T).

Proof. TOPROVE 2 $\qquad\square$

Claim 16. (G'', T) is a quality-$(1 + \varepsilon')$ pattern emulator for (G, T).

Proof. TOPROVE 3 $\qquad\square$

From Claim 16 and Claim 15, (H, T) is a $(1 + \varepsilon'') \cdot (1 + \varepsilon') \leq (1 + \varepsilon)$ pattern emulator for (G, T). This completes the proof by induction of Lemma 13.

3.5 Completing the proof of Theorem 1

In this section, we complete the proof of Theorem 1 using the results in previous subsections.

Structured r-divisions. Let G be a planar graph on n vertices and let T be its terminals. For any integer $r > 1$, a *structured r-division* of G is a collection $\mathcal{G}$ of subgraphs of G, such that

- $|\mathcal{G}| = O(n/r)$;

- the edge sets $\{E(G') \mid G' \in \mathcal{G}\}$ partitions $E(G)$;

- for each $G' \in \mathcal{G}$, $|V(G')| \leq r$, $|T \cap V(G')| = O(1 + |T| \cdot r/n)$, and if we call vertices in G' that appear in some other graph in $\mathcal{G}$ its *boundary vertices*, then
    - the number of boundary vertices is $O(\sqrt{r})$;
    - in the planar drawing of G' induced by the planar drawing of G, all boundary vertices lie on $O(1)$ faces.

We use the following lemma in [CKT22] for computing structured r-divisions.

Lemma 17 (Lemma 5.5 in [CKT22]). *Given any planar instance (G, T) and any integer $r \geq 0$, we can efficiently compute a structured r-division.*

We now prove Theorem 1. The algorithm is simply: compute a structured r-division $\mathcal{G}$ of the input graph G, construct cut sparsifiers for graphs $G' \in \mathcal{G}$ using Lemma 10, and then glue them together to obtain a cut sparsifier for G. It turns out that we need to apply the algorithm several times in order to obtain a near-linear size cut sparsifier, each time with different parameters r, ε.

Lemma 18. *Given any planar instance (H, U) and any $0 < \varepsilon < 1$, we can efficiently compute a quality-$(1 + \varepsilon)$ planar cut sparsifier $(\hat{H}, U)$ with size $|V(\hat{H})| \leq O\big(\sqrt{|V(H)||U|} \cdot (\log |V(H)|/\varepsilon)^{c'}\big)$, for some universal constant $c' > 0$.*

Proof. TOPROVE 4 □

We now complete the proof of Theorem 1 using Lemma 18. We start by applying the algorithm in [KR17] to obtain an exact cut sparsifier (G_0, T) of (G, T) with $|V(G)| = 2^{2k} \cdot \mathsf{poly}(k) \leq 2^{3k}$. Then the algorithm consists of two phases.

In the first phase, we set parameters $L = \log k$ and $\varepsilon' = \varepsilon/6L$, and sequentially for each $i = 1, \ldots, L$, we apply algorithm Lemma 18 to graph G_{i-1} to obtain graph G_i. We show that $|V(G_L)| = (k/\varepsilon)^{O(1)}$ (applying the next lemma with $i = L = \log k$).

Claim 19. *For each $1 \leq i \leq L$, $|V(G_i)| \leq 2^{3k/2^i} \cdot (18k \log k/\varepsilon)^{4c'}$.*

Proof. TOPROVE 5 □

In the second phase, we set parameters $\bar{L} = 2 \log k \log k$ and $\bar{\varepsilon}' = \varepsilon/6L$, and sequentially for each $i = 1, \ldots, \bar{L}$, we apply algorithm Lemma 18 to graph G_{L+i-1} to obtain graph G_{L+i}.

Claim 20. *For each $1 \leq i \leq \bar{L}$, $|V(G_{L+i})| \leq k^{1+5c'/2^i} \cdot (5c' \log k/\bar{\varepsilon}')^{2c'}$.*

Proof. TOPROVE 6 □

14

Applying $i = \bar{L}$, we get that $|V(G_{L+\bar{L}})| \leq O(k \cdot \mathsf{poly}(\log k/\varepsilon))$ (as c' is a universal constant).

On the other hand, from Observation 5, we get that instance $(G_{L+\bar{L}}, T)$ is a cut sparsifier of instance (G, T) with quality $(1 + \varepsilon/6L)^L \cdot (1 + \varepsilon/6\bar{L})^{\bar{L}} \leq (1 + \varepsilon)$. The proof is complete.

4   Proof of Theorem 2

In this section, we provide the proof of Theorem 2.

We first prove the upper bound that every quasi-bipartite graph with k terminals admits an exact contraction-based cut sparsifier on $k^{O(k^2)}$ vertices. Let G be a quasi-bipartite graph with a set T of k terminals. Throughout, we will assume that for every $\emptyset \subsetneq S \subsetneq T$, there is a unique min-cut in G separating S from $T \setminus S$. This can be achieved by slightly perturbing the edge weights $\{w(e)\}_{e \in E(G)}$.

We use the definition of *profiles* studied in previous works [HKNR98, KR14]. For a vertex $v \in V(G)$, its profile π^v is defined to be a $(2^{|T|} - 2)$-dimensional vector whose coordinates are indexed by proper subsets S of T. Specifically, for every $\emptyset \subsetneq S \subsetneq T$, $\pi_S^v = 1$ iff v lies on the side of S in the $(S, T \setminus S)$ min-cut in G, otherwise $\pi_S^v = 0$. Let $\Pi(G)$ be the collection of all distinct profiles of the vertices in G. The following result was proved in [HKNR98].

Lemma 21. *Every graph G admits a quality-1 contraction-based cut sparsifier of size $|\Pi(G)|$.*

Therefore, to prove Theorem 2, it is sufficient to prove the following lemma.

Lemma 22. *For any quasi-bipartite graph G with k terminals, $|\Pi(G)| = k^{O(k^2)}$.*

To prove Lemma 22, we will show that $\Pi(G)$, when viewed as a family of sets (instead of vectors), has bounded VC-dimension. Specifically, consider the ground set $\mathcal{U} = \{S \mid \emptyset \subsetneq S \subsetneq T\}$. Now each profile π^v naturally defines a subset of the ground set $\mathcal{U}$: $\Pi^v = \{S \mid \pi_S^v = 1\}$. Abusing the notation, we also write $\Pi(G) = \{\Pi^v \mid v \in V(G)\}$. We will show that the set family $\Pi(G)$ on the ground set $\mathcal{U}$ has VC-dimension $O(k \log k)$. Then from Sauer-Shelah lemma [She72, Sau72],

$$|\Pi(G)| \leq |\mathcal{U}|^{O(k \log k)} \leq (2^k)^{O(k \log k)} = k^{O(k^2)}.$$

Shattering sets and VC dimension.   Let $\mathcal{F}$ be a set family on the ground set $\mathcal{U}$. Let U be a subset of $\mathcal{U}$. We say that the family $\mathcal{F}$ *shatters* U, iff for every subset $U' \subseteq U$, there exists some set $F \in \mathcal{F}$, such that $F \cap U = U'$. The VC-dimension of the family $\mathcal{F}$ is the maximum size of a set U shattered by $\mathcal{F}$.

The remainder of this section is dedicated to the proof of the following claim.

Claim 23. The VC-dimension of $\Pi(G)$ on the ground set $\mathcal{U}$ is $O(k \log k)$.

Proof. TOPROVE 7   □

We now prove the lower bound that for every integer k, there exists a quasi-bipartite graph with k terminals whose exact contraction-based cut sparsifier must contain $\Omega(2^k)$ vertices.

We construct a graph G as follows. The terminal set contains k vertices and is denoted by T. For every subset $\emptyset \subsetneq S \subsetneq T$ with even size, we add a new vertex v_S and connect it to every vertex of S. Every edge is given a capacity that is chosen uniformly at random from the interval $(1-2^k, 1+2^{-k})$. In this way we can guarantee that with high probability, sets E', E'' of edges of G have the same total weight iff $E' = E''$. And clearly, G is a quasi-bipartite graph with $|V(G)| = \Omega(2^k)$.

The proof of the $\Omega(2^k)$ size lower bound is completed by the following claims.

Claim 24. Every pair of distinct vertices have different profiles.

Proof. TOPROVE 8 □

Claim 25. No pair of vertices can be contracted in an exact contraction-based cut sparsifier of G.

Proof. TOPROVE 9 □

5 Proof of Theorem 4

In this section, we provide the proof of Theorem 4, showing that every quasi-bipartite graph with k terminals admits a quality-$(1 + \varepsilon)$ contraction-based cut sparsifier of size $k^{O(1/\varepsilon^2)} \cdot (1/\varepsilon)^{O(1/\varepsilon^4)}$.

5.1 Sparsifying stars by sampling

Throughout, we use the parameter $c = \lceil 100/\varepsilon^2 \rceil$.

For every non-terminal v in G, denote by G_v the star induced by all incident edges of v. We will construct another star H_v that contains at most c edges to mimic the behaviors of G_v, as follows.

Denote by w the total weight of all edges in G_v. We say an edge e is *heavy* iff $w(e) \geq w/c$, otherwise we say it is *light*. Denote by h the number of heavy edges in G_v and by w^h the total weight of them, so $h \leq c$. We define a distribution $\mathcal{D}$ on light edges as follows: the probability for a light edge e is $\Pr_{\mathcal{D}}(e) = \frac{w(e)}{w-w^h}$. If $|V(G_v)| \leq c$, then we let $H_v = G_v$. Otherwise, we sample $c - h$ light edges from $\mathcal{D}$ with replacement. For each sampled edge, we assign to it a new weight of $\frac{w-w^h}{c-h}$. The graph H_v simply contains all heavy edges in G_v and all sampled light edges (with their new weight).

For a subset $S \subseteq T$, let $w_S(v)$ be the total weight of edges in G_v connecting v to a vertex in S, so $w = w_T(v)$, and similarly let $w'_S(v)$ be the total weight of edges in H_v connecting v to a vertex in S. In this subsection we will omit v and only write w_S, w'_S for notational simplicity. The following observation is immediate.

Observation 26. *Graph H_v contains at most c edges with a total weight of w. Moreover, for every subset $S \subseteq T$, $\mathbb{E}[w'_S] = w_S$.*

We now measure the similarity of graphs H_v and G_v in terms of terminal cuts. For a subset $S \subseteq T$, if $w'_S > w/2$, then in graph H_v, v lies on the S side of the $(S, T \setminus S)$ min-cut, otherwise v lies on the $T \setminus S$ side. We prove the following claim.

Claim 27. For a subset $S \subseteq T$ with $w_S < w/2$, $(w - 2 \cdot w_S) \cdot \Pr[w'_S > w/2] \leq \varepsilon \cdot w_S$.

Proof. TOPROVE 10 □

The following statement is a byproduct of the Case 3 above and will be useful later.

Claim 28. If $w_S \leq 1/(2e)$, $\Pr[w'_S > 1/2] \leq e^{-\frac{w}{2w_S}}$.

5.2 Construction of the cut sparsifier

Step 1. Computing special cuts and keeping important vertices

For every pair t, t' of terminals in T, we compute a min-cut in G separating t and t'. This min-cut induces a terminal partition $(S_{t,t'}, T \setminus S_{t,t'})$ where $t \in S_{t,t'}$ and $t' \in T \setminus S_{t,t'}$. We call such partitions *special cuts*.

Claim 29. For every vertex v and every subset $S \subseteq T$, there is a special terminal cut $(\tilde{S}, T \setminus \tilde{S})$, such that
$$\frac{\min\{w_{\tilde{S}}(v), w(v) - w_{\tilde{S}}(v)\}}{\mathsf{mincut}_G(\tilde{S})} \geq \frac{1}{k} \cdot \frac{\min\{w_S(v), w(v) - w_S(v)\}}{\mathsf{mincut}_G(S)}.$$

Proof. TOPROVE 11 □

Let $\eta = \frac{\varepsilon^4}{1000k}$. We call a vertex v *important* if there is a special terminal cut $(S, T \setminus S)$ such that the contribution of v on it (that is, the value of $\mathsf{mincut}_{G_v}(S, T \setminus S)$) is at least $(\eta/k) \cdot \mathsf{mincut}_G(S)$. There are at most $k \cdot k/\eta = O(k^3/\varepsilon^4)$ important vertices. We keep all important vertices and their incident edges in the cut sparsifier. That is, important vertices are not contracted with any other vertices.

According to Claim 29, every non-important vertex v contributes at most $\eta \cdot \mathsf{mincut}_G(S)$ to any cut $\mathsf{mincut}(S, T \setminus S)$. We handle them next.

Step 2. Contracting non-important vertices based on their sparsified stars

Recall that for each non-important non-terminal v, we have constructed in Section 5.1 a sparsified star H_v mimicking the original star G_v in G. Let π_v be the profile of vertex v in graph H_v. For each profile π, we contract all non-important vertices v with $\pi_v = \pi$ into a single node. This completes the construction of the contraction-based sparsifier. Denote the resulting graph by G'.

Clearly, G' is obtained from G by contractions only, so it is indeed a contraction-based sparsifier. The next observation shows that it contains at most $k^{O(1/\varepsilon^2)} \cdot (1/\varepsilon)^{O(1/\varepsilon^4)}$ vertices.

Observation 30. $|V(G')| \leq k^{O(1/\varepsilon^2)} \cdot (1/\varepsilon)^{O(1/\varepsilon^4)}$.

Proof. TOPROVE 12 □

In the rest of this section, we prove that G' is indeed a quality-$(1 + O(\varepsilon))$ cut sparsifier.

Fix a subset $S \subseteq T$ and we analyze the difference between $\mathsf{mincut}_G(S)$ and $\mathsf{mincut}_{G'}(S)$. For a non-terminal v, let x_v and x'_v be the contribution of the vertex v for the cut S in G and G' respectively, so $\mathsf{mincut}_G(S) = \sum_v x_v$.

Observation 31. *If v is an important vertex, then $x'_v = x_v$. If v is a non-important vertex, then x'_v is either x_v or $w(v) - x_v$. Moreover, $\mathbb{E}\left[x'_v\right] \leq (1 + \varepsilon)x_v$ and $x_v \leq \eta \cdot \mathsf{mincut}_G(S)$.*

Proof. TOPROVE 13 ◻

We classify all non-important vertices into two types. Let V_1 be the set that contains all vertices v such that $x_v/w(v) \geq \varepsilon^2/k^2$, and let V_2 contain the rest of non-important vertices. The proof is completed by the following claims.

Claim 32. $\Pr\left[\sum_{v \in V_1}(x'_v - \mathbb{E}\left[x'_v\right]) \leq \varepsilon \cdot \mathsf{mincut}_G(S)\right] \leq 2^{-2k}$.

Proof. TOPROVE 14 ◻

Claim 33. $\Pr\left[\sum_{v \in V_2} x'_v > \sum_{v \in V_2} x_v + 2\varepsilon \cdot \mathsf{mincut}_G(S)\right] < e^{-2k}$.

Proof. TOPROVE 15 ◻

Therefore, $\Pr\left[\sum_v x'_v > (1 + 3\varepsilon) \cdot \mathsf{mincut}_G(S)\right] < 2^{-2k+1}$. Taking the union bound over all subsets $S \subseteq T$, with probability at least $1 - 2^{k-1}$, G' is indeed a quality-$(1 + 3\varepsilon)$ cut sparsifier.

6   Proof of Theorem 3

In this section we prove Theorem 3, showing that for every $k \geq 1$ and $\varepsilon > 0$, there exists a quasi-bipartite graph with k terminals, whose quality-$(1+\varepsilon)$ contraction-based cut sparsifier must contain $k^{\Omega\left(\frac{1}{\varepsilon \log(1/\varepsilon)}\right)}$ vertices. We will make use of the following intermediate problem, which is a variant of the 0-Extension problem.

Boolean Hypercube Contraction (BHC).   In an instance of BHC, the input consists of

- a graph $G = (V, E)$ with $V = \{0, 1\}^d$, namely each vertex is a d-dimensional 0/1-string;

- a set $T \subseteq V$ of k terminals, such that every edge is incident to some terminal.

A solution is a mapping $f : V \to V$, such that for each terminal $t \in T$, $f(t) = t$. For solution f, its

- *cost* is defined as $\mathsf{cost}(f) = \sum_{(u,v) \in E} ||f(u) - f(v)||_1$; and

- *size* is defined as $|\{f(v) \mid v \in V\}|$, namely the size of its image set.

The goal is to compute a solution f with small size and cost. Specifically, the following ratio called *stretch*, is to be minimized:
$$\rho = \frac{\sum_{(u,v) \in E} ||f(u) - f(v)||_1}{\sum_{(u,v) \in E} ||u - v||_1}.$$

In fact, the BHC problem is the special case of the 0-Extension problem first introduced by Karzanov [Kar98], as a generalization of the multi-way cut problem and a special case of the metric labeling problem. The 0-Extension problem and some of its variants were shown to be concretely connected to cut/flow sparsifiers [Moi09, LM10, AGK14, CT24a, CT24b]. Specifically, BHC focuses on searching for "canonical solutions" of the 0-*Extension with Steiner Nodes* problem studied [CT24a], and moreover, instead of allowing the input graph to be any general graph, BHC only consider instances on the boolean hypercube.

The proof of Theorem 3 consists of the following two lemmas.

Lemma 34. *If for every $k \geq 1$ and $q \geq 1$, every k-terminal quasi-bipartite graph admits a quality-q contraction-based cut sparsifier of size $s(k, q)$, then every k-terminal quasi-bipartite instance of* BHC *admits a solution of stretch q and size $s(k, q)$.*

Proof. TOPROVE 16 □

Lemma 35. *For every $k \geq 1$ and $\varepsilon > 0$, there exists a k-terminal quasi-bipartite instance of* BHC, *such that any solution of stretch $(1 + \varepsilon)$ and must have size at least $k^{\Omega(1/(\varepsilon \log(1/\varepsilon)))}$.*

Proof. TOPROVE 17 □

A Missing Proofs

A.1 Proof of Lemma 6

Graph H' is constructed as follows. The vertex set is $V(H') = \bigcup_{G' \in \mathcal{G}} V(H'_G)$. The terminal set is the union of T and the set of all vertices of G that appear in more than one graphs in $\mathcal{G}$. The edge set of H' is simply the union of all edges $E(H_{G'})$. Note that $V(H')$, according to our definition, indeed contains all vertices in any graph $G' \in \mathcal{G}$, so the H' is well-defined, and it is easy to verify that $|V(H')| \leq \sum_{G' \in \mathcal{G}} |V(H_{G'})|$.

It remains to show that the graph H' constructed above is a $(1 + \varepsilon)$ cut sparsifier of G with respect to T. Let (T^1, T^2) be a partition of T into two subsets.

On the one hand, consider a min-cut $\hat{E}$ in G separating T^1 from T^2, and let (V^1, V^2) be the vertex partition of $V(G)$ formed by the cut $\hat{E}$, where $T^1 \subseteq V^1$ and $T^2 \subseteq V^2$. We construct a cut E' in H' with $w(E') \leq (1 + \varepsilon) \cdot w(\hat{E})$ as follows. For each $G' \in \mathcal{G}$, define $T^1_{G'} = T_{G'} \cap V^1$ and $T^2_{G'} = T_{G'} \cap V^2$, and let $E'_{G'}$ be a min-cut in $H'_{G'}$ separating $T^1_{G'}$ from $T^2_{G'}$. Then we let $E' = \bigcup_{G' \in \mathcal{G}} E'_{G'}$.

First we show that $w(E') \leq (1 + \varepsilon) \cdot w(\hat{E})$. Recall that the edge sets $\{E(G') \mid G' \in \mathcal{G}\}$ partition $E(G)$. For each $G' \in \mathcal{G}$, denote $\hat{E}_{G'} = \hat{E} \cap E(G')$, so the sets $\left\{\hat{E}_{G'} \mid G' \in \mathcal{G}\right\}$ partitions $\hat{E}$ and so

$w(\hat{E}) = \sum_{G' \in \mathcal{G}} w(\hat{E}_{G'})$. Note that for each $G' \in \mathcal{G}$, the set $\hat{E}_{G'}$ separates $T^1_{G'}$ from $T^2_{G'}$, and by definition we have $w(E'_{G'}) \leq (1 + \varepsilon)w(\hat{E}_{G'})$. Altogether, $w(E') \leq (1 + \varepsilon) \cdot w(\hat{E})$.

Second we show that E' indeed separates T^1 from T^2 in H'. For each $G' \in \mathcal{G}$, denote by $(V^1_{G'}, V^2_{G'})$ the partition formed by the cut $E'_{G'}$, where $T^1_{G'} \subseteq V^1_{G'}$ and $T^2_{G'} \subseteq V^2_{G'}$. We let $V^1_{H'} = \bigcup_{G' \in \mathcal{G}} V^1_{G'}$ and $V^2_{H'} = \bigcup_{G' \in \mathcal{G}} V^2_{G'}$. It is easy to verify that (i) $V^1_{H'}$ and $V^2_{H'}$ partition $V(H')$; (ii) $T^1 \subseteq V^1_{H'}$; (iii) $T^2 \subseteq V^2_{H'}$; and (iv) all edges in $E_{H'}(V^1_{H'}, V^2_{H'})$ lies in E', so E' indeed separates T^1 from T^2 in H'.

On the other hand, consider a min-cut E' in H' separating T^1 from T^2, and let $(V^1_{H'}, V^2_{H'})$ be the corresponding vertex partition of $V(H')$, where $T^1 \subseteq V^1$ and $T^2 \subseteq V^2$. We construct a cut $\hat{E}$ in G with $w(\hat{E}) \leq (1+\varepsilon) \cdot w(E')$ as follows. For each $G' \in \mathcal{G}$, define $T^1_{G'} = T_{G'} \cap V^1_{H'}$ and $T^2_{G'} = T_{G'} \cap V^2_{H'}$, and let $\hat{E}_{G'}$ be a min-cut in G' separating $T^1_{G'}$ from $T^2_{G'}$. Then we let $\hat{E} = \bigcup_{G' \in \mathcal{G}} \hat{E}_{G'}$. The arguments for showing that $w(\hat{E}) \leq (1+\varepsilon) \cdot w(E')$ and that $\hat{E}$ indeed separates T^1 from T^2 in G are symmetric.

A.2   Proof of Lemma 14

Let P be the Φ-pattern shortest path connecting v to any vertex on R. Let z be the R-endpoint of P, and let ℓ be the total length of P. Vertex z separates R into two subpaths, that we call the *forward* subpath from z to the one endpoint of R and the *backward* subpath from z to the other endpoint of R. We now iteratively computes a set C_F of vertices in the forward subpath as follows. Initially set $C_F = \emptyset$ and a variable vertex $u = z$. In each iteration, we find the first vertex u' lying in the forward direction from u, such that $\mathsf{dist}^\Phi(v, u) + \mathsf{dist}(u, u') > (1 + \varepsilon) \cdot \mathsf{dist}^\Phi(v, u')$. We add u to C_F and update $u \leftarrow u'$. We continue until we cannot find any such u' in the forward path and obtain the resulting C_F. We then computes a set C_B of vertices in the back subpath in the same way, and eventually returns $C(v, R, \Phi) = C_F \cup C_B$ as the Φ-respecting ε-cover. Clearly, by definition we indeed get a Φ-respecting ε-cover of v in R. The next claim shows that $|C(v, R, \Phi)| = O(1/\varepsilon)$.

Claim 36. The number of iterations is $O(1/\varepsilon)$.

Proof. TOPROVE 18   □

References

[AGK14]   Alexandr Andoni, Anupam Gupta, and Robert Krauthgamer. Towards $(1+\varepsilon)$-approximate flow sparsifiers. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 279–293. SIAM, 2014.

[Ben96]   András A Benczúr. Approximate s-t min-cuts in $\tilde{O}(n^2)$ time. In *Proc. 28th ACM Symposium on Theory of Computing, 1996*, 1996.

[BSS12]   Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

[CDK⁺21] Parinya Chalermsook, Syamantak Das, Yunbum Kook, Bundit Laekhanukit, Yang P Liu, Richard Peng, Mark Sellke, and Daniel Vaz. Vertex sparsification for edge connectivity. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1206–1225. SIAM, 2021.

[CGH+20] Li Chen, Gramoz Goranci, Monika Henzinger, Richard Peng, and Thatchaphol Saranurak. Fast dynamic cuts, distances and effective resistances via vertex sparsifiers. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1135–1146. IEEE, 2020.

[Chu12] Julia Chuzhoy. On vertex sparsifiers with steiner nodes. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 673–688, 2012.

[CKT22] Hsien-Chih Chang, Robert Krauthgamer, and Zihan Tan. Near-linear ε-emulators for planar graphs. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022.

[CLLM10] Moses Charikar, Tom Leighton, Shi Li, and Ankur Moitra. Vertex sparsifiers and abstract rounding algorithms. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 265–274. IEEE, 2010.

[CT24a] Yu Chen and Zihan Tan. Lower bounds on 0-extension with steiner nodes. *arXiv preprint arXiv:2401.09585*, 2024.

[CT24b] Yu Chen and Zihan Tan. On (1+ eps)-approximate flow sparsifiers. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1568–1605. SIAM, 2024.

[DGGP19] David Durfee, Yu Gao, Gramoz Goranci, and Richard Peng. Fully dynamic spectral vertex sparsifiers and applications. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 914–925, 2019.

[DKV24] Syamantak Das, Nikhil Kumar, and Daniel Vaz. Nearly-tight bounds for flow sparsifiers in quasi-bipartite graphs. *arXiv preprint arXiv:2407.09433*, 2024.

[DR96] Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

[FH10] Wai Shing Fung and Nicholas JA Harvey. Graph sparsification by edge-connectivity and random spanning trees. *arXiv preprint arXiv:1005.0265*, 2010.

[GHP17] Gramoz Goranci, Monika Henzinger, and Pan Peng. Improved guarantees for vertex sparsification in planar graphs. *arXiv preprint arXiv:1702.01136*, 2017.

[GR17] Gramoz Goranci and Harald Räcke. Vertex sparsification in trees. In *Approximation and Online Algorithms: 14th International Workshop, WAOA 2016, Aarhus, Denmark, August 25–26, 2016, Revised Selected Papers*, pages 103–115. Springer, 2017.

[GRST21] Gramoz Goranci, Harald Räcke, Thatchaphol Saranurak, and Zihan Tan. The expander hierarchy and its applications to dynamic graph algorithms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2212–2228. SIAM, 2021.

[HKNR98] Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar Ragde. Characterizing multiterminal flow networks and computing flows in networks of small treewidth. *Journal of Computer and System Sciences*, 57(3):366–375, 1998.

[JLLS23]   Arun Jambulapati, James R Lee, Yang P Liu, and Aaron Sidford. Sparsifying sums of norms. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1953–1962. IEEE, 2023.

[Kar93]   David R Karger. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. In *SODA*, volume 93, pages 21–30, 1993.

[Kar98]   Alexander V Karzanov. Minimum 0-extensions of graph metrics. *European Journal of Combinatorics*, 19(1):71–101, 1998.

[Kar99]   David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[KM23]   Robert Krauthgamer and Ron Mosenzon. Exact flow sparsification requires unbounded size. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2354–2367. SIAM, 2023.

[KPZP17]   Nikolai Karpov, Marcin Pilipczuk, and Anna Zych-Pawlewicz. An exponential lower bound for cut sparsifiers in planar graphs. *arXiv preprint arXiv:1706.06086*, 2017.

[KR13]   Robert Krauthgamer and Inbal Rika. Mimicking networks and succinct representations of terminal cuts. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1789–1799. SIAM, 2013.

[KR14]   Arindam Khan and Prasad Raghavendra. On mimicking networks representing minimum terminal cuts. *Information Processing Letters*, 114(7):365–371, 2014.

[KR17]   Robert Krauthgamer and Inbal Rika. Refined vertex sparsifiers of planar graphs. *arXiv preprint arXiv:1702.05951*, 2017.

[KS98]   P. N. Klein and S. Subramanian. A fully dynamic approximation scheme for shortest paths in planar graphs. *Algorithmica*, 22(3):235–249, 1998.

[KW12]   Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 450–459. IEEE, 2012.

[Liu20]   Yang P Liu. Vertex sparsification for edge connectivity in polynomial time. *arXiv preprint arXiv:2011.15101*, 2020.

[LM10]   F Thomson Leighton and Ankur Moitra. Extensions and limits to vertex sparsification. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 47–56. ACM, 2010.

[MM10]   Konstantin Makarychev and Yury Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 255–264. IEEE, 2010.

[Moi09]   Ankur Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 3–12. IEEE, 2009.

[Sau72]   Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[She72]   Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

[SS11]   Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[Tho04]   Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM (JACM)*, 51(6):993–1024, 2004.