Deterministic Lower Bounds for k-Edge Connectivity in the Distributed Sketching Model

PETER ROBINSON, Augusta University, Georgia, USA
MING MING TAN, Augusta University, Georgia, USA

We study the k-edge connectivity problem on undirected graphs in the distributed sketching model, where we have n nodes and a referee. Each node sends a single message to the referee based on its 1-hop neighborhood in the graph, and the referee must decide whether the graph is k-edge connected by taking into account the received messages.

We present the first lower bound for deciding a graph connectivity problem in this model with a deterministic algorithm. Concretely, we show that the worst case message length is $\Omega(k)$ bits for k-edge connectivity, for any super-constant $k = O(\sqrt{n})$. Previously, only a lower bound of $\Omega\left(\log^3 n\right)$ bits was known for (1-edge) connectivity, due to Yu (SODA 2021). In fact, our result is the first super-polylogarithmic lower bound for a connectivity decision problem in the distributed graph sketching model.

To obtain our result, we introduce a new lower bound graph construction, as well as a new communication complexity problem that we call UniqueOverlap. As this problem does not appear to be amenable to reductions to existing hard problems such as set disjointness or indexing due to correlations between the inputs of the three players, we leverage results from cross-intersecting set families to prove the hardness of UniqueOverlap for deterministic algorithms in the 3-party model with simultaneous messages. Finally, we obtain the sought lower bound for deciding k-edge connectivity via a novel simulation argument that, in contrast to previous works, does not introduce any probability of error and thus works for deterministic algorithms.

CCS Concepts: • **Theory of computation → Distributed algorithms**; **Communication complexity**.

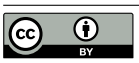
Additional Key Words and Phrases: distributed graph sketching, communication complexity lower bound

ACM Reference Format:
Peter Robinson and Ming Ming Tan. 2026. Deterministic Lower Bounds for k-Edge Connectivity in the Distributed Sketching Model. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 101 (May 2026), 21 pages. https://doi.org/10.1145/3801897

1 Introduction

We consider the distributed graph sketching model [6], where n distributed nodes each observe their list of neighbors in a graph. Every node sends a single message to a central entity, called referee, who must compute a function of the graph based on the received messages. In this setting, any graph problem can be solved trivially by instructing each one of the n nodes to simply include their entire neighborhood information in the message to the referee. To obtain more efficient algorithms that scale to large graphs, the main goal is to keep the maximum size of these messages (also called *sketches*) as small as possible, preferably polylogarithmic in n. In contrast, the full list of neighbors of a node could be as large as $\Theta(n)$ bits, under the standard assumption that the node IDs are a permutation of $\{1, \ldots, n\}$. Consequently, nodes can afford to convey only incomplete information

Authors' Contact Information: Peter Robinson, perobinson@augusta.edu, School of Computer & Cyber Sciences, Augusta University, Augusta, Georgia, USA; Ming Ming Tan, mtan@augusta.edu, School of Computer & Cyber Sciences, Augusta University, Augusta, Georgia, USA.



ACM 2836-6573/2026/5-ART101
https://doi.org/10.1145/3801897

(hence the name "sketch") of their local neighbors to the referee, which may paint a somewhat ambiguous picture of the actual graph. While it may be tempting to conclude that small sketches do not allow us to solve interesting graph problems in this model, the surprising breakthrough of Ahn, Guha, and McGregor [1] showed that several fundamental graph problems, such as connectivity and spanning trees, can indeed be solved with sketches of only $O\left(\log^3 n\right)$ bits, if nodes have access to shared randomness and one is willing to accept a polynomially small probability of error. Their technique (called "AGM sketches") paved the way for sketch-based graph algorithms for many other graph problems, including vertex connectivity [8] and approximate graph cuts [2]; see the survey of [19] for a list of additional graph problems.[1]

1.1 Randomized Lower Bounds for Connectivity Problems

The question whether AGM sketches are optimal for connectivity problems remained an open problem for several years, until the seminal work by Nelson and Yu [21] showed that $\Omega\left(\log^3 n\right)$ bits are indeed required for computing spanning forests in the distributed graph sketching model. The main idea of their lower bound is to use a reduction to a variant of the universal relation problem [16] in 2-party communication complexity, denoted by $\mathsf{UR}^{\subset}$, where Alice starts with a subset S of some universe U, and Bob gets a proper subset $T \subset S$. Alice sends a single message to Bob, who must output some element in $S \setminus T$. In [14], Kapralov, Nelson, Pachoki, Wang, and Woodruf showed that $\Omega\left(\log^3 n\right)$ bits are required for solving $\mathsf{UR}^{\subset}$ with high probability. To obtain a reduction to $\mathsf{UR}^{\subset}$ in the distributed sketching model, [21] define a tripartite graph on vertex sets V^l, V^m, and a much smaller set V^r. Each node $v \in V^m$ has a set of unique neighbors in V^l (not shared with any other node in V^m) as well as some neighbors in V^r. Since every node in V^l is connected to at most one node $v \in V^m$, any spanning forest must include some edge between v and a node in V^r. The neighborhood of a randomly chosen $v^* \in V^m$ is determined by the input of $\mathsf{UR}^{\subset}$ such that Alice's set S corresponds to all of v^*'s neighbors, whereby Bob's set T specifies the subset of neighbors in V^l. They show that the players can jointly simulate the nodes in V^l and V^m (but not V^r!), whereby Bob also simulates the referee. From the resulting spanning forest, Bob can extract an edge between v^* and V^r, which corresponds to an element in $S \setminus T$, and thus solves $\mathsf{UR}^{\subset}$. Notably absent in their simulation are the nodes in V^r, which neither Alice or Bob can simulate, due to their lack of knowledge of set $S \setminus T$. In fact, trying to faithfully simulate *every* node executing a distributed sketching algorithm in a 2-party model poses a major technical challenge since every edge is shared between its neighbors. However, [21] show that choosing V^r to be polynomially smaller than V^m suffices to overcome this obstacle: Since each $w \in V^r$ will be a neighbor of many $v \in V^m$ and w does not know which of them is the important node v^* (as all of these neighborhoods are sampled from the same hard input distribution for $\mathsf{UR}^{\subset}$), it follows that the amount of information that the referee can learn about the neighbors of v^* from the sketches of the nodes in V^r is only (roughly) $\tilde{O}(\frac{|V^r|}{|V^m|}) = o(1)$, and thus negligible. Hence, Alice and Bob can omit these messages in their simulation altogether, while only causing a small increase in the overall error probability, due to Pinsker's inequality [23]. We remark that the actual lower bound construction of [21] uses multiple copies of this basic building block.

More recently, Robinson [24] extended the approach of [21] to constructing a k-edge connected spanning subgraph (k-ECSS), with the main difference being a modified graph construction that enables a reduction to the multi-output variant of universal relation, called $\mathsf{UR}^{\subset}_k$, which has a lower bound of $\Omega\left(k \log^2 n\right)$ bits [14] and requires Bob to find k elements of $S \setminus T$. The simulation argument in [24] does not directly use Pinsker's inequality as in [21], and instead obtains a slightly

[1] Strictly speaking, [1] showed these results for fully dynamic graph streams in the semi-streaming model, but it is straightforward to adapt them to the distributed graph sketching model, as observed in [7].

smaller, but nevertheless nonzero, probability of error, by devising a mechanism of reconstructing the output when omitting the sketches of the nodes in V^r. At the risk of stating the obvious, we point out that this result does not have any implications for the problem of *deciding* whether a given graph is k-edge connected, which is the focus of the current paper.

In [26], Yu proved that the decision problem of graph connectivity is subject to the same lower bound on the message length as computing a spanning forest, which fully resolved the complexity of connectivity in the distributed sketching model. For this purpose, [26] introduced and proved the hardness of a new decision version of universal relation, called $\mathsf{UR}^{\subset}_{\mathrm{dec}}$. Just as for the $\mathsf{UR}^{\subset}$ problem, Alice again gets a set $S \subseteq U$. However, Bob not only gets $T \subset S$ but, in addition, he also knows a partition (P_1, P_2) of $U \setminus T$. The inputs adhere to the promise that either $S \setminus T \subseteq P_1$ or $S \setminus T \subseteq P_2$, and, upon receiving Alice's message, Bob needs to decide which is the case. In order to embed an instance of $\mathsf{UR}^{\subset}_{\mathrm{dec}}$ in a graph, [26] adapted the construction of [21] by partitioning the set V^r into V^r_1 and V^r_2, and by sampling the neighborhood of each node in V^m according to the hard distribution of $\mathsf{UR}^{\subset}_{\mathrm{dec}}$. Following the overall simulation approach of [21], Alice's input of $\mathsf{UR}^{\subset}_{\mathrm{dec}}$ is embedded at some special node $v^* \in V^m$. Accordingly, the partition-promise of $\mathsf{UR}^{\subset}_{\mathrm{dec}}$ translates to each $v \in V^m$ having all its V^r-neighbors in either V^r_1 or V^r_2.

We point out that there does not appear to be a natural extension of the connectivity lower bound of [26] to k-edge connectivity. Even though $\mathsf{UR}^{\subset}_{\mathrm{dec}}$ can be generalized to $k > 1$, proving a lower bound that scales linearly with k for this generalization appears to be far from straightforward. Interestingly, the situation is reversed for the standard "search" version of $\mathsf{UR}^{\subset}$, where the proof of the lower bound for $\mathsf{UR}^{\subset}_k$ is technically less involved than the corresponding result for $\mathsf{UR}^{\subset}$, see [14]. A second obstacle is that, in contrast to the problem of finding a k-edge connected spanning subgraph [24], designing a suitable lower bound graph for deciding k-edge connectivity is itself nontrivial. This is because choosing the neighborhood of every node in V^m according to the hard distribution of the generalized variant of $\mathsf{UR}^{\subset}_{\mathrm{dec}}$ and embedding it in the constructions of [21, 26] may result in a graph that is very likely to be k-edge connected, independently of the neighborhood of the special node v^*.

1.2 Deterministic Lower Bounds for Graph Connectivity Problems?

Interestingly, none of the above mentioned lower bound results provide a straightforward way towards obtaining stronger bounds for deterministic graph sketching algorithms (that never fail): Since any deterministic one-way algorithm for $\mathsf{UR}^{\subset}$, $\mathsf{UR}^{\subset}_k$, or $\mathsf{UR}^{\subset}_{\mathrm{dec}}$ needs to send Alice's entire input to Bob,[2] it may appear that, at first glance, we can simply "plug in" these results for deterministic protocols and directly obtain stronger lower bounds in the graph sketching model. This intuition, however, turns out to be misleading, as a major technical challenge is that the simulations used in [21, 24, 26] are themselves randomized and, more importantly, are necessarily "lossy" in the sense that they introduce a nonzero probability of error, due to forgoing the simulation of some nodes—in particular, all nodes in V^r. This poses a major technical obstacle towards obtaining lower bounds for algorithms that fail with exponentially small probability, including deterministic ones.[3]

Obtaining lower bounds on the *deterministic* complexity of distributed graph sketches for connectivity is listed under "hard open problems" as a "longstanding open question" in [4] (p. 102). Yet, there has been no progress on this class of problems to date, which is indicative of the difficulty of finding lower bounds for deterministic connectivity problems. Thus, besides the actual lower bound for k-connectivity, a major contribution of our work is to show that 0-error simulations

[2]While there is no published deterministic lower bound for $\mathsf{UR}^{\subset}_{\mathrm{dec}}$, the same communication complexity bound as for the search version $\mathsf{UR}^{\subset}$ follows from standard fooling set and counting arguments for one-way communication; see, e.g., [17].
[3]We discuss prior work on lower bounds for other graph problems in the sketching model in Section 1.4.

are feasible in this model, which may pave the way for other deterministic lower bounds in graph sketching.

1.3 Our Contributions

In this work, we take the first step toward determining lower bounds for deterministic connectivity algorithms in the graph sketching model. In more detail, we focus on the problem of deciding k-edge connectivity, and we show that any deterministic sketching algorithm has a worst case sketch length that is almost linear (in k):

Theorem 1. *Every deterministic algorithm that decides k-edge connectivity on n-node graphs in the distributed sketching model has a worst case message length of $\Omega(k)$ bits, for any super-constant $k = k(n) \leq \gamma\sqrt{n}$, where $\gamma > 0$ is a suitable constant.*

The proof of Theorem 1 requires us to overcome several technical challenges:

- To obtain a graph that is hard for deciding k-edge connectivity in the distributed sketching model, we need to find a suitable graph construction. We cannot directly extend the constructions used for deciding graph connectivity in [26], as this crucially rests on the assumption that the neighborhood of every node in V^m (see Sec. 1.1) must be chosen from the *same* distribution. In fact, the straightforward generalization of this idea to k-edge connectivity would yield a graph that is very likely to be k-edge connected, for large values of k. In this work, we propose a new construction that does not preserve the uniformity property of [26] regarding the input distribution. Instead, there is a special node v_σ, with a somewhat larger degree than its peers, which we connect in a way such that the k-edge connectivity of the graph entirely rests on the choice of neighborhood for this node. Nevertheless, we ensure that its neighbors have no easy way of distinguishing v_σ from the vast majority of unimportant nodes. This approach introduces additional technical difficulties that we address in our simulation.
- As elaborated in more detail in Section 1.2, we need to depart from the well-trodden path of using the "partial simulation" argument pioneered in [21] (and also used in the work of [26] and [24]), where some nodes are omitted from being simulated, as this would result in a nonzero error algorithm. To avoid this pitfall, we design a completely different simulation that is suitable for deterministic algorithms. We point out that the main technical difficulty of the distributed sketching model, namely the "sharing" of edges between nodes, also surfaces in our setting and prevents us from devising a simulation, where every node v's neighborhood is part of the input of one of the simulating players. Instead, we exploit some structural properties of the algorithm at hand that must hold for any deterministic protocol. In more detail, the main idea behind our approach is that it suffices to compute a sketch for v that looks "compatible" with our (limited) knowledge of v's neighborhood, and we prove that this indeed ensures an error free simulation.
- Our simulation works in a simultaneous 3-party model of communication complexity, where two players, Alice and Bob, each send a single message to Charlie who computes the output. For this setting, we introduce the UniqueOverlap problem, which abstracts the core difficulty present in our lower bound graph construction: That is, nodes are unaware which of their incident edges point to the crucial node v_σ, even though v_σ itself is aware of its special role. We capture this property in the communication complexity setting by equipping Alice and Bob each with an input vector that has a single common coordinate whose XOR is 1. While Charlie knows the location of this coordinate, he does not know the values of the respective bits of Alice and Bob, which are needed for the correct output. The hardness of UniqueOverlap does not appear to follow from reductions to standard problems such as set

disjointness or variants of indexing (see Sec. 4.1 for a more detailed discussion), due to the correlation between the inputs of the three players and since we consider ternary input vectors, whose entries may be either 0, 1, or $\perp$. A further complication is that we need to assume Charlie knows the entire support (i.e., the indices of all non-$\perp$ entries) of Alice's and Bob's inputs. Instead, we devise a purely combinatorial argument based on properties of cross-intersecting set families to show the following result:

Theorem 2. *For any sufficiently large m, there exists an $s = \Theta(m)$ such that the deterministic one-way communication complexity of* $\mathsf{UniqueOverlap}_{m,s}$ *in the simultaneous 3-party model is $\Omega(m)$, where each m-length input vector has support s.*

1.4 Additional Related Work

The distributed graph sketching model was first introduced by Becker, Matamala, Nisse, Rapaport, Suchan, and Todinca in [6], where they proved the hardness of several fundamental graph problems for deterministic sketching algorithms: In particular, they showed that problems including deciding whether the diameter is at most 3, and whether the graph contains a certain subgraph such as a triangle or a square, require sketches of large (i.e., super-polylogarithmic) size. Their approach works by showing that the existence of an efficient protocol for any of these problems can be leveraged for reconstructing graphs with certain properties using similar-sized sketches. However, as pointed out in [6], their approach does not extend to graph connectivity problems.

The model was further explored by Becker, Montealegre, Rapaport, and Todinca in [7], where they show separations between the power of deterministic, private randomness and public randomness algorithms. Apart from the aforementioned works of [21, 24, 26] on connectivity-related problems, Assadi, Kol, and Oshman [5] showed a lower bound of $\Omega\left(n^{1/2-\varepsilon}\right)$ bits on the sketch size for computing a maximal independent set. For the distributed sketching model with *private* randomness, Holm, King, Thorup, Zamir, and Zwick [10] showed that computing a spanning tree is possible with sketches of $O(\sqrt{n}\log n)$ bits.

We point out that the distributed graph sketching model is known to be equivalent to a single-round variant of the broadcast congested clique, as observed in several earlier works [5, 12], where each one of n nodes can broadcast a single message per round that becomes visible to all other nodes at the end of the current round. In that context, the worst case sketch size corresponds to the available bandwidth, i.e., size of the broadcast message. Several works have studied tradeoffs between the number of rounds and the message size in the (multi-round) broadcast congested clique: Pai and Pemmaraju [22] showed that $\Omega\left(\frac{\log n}{b}\right)$ rounds are required if nodes can broadcast messages of size b. The work of Montealegre and Todinca [20] revealed that there is a deterministic r-round connectivity algorithm that sends messages of size $O(n^{1/r}\log n)$. Subsequently, Jurdzinski and Nowicki [13] obtained an improved round complexity of $O(\log n/\log\log n)$ when considering the standard assumption of $O(\log n)$ bits per message.

The related problem of computing a k-edge connectivity certificate has also been studied in dynamic graph streams. The recent work of Sawettamalya and Yu [25] give an algorithm that uses $O\left(n\log^2 n\cdot\max\{k\log n\log k\}\right)$ bits of space and thus nearly matches the lower bound of $\Omega\left(n\log^2 n\cdot\max\{k,\log n\}\right)$ established in [21] and [24].

1.5 Notation and Preliminaries

In the distributed sketching model each node v is equipped with an ID of $O(\log n)$ bits. We sometimes abuse notation and use v to refer to either the node itself or its ID, depending on the context. Thus, when referring to the neighborhood of a node, we also mean the list of IDs associated to the node's neighbors.

We use $[m]$ to denote the set of integers from 1 to m. For a set S and some positive integer m, we borrow the standard notation from extremal combinatorics (e.g., see [11]) and use $\binom{S}{m}$ to denote the set family of all m-element subsets of S.

1.6 Roadmap

We structure the proof of Theorem 1 as follows: In Section 2, we introduce our new graph construction and analyze how its properties relate to k-connectivity. Then, we show in Section 3 that, for any given deterministic algorithm that sends short sketches, there exists an instance of this lower bound graph, such that many nodes send the same message for certain pairs of inputs. After introducing and proving the hardness of the UniqueOverlap problem in Section 4, we show in Section 5 how to solve it by simulating a k-edge connectivity algorithm on our lower bound graph. In Section 6, we combine these results to complete the proof of Theorem 1. Finally, we discuss some open problems in Section 7.

2 The Lower Bound Graph

In this section, we define the class of lower bound graphs for proving Theorem 1. Consider parameters n and k, where

$$\omega(1) \leq k \leq \gamma\sqrt{n}, \tag{1}$$

for some suitable constant $\gamma > 0$, which, in particular, rules out k being a constant. We define the class of n-node graphs $\mathcal{G}_{k,n}$ on a vertex set $V \cup W \cup \{u_A, u_B\}$, such that

$$|V| = n - \left\lfloor \sqrt{n} \right\rfloor - 2 \tag{2}$$

$$|W| = \left\lfloor \sqrt{n} \right\rfloor. \tag{3}$$

The set W is further partitioned into vertex sets A and B, each of size at least k. Every vertex has a unique *identifier* (ID), and we use any fixed arbitrary ID assignment from the set $[n]$ for all graphs in $\mathcal{G}_{k,n}$. Every graph $G \in \mathcal{G}$ is associated with a *determining index* $\sigma \in [|V|]$. As we will see below, this name is motivated by the fact that the edges in the cut $E(v_\sigma, A \cup B)$ determine whether the graph is k-connected.

Next, we define the edges of G:

(E1) $G[A]$ and $G[B]$ are cliques.
(E2) For every $w_i \in A$, we add the edge $\{u_A, w_i\}$, and, similarly, for every $w_j \in B$, we add $\{u_B, w_j\}$.

We now define the edges in the cut (V, W): For every $v_i \in V$ $(i \neq \sigma)$, the incident edges are such that exactly one of the following two properties hold:

(E3) $|E(v_i, A)| \geq 1$ and $|E(v_i, B)| = 0$, or $|E(v_i, A \cup B)| = 0$; and there are k parallel edges between u_A and v_i.
(E4) $|E(v_i, A)| = 0$ and $|E(v_i, B)| \geq 1$, and there are k parallel edges between u_B and v_i.

We say that a node v_i is *A-restricted* if it satisfies (E3) and call v_i *B-restricted* if (E4) holds. Note that (E3) holds for a node v_i even in the case that there are no edges at all between v_i and $A \cup B = W$.

Node v_σ, on the other hand, has exactly $2k - 1$ edges to nodes in W, and

(E5) there are k parallel edges between v_σ and u_A.

Moreover, one of the following two conditions hold:

(C0) $|E(v_\sigma, A)| \geq k$ and $|E(v_\sigma, B)| \leq k - 1$;
(C1) $|E(v_\sigma, A)| \leq k - 1$ and $|E(v_\sigma, B)| \geq k$.

We point out that (E3), (E4), and (E5) require the input graph to be a multi-graph. This can be avoided by simply replacing u_A and u_B by cliques of size k if desired, without changing the asymptotic

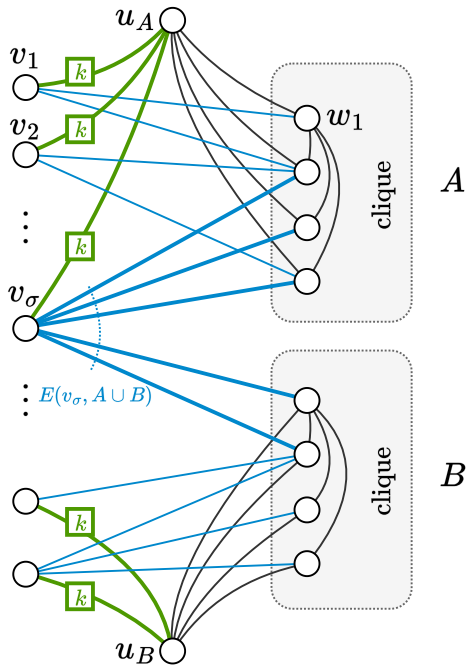


Fig. 1. An instance of a lower bound graph for $k = 3$. Notice that v_σ is the only vertex with edges in both A and B. In this example, condition (C0) holds, since v_σ has k edges to A. Thus, the graph is not k-edge connected.

bounds. Apart from the list of their neighbors, we provide additional power to the algorithm by revealing to each node v_i whether it is A-restricted, B-restricted, or v_σ. Clearly, this can only strengthen the lower bound. Figure 1 shows an example of a graph $G \in \mathcal{G}_{k,n}$.

We now state some properties of this construction. The next lemma is immediate from the description:

Lemma 1. *Every node $v_i \in V$ is either v_σ, A-restricted, or B-restricted. Moreover, v_i knows this information as part of its initial state.*

Lemma 2. *Graph $G \in \mathcal{G}_{k,n}$ is k-edge connected if and only if (C1) holds.*

Proof. Let $V_A \subseteq V$ be the subset of nodes that are A-restricted, i.e., satisfy (E3), and also include v_σ in V_A. Similarly, we define $V_B \subseteq V$ to contain all B-restricted nodes, which satisfy (E4). Every $v_j \in V_A$ has k parallel edges to u_A, which itself has an edge to each node in A. Since A is a clique of size at least k, it follows that there are at least k edge-disjoint paths between every pair of nodes in $G[A \cup \{u_A\} \cup V_A]$, which implies that $G[A \cup \{u_A\} \cup V_A]$ is k-edge connected. A similar argument shows that B, u_B, and V_B induce a k-edge connected subgraph. By properties (E1)-(E5), any edge in the cut $(A \cup \{u_A\} \cup V_A, B \cup \{u_B\} \cup V_B)$ must be in $E(v_\sigma, B)$. Thus, the graph is k-edge connected if and only if Case (C1) occurs. □

3 Existence of Indistinguishable Separated Pairs

As our overall goal is to prove Theorem 1, we assume towards a contradiction that each node sends at most $L = o(k)$ bits throughout this section.

For a node $v_i \in V$, we call the subset of its neighbors in W its *W-neighborhood*. Here, our main focus will be to show that there is a way to choose the partition $\{A, B\}$ of W that has properties suitable for our simulation in Section 5.

Definition 1 (Separated Pair)**.** We say that two W-neighborhoods S_0 *and* S_1 *form a separated pair for* v_σ *with respect to* A *and* B, when the following are true:

- if v_σ's W-neighborhood corresponds to S_0, then condition (C0) holds. In other words, $|S_0 \cap A| \geq k$ and $|S_0 \cap B| \leq k-1$.
- if v_σ's W-neighborhood is given by S_1, then condition (C1) holds. Thus, $|S_1 \cap A| \leq k-1$ and $|S_1 \cap B| \geq k$.

Hence, if v_σ's W-neighborhood is S_1, the graph is k-edge connected whereas if it is S_0, then the graph is not k-edge connected.

Since node v_σ has exactly $2k-1$ neighbors in W according to our lower bound graph description, we know that the set of all W-neighborhoods of v_σ is of size $\binom{|W|}{2k-1}$. For technical reasons, however, we need to ensure that any pair of possible W-neighborhoods of v_σ have an intersection that is at most a small constant fraction of k. The proof of the next lemma follows via the probabilistic method and similar to Lemma 6 in [14].[4] We include a full proof in Appendix A for completeness.

Lemma 3. *Consider a set family W, any small $\varepsilon > 0$, and any positive integer $d \leq \frac{\varepsilon|W|}{2e^{2+4/\varepsilon}}$. There exists a set family $\mathcal{S} \subseteq \binom{W}{d}$ such that, for any two distinct $S_1, S_2 \in \mathcal{S}$, we have $|S_1 \cap S_2| \leq \frac{\varepsilon d}{2}$, and $|\mathcal{S}| \geq e^{d-1}$.*

Instantiating Lemma 3 with $d = 2k-1$ and a suitable small constant $\varepsilon > 0$, shows that

$$|\mathcal{S}| \geq e^{2k-2}. \tag{4}$$

Given an arbitrary set D, we say that $\{\mathcal{B}_1, \ldots, \mathcal{B}_\ell\}$ is a *partition $\mathcal{P}$ of D* and each set $\mathcal{B}_i \subseteq D$ is called a *block of $\mathcal{P}$*, if $\bigcup_{i=1}^{\ell} \mathcal{B}_i = D$ and all blocks are pairwise disjoint. For a fixed partition $\{A, B\}$ of W and any $S \in \mathcal{S}$, we define the *projections* $\phi_A(S) = S \cap A$ and $\phi_B(S) = S \cap B$, and we also define the resulting image sets $\Phi_A = \{\phi_A(S) \mid S \in \mathcal{S}\}$ and $\Phi_B = \{\phi_B(S) \mid S \in \mathcal{S}\}$. Consider the algorithm executed by some node v_i. Let us assume that, for the time being, $\sigma = i$, and that node v_i is equipped with ID i. Together with the node IDs in its neighborhood, this fully determines the message v_i sends to the referee. In particular, v_i can send at most 2^L distinct messages, and hence the given algorithm induces a partition $\mathcal{P}_{i,\sigma}$ of $\mathcal{S}$ into at most 2^L distinct *blocks*, where each block is a set of W-neighborhoods for which v_i sends the same message to Charlie. Let $\mathcal{B}_{i,\sigma}$ denote the largest one of these blocks, and observe that $\mathcal{B}_{i,\sigma} \geq \frac{|\mathcal{S}|}{2^L}$. We emphasize that the $\mathcal{B}_{i,\sigma}$ depends on the ID i of v_i; it may be the case that the resulting set $\mathcal{B}_{j,\sigma}$ contains a completely different set of neighborhoods for node v_j ($i \neq j$). Recall from Lemma 1 that a node $v_i \in V$ may deduce whether $\sigma = i$ by inspecting its degree and, consequently, the algorithm at v_i may behave differently when $\sigma \neq i$ compared to the case where v_i is either A-restricted or B-restricted. That is, for a given partition $\{A, B\}$ of W, the message sent by any A-restricted node $v_i \in V$ ($i \neq \sigma$) induces a partition $\mathcal{P}_{i,A}$ on Φ_A, whereas the message sent by a B-restricted node v_j induces a partition $\mathcal{P}_{j,B}$ on Φ_B.

The next definition places some additional restrictions on a separated pair with respect to the three partitions $\mathcal{P}_{i,\sigma}$, $\mathcal{P}_{i,A}$, and $\mathcal{P}_{i,B}$, which we will leverage in Section 5.

Definition 2 (Indistinguishable Separated Pair)**.** We say that S_0 *and* S_1 *form an indistinguishable separated pair for* v_i, if the following properties hold:

(i) S_0 and S_1 are in the same block of $\mathcal{P}_{i,\sigma}$.

[4] This is Lemma 7 in the full version of their paper on arXiv [15].

(ii) $\phi_A(S_0)$ and $\phi_A(S_1)$ are distinct and in the same block of $\mathcal{P}_{i,A}$, and
(iii) $\phi_B(S_0)$ and $\phi_B(S_1)$ are distinct and in the same block of $\mathcal{P}_{i,B}$.
(iv) S_0 and S_1 form a separated pair for v_i.

Lemma 4. *Suppose that we obtain the partition $\{A, B\}$ by assigning each vertex in W uniformly at random to either A or B. Consider any $v_i \in V$. Then, with probability at least $\frac{3}{4}$, there exist distinct $S_0, S_1 \in \mathcal{S}$ that form an indistinguishable separated pair for v_i.*

Proof. We start by defining additional partitions $\tilde{\mathcal{P}}_{i,A}$ and $\tilde{\mathcal{P}}_{i,B}$ that we obtain from $\mathcal{P}_{i,A}$ and $\mathcal{P}_{i,B}$, respectively, as follows: For each block $\mathcal{B} \in \mathcal{P}_{i,A}$, we define

$$\phi_A^{-1}(\mathcal{B}) = \bigcup_{T \in \mathcal{B}} \{S \in \mathcal{S} \mid \phi_A(S) = T\}.$$

Partition $\tilde{\mathcal{P}}_{i,A}$ has exactly the same number of blocks as $\mathcal{P}_{i,A}$, and we obtain $\tilde{\mathcal{B}} \in \tilde{\mathcal{P}}_{i,A}$ from $\mathcal{B} \in \mathcal{P}_{i,A}$ by defining it as the set containing all elements in $\phi_A^{-1}(\mathcal{B})$. Analogously, we construct $\tilde{\mathcal{P}}_{i,B}$, which will have the same number of blocks as $\mathcal{P}_{i,B}$, and, for each $\mathcal{B}' \in \mathcal{P}_{i,B}$, we get block $\tilde{\mathcal{B}}' \in \tilde{\mathcal{P}}_{i,B}$, which contains all elements in $\phi_B^{-1}(\mathcal{B}') = \bigcup_{T \in \mathcal{B}'} \{S \in \mathcal{S} \mid \phi_B(S) = T\}$. Thus, the number of blocks in $\tilde{\mathcal{P}}_{i,A}$, $\tilde{\mathcal{P}}_{i,B}$, and $\mathcal{P}_{i,\sigma}$ is at most 2^L, and all three of them partition the set $\mathcal{S}$ given by Lemma 3.

To show the existence of the sought sets S_0 and S_1, we need the following combinatorial statement:

Claim 1. *Let D be a set of size s, and let $\mathcal{P}_1, \ldots, \mathcal{P}_r$ be r partitions of D, such that the number of blocks in each of the partitions is at most $n' < s^{1/r+1}$. Then, for every $i \in [r]$, there exists a block B_i in $\mathcal{P}_i$ that contains at least $s^{1-i/(r+1)}$ elements that are in the same block in each of the partitions $\mathcal{P}_1, \ldots, \mathcal{P}_i$.*

Proof of Claim 1. The case $i = 1$ is immediate by the pigeonhole principle. For the inductive step, assume that the claim is true for some $i < r$. That is, there is some block $B_i \in \mathcal{P}_i$ containing at least $s^{1-i/(r+1)}$ elements that are also in the same block in each of the partitions $P_1, \ldots, P_i$. Again, by the pigeonhole principle, we conclude that there exists a block $B_{i+1} \in \mathcal{P}_{i+1}$ that has at least $s^{1-(i+1)/(r+1)}$ elements of B_i, as required. □

Instantiating Claim 1 with $n' = 2^L = 2^{o(k)}$ and partitions $\tilde{\mathcal{P}}_{i,A}$, $\tilde{\mathcal{P}}_{i,B}$, and $\mathcal{P}_{i,\sigma}$, it follows that there exists a set $\mathcal{T} \subseteq \mathcal{S}$, such that all sets in $\mathcal{T}$ are in the same block in all three partitions. Moreover, (4) tells us that

$$|\mathcal{T}| \geq |\mathcal{S}|^{1/4} \geq e^{(k-1)/2}.$$

Without loss of generality, assume that $|\mathcal{T}|$ is even, and pair up the sets in $\mathcal{T}$ in some arbitrary way. Let $\hat{\mathcal{T}}$ denote this set of pairs, and observe that

$$|\hat{\mathcal{T}}| \geq e^{(k-1)/2-\log 2}. \tag{5}$$

Consider any pair $(S_0, S_1) \in \hat{\mathcal{T}}$, and note that we have already shown that Property (i) of Definition 2 holds. To show Properties (ii) and (iii), suppose that $S_0, S_1 \in \tilde{\mathcal{B}}_A \in \tilde{\mathcal{P}}_{i,A}$. By construction of $\tilde{\mathcal{P}}_{i,A}$, we know that there exist some $T_0, T_1 \in \mathcal{B}_A$ where $\mathcal{B}_A$ is a block in $\mathcal{P}_{i,A}$, such that $T_0 = \phi_A(S_0)$ and $T_1 = \phi_A(S_1)$, and, analogously, it follows that $\phi_B(S_0)$ and $\phi_B(S_1)$ are in a common block of $\mathcal{P}_{i,B}$.

To complete the proof, we need to bound the probability that S_0 and S_1 form a separated pair. Note that this will also imply $\phi_A(S_0) \neq \phi_A(S_1)$ and $\phi_B(S_0) \neq \phi_B(S_1)$, by Definition 1 (see page 8).

Since $|S_0| = |S_1| = 2k - 1$, it must be true that either $|S_0 \cap A| \geq k$ or $|S_0 \cap B| \geq k$. Without loss of generality, we can assume the former inequality holds—otherwise, simply exchange S_0

and S_1. Recalling that $S_0, S_1 \in \mathcal{S}$, we know that $|S_0 \cap S_1| \leq \varepsilon k$, which means that, conditioned on $|S_0 \cap A| \geq k$, there are still at least

$$|S_1 \setminus S_0| \geq 2k - 1 - \lceil \varepsilon k \rceil \tag{6}$$

elements of S_1, whose random assignment is not fixed by the conditioning. Let ℓ be the largest even integer such that $\ell \leq 2k - 1 - \lceil \varepsilon k \rceil$, and note that

$$k(2-\varepsilon) - 2 \leq \ell \leq k(2-\varepsilon) - 1. \tag{7}$$

Fix some order of the elements in $S_1 \setminus S_0$, and define Z_i ($i \in [\ell]$) to be the indicator random variable that is 1 if and only if the i-th element is in B. Let $Z = \sum_{i=1}^{\ell} Z_i$. Our goal is to show that $|(S_1 \setminus S_0) \cap B| \geq Z \geq k$. In the worst case, all $\lceil \varepsilon k \rceil$ elements in $S_1 \cap S_0$ lie in A, and hence $\mathbf{E}[Z \mid |S_0 \cap A| \geq k] < k$. Therefore, we need to make use of the following anti-concentration result to show that there is a sufficiently large probability for $Z \geq k$:

Lemma 5 (Proposition 7.3.2 in [18])**.** *Let ℓ be even, $X_1, \ldots, X_\ell$ be independent random variables, where each takes the values of 0 and 1 with probability $\frac{1}{2}$. Let $X = X_1 + X_2 + \ldots X_\ell$. For any integer $t \in [\ell/8]$, it holds that $\mathbf{Pr}\left[X \geq \frac{\ell}{2} + t\right] \geq \frac{1}{15} e^{-16t^2/\ell}$.*

To apply Lemma 5, define t to be an integer such that

$$\frac{\varepsilon k + 1}{2} \leq t \leq \frac{\varepsilon k + 4}{2}, \tag{8}$$

and note that $t < \ell/8$ holds, as long as $\varepsilon \leq \frac{2}{5} - \frac{18}{5k}$ since $k = \omega(1)$. By Lemma 5, we have

$$\begin{aligned}
\mathbf{Pr}\left[Z \geq k \,\middle|\, |S_0 \cap A| \geq k\right] &\geq \mathbf{Pr}\left[Z \geq \frac{\ell}{2} + t \,\middle|\, |S_0 \cap A| \geq k\right] \\
\text{(by Lem. 5, and } t < \ell/8) \quad &\geq \frac{1}{15} \exp\left(-\frac{\ell}{4}\right) \\
\text{(by (7))} \quad &\geq \frac{1}{15} \exp\left(-\frac{k(2-\varepsilon)}{4}\right)
\end{aligned}$$

Recalling (5), it follows that the probability that none of the pairs in $\hat{\mathcal{T}}$ is a separated pair is at most

$$\left(1 - \frac{e^{-\frac{k(2-\varepsilon)}{4}}}{15}\right)^{e^{\frac{k-1}{2} - \log 2}} \leq \exp\left(-\frac{e^{-\frac{k(2-\varepsilon)}{4} + \frac{k-1}{2} - \log 2}}{15}\right) \leq \exp\left(-\frac{e^{\frac{\varepsilon k}{4} - 2}}{15}\right)$$

which is at most $\frac{1}{4}$ for sufficiently large n.

It follows that one of the pairs in $\hat{\mathcal{T}}$, all of which satisfy Properties (i)-(iii), is a separated pair with probability at least $\frac{3}{4}$, which shows Property (iv) and completes the proof of Lemma 4. □

Recall that $|V| = \Theta(n)$. For a randomly chosen partition $\{A, B\}$ of W, Lemma 4 implies that a constant fraction of the nodes $v_i \in V$ has indistinguishable separated pairs in expectation. Therefore, there must be some concrete choice $\{A^*, B^*\}$ for $\{A, B\}$ that ensures these properties. We are now ready to summarize the main result of this section while recalling the assumed upper bound on the message size:

Corollary 1. *For a given k-edge connectivity algorithm, let $V_{good} \subseteq V$ be the set of nodes for which there exist indistinguishable separated pairs (see Def. 2 on page 8). If each node sends at most $o(k)$ bits, then there exists a partition $\{A^*, B^*\}$ of W such that $|V_{good}| \geq \frac{3}{4}|V|$.*

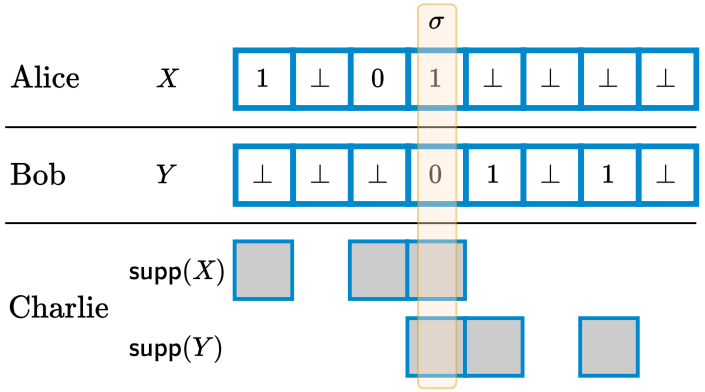


Fig. 2. An instance of UniqueOverlap$_{8,3}$. Since $(X_\sigma, Y_\sigma) = (1, 0)$, Charlie needs to answer "no".

4 The UniqueOverlap Problem

We now introduce a communication complexity problem in a 3-party model with simultaneous messages, whose hardness turns out to be crucial for obtaining a lower bound for deciding k-edge connectivity in the sketching model.

In the UniqueOverlap$_{m,s}$ problem, we are given two positive integers m and s, where $s \leq \lceil m/2 \rceil$, and there are three players Alice, Bob, and Charlie. We focus on one-way communication protocols, where Alice and Bob each send a single message to Charlie, who outputs the result. Alice's input is a ternary vector of length m, denoted by $X \in \{0, 1, \perp\}^m$, and, similarly, Bob gets a vector $Y \in \{0, 1, \perp\}^m$. We use $\mathrm{supp}(Z)$ to denote the support of a vector Z, i.e., the set of indices in $[m]$, for which $Z_i \neq \perp$. Charlie does not know anything about X and Y, except $\mathrm{supp}(X)$ and $\mathrm{supp}(Y)$.

Definition 3 (Valid Instance for UniqueOverlap$_{m,s}$)**.** The tuple (X, Y) is a *valid instance for* UniqueOverlap$_{m,s}$, if $|X| = |Y| = s$ and the following properties hold:

(P1) There is exactly one index $\sigma \in [m]$ such that $X_\sigma \oplus Y_\sigma = 1$, where $\oplus$ denotes the XOR operator.
(P2) For every $i \in [m] \setminus \{\sigma\}$, it holds that $X_i = \perp$ or $Y_i = \perp$.

In conjunction, Properties (P1) and (P2) ensure that $|\mathrm{supp}(X) \cap \mathrm{supp}(Y)| = 1$, and thus Charlie can deduce the value of σ directly from his input. To solve the problem, Alice and Bob each send a single message to Charlie, who must output "yes" if $X_\sigma = 0$ and $Y_\sigma = 1$, and "no" otherwise. See Figure 2 for an example.

4.1 Relationship to Other Problems in Communication Complexity.

In [3], the authors define the Leave-One Index problem (Problem 3 in [3]), where there are three players Alice, Bob, and Charlie. Alice and Bob hold the same string $x \in \{0, 1\}^N$ and Charlie knows an index $i^* \in [N]$. In addition, Alice has a set $S \subseteq [N]$ and Bob has a set $T \subseteq [N]$ with the property that $S \cup T = [N] \setminus \{i^*\}$, and the goal is that the players output x_{i^*}, whereby the order of communication is Alice, Bob, and then Charlie. We point out that Leave-One Index lacks an important requirement of the UniqueOverlap problem, namely that Charlie also knows the exact support sets (not just their size) of the sets S and T given to Alice and Bob, which is crucial for the correctness of our simulation. Thus, it is unclear whether there exists a reduction between these problems.

The uniqueness of the shared index σ may give the impression, at first, that UniqueOverlap is related to a (promise) variant of computing the set intersection. However, we doubt the existence

of a straightforward reduction, as, for the UniqueOverlap problem, the input vectors of Alice and Bob are ternary, and Charlie knows the (single) intersecting index as well as the support of the two input sets.

4.2 Hardness of the UniqueOverlap Problem

We now show that either Alice or Bob needs to send a message of size $\Omega(m)$ bits to Charlie in the worst case.

Theorem 2 (restated). *For any sufficiently large m, there exists an $s = \Theta(m)$ such that the deterministic one-way communication complexity of* $\mathsf{UniqueOverlap}_{m,s}$ *in the simultaneous 3-party model is $\Omega(m)$, where each m-length input vector has support s.*

Proof. Consider any protocol for $\mathsf{UniqueOverlap}_{m,s}$ and let C be the worst case length of any message sent. Assume towards a contradiction that

$$C \leq \left\lfloor \frac{m}{6} \right\rfloor - 1. \tag{9}$$

Suppose that we fix Alice's support to be a set I of $C + f$ indices from $[m]$, where f is an integer parameter described below. We can partitions her 2^{C+f} possible input vectors into 2^C blocks such that the i-th block contains all inputs on which Alice sends the number i to Charlie. Clearly, the largest block $\mathcal{B}$ in this partition contains at least 2^f inputs. Observe that there must exist a set $F_{\text{Alice}}(I) \subseteq I$ of f indices in I such that, for each $i \in F_{\text{Alice}}(I)$, there are inputs $X, \hat{X} \in \mathcal{B}$ such that $X_i \neq \hat{X}_i$, i.e., Alice sends the same message for two inputs in which the bit at index i is 0 and 1, respectively. We say that index i is *flipped for I*, and we refer to $F_{\text{Alice}}(I)$ as *Alice's flipped indices for support I*. Similarly, we can identify a set of flipped indices $F_{\text{Bob}}(I)$ for Bob by partitioning all possible input vectors with support I into blocks. Note that $F_{\text{Alice}}(I)$ and $F_{\text{Bob}}(I)$ need not be the same since Alice and Bob can execute different algorithms.

A crucial observation is that the algorithm fails, if there exist support sets I_1 and I_2 such that

(a) $|I_1 \cap I_2| = 1$, and
(b) $(I_1 \cap I_2) \subseteq (F_{\text{Alice}}(I_1) \cap F_{\text{Bob}}(I_2))$.

These conditions imply that the important index $\sigma \in I_1 \cap I_2$ is flipped for Alice's as well as Bob's support. In more detail, this means that there are two inputs X and $\hat{X}$ for Alice that are in the same block $\mathcal{B}_{\text{Alice}}$, where $X_\sigma \neq \hat{X}_\sigma$ and Alice sends the same message $\pi(\mathcal{B}_{\text{Alice}})$, and there are two inputs Y and $\hat{Y}$ in the same block $\mathcal{B}_{\text{Bob}}$, where $Y_\sigma \neq \hat{Y}_\sigma$, for which Bob sends $\pi(\mathcal{B}_{\text{Bob}})$, thus causing Charlie to fail in one of the (two) valid input combinations.

We show that if an algorithm manages to avoid (a) and (b) for all possible pairs of support sets, then it must send at least $C = \Omega(m)$ bits. Let $\mathcal{I} = \binom{[m]}{s}$ be the set family of all possible s-element support sets for the inputs of Alice and Bob. For each $i \in [m]$, define

$$\mathcal{R}_i^{\text{Alice}} = \{I \in \mathcal{I} \mid i \in F_{\text{Alice}}(I)\},$$

which means that $\mathcal{R}_i^{\text{Alice}}$ contains every support set I where i is a flipped index for I of Alice, and we define $\mathcal{R}_i^{\text{Bob}}$ analogously.

For the rest of the proof, if a statement holds for both $\mathcal{R}_i^{\text{Alice}}$ and $\mathcal{R}_i^{\text{Bob}}$, we simply omit the superscript and write $\mathcal{R}_i$ instead. Recall that each support $I \in \mathcal{I}$ contains at least f flipped indices, i.e., $|F_{\text{Alice}}(I)| \geq f$ and $|F_{\text{Bob}}(I)| \geq f$. This tells us that I must be a member of at least f of the $\mathcal{R}_i$ set families, and thus

$$\sum_{i=1}^{m} |\mathcal{R}_i^{\text{Alice}}| + |\mathcal{R}_i^{\text{Bob}}| \geq 2f\binom{m}{r}. \tag{10}$$

Consider $I_1 \in \mathcal{R}_i^{\text{Alice}}$ and $I_2 \in \mathcal{R}_i^{\text{Bob}}$. The only way to avoid (a) and (b) is if I_1 and I_2 are not valid choices for the support of Alice's input X and Bob's input Y, respectively (see Def. 3), which is the case if $|I_1 \cap I_2| \geq 2$; (recall that $I_1 \cap I_2 = \emptyset$ is impossible since $i \in I_1 \cap I_2$). For any $\mathcal{R}_i$, we can obtain a new set family $\hat{\mathcal{R}}_i$ by removing the index i from each set in $\mathcal{R}_i$, i.e.,

$$\hat{\mathcal{R}}_i = \left\{ I' \in \binom{[m]}{s-1} \mid \exists I \in \mathcal{R}_i \colon I = I' \cup \{i\} \right\}.$$

The correctness of the assumed protocol implies the following:

Fact 1. *Set families $\hat{\mathcal{R}}_i^{Alice}$ and $\hat{\mathcal{R}}_i^{Bob}$ form a cross-intersecting family, which means that*

$$\forall I_1' \in \hat{\mathcal{R}}_i^{Alice} \, \forall I_2' \in \hat{\mathcal{R}}_i^{Bob} \colon |I_1' \cap I_2'| \geq 1. \tag{11}$$

Next, we employ an upper bound on the sum of the cardinalities of any two cross-intersecting set families:

Lemma 6 ([9])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be such that each is a family of ℓ-element sets on the same underlying set of size p and assume that $\mathcal{A}$ and $\mathcal{B}$ are cross-intersecting. If $p \geq 2\ell$, then $|\mathcal{A}| + |\mathcal{B}| \leq 1 + \binom{p}{\ell} - \binom{p-\ell}{\ell}$.*

Choosing $f = \left\lfloor \frac{m}{3} \right\rfloor + 1$ and recalling (9), implies that

$$s = C + f \leq \frac{m}{2}, \tag{12}$$

and therefore $2(s-1) \leq m-1$. Thus, we can apply Lemma 6 with parameters $p = m-1$ and $\ell = s-1$ to obtain

$$|\hat{\mathcal{R}}_i^{\text{Alice}}| + |\hat{\mathcal{R}}_i^{\text{Bob}}| \leq 1 + \binom{m-1}{s-1} - \binom{m-s}{s-1}. \tag{13}$$

Since $|\hat{\mathcal{R}}_i| = |\mathcal{R}_i|$, it follows from (10) and (13) that

$$2f\binom{m}{s} \leq 1 + \binom{m-1}{s-1} - \binom{m-s}{s-1}. \tag{14}$$

However, it also holds that

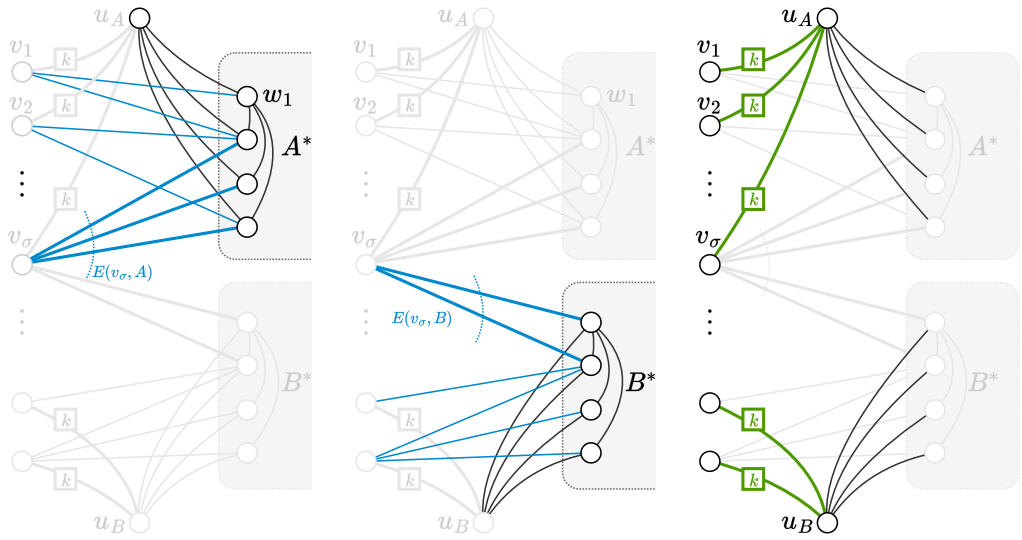
$$\begin{aligned} 2f\binom{m}{s} &= 2\left(\left\lfloor \frac{m}{3} \right\rfloor + 1\right) \frac{m}{s}\binom{m-1}{s-1} \\ \text{(by (12))} \quad &\geq 4\left(\left(\frac{m}{3} - 1\right) + 1\right)\binom{m-1}{s-1} \\ &= \frac{4}{3}m\binom{m-1}{s-1} \\ &> m\left(1 + \binom{m-1}{s-1}\right), \end{aligned}$$

which provides a contradiction to (14), thus completing the proof of Theorem 2. □

4.3 A Deterministic Algorithm for UniqueOverlap

The argument developed for our lower bound proof in Section 4.2 inspires the design of a simple deterministic algorithm that allows Alice and Bob to save a single bit by sending messages of length $s-1$ under certain conditions. As this is not needed for proving our main result, we relegate the details to Appendix B.

Theorem 3. *If $\lceil \frac{m}{3} \rceil < s$, then there exists a deterministic one-way protocol for $\mathsf{UniqueOverlap}_{m,s}$ in the simultaneous 3-party model, such that Alice and Bob send at most $s-1$ bits to Charlie.*



(a) Alice simulates all nodes in A^*.

(b) Bob simulates all nodes in B^*.

(c) Charlie simulates u_A and u_B, as well as all nodes in V, even though he only knows a subset of their incident edges.

Fig. 3. Overview of the Simulation. Each player simulates a certain subset of the nodes and their incident edges. Note that the edges of v_σ are "split" between Alice and Bob.

5 Simulation

In this section, we show how Alice, Bob, and Charlie can jointly simulate a given k-edge connectivity algorithm $\mathcal{A}^{k\text{-conn}}$ in the 3-party model to solve the UniqueOverlap$_{m,s}$ problem, for some given integers m and s, assuming that the sketches are sufficiently small for Corollary 1 to hold.

We now describe the details of the simulation and how the players can create the lower bound graph (see Sec. 2), given a valid instance of UniqueOverlap$_{m,s}$. First, each player locally computes

$$n = 2\lceil 4m/3 \rceil, \tag{15}$$

which will correspond to the number of nodes in the graph $G \in \mathcal{G}_{k,n}$ that the players construct. According to Corollary 1, there is a partitioning of W into $\{A^*, B^*\}$, as well as a set V_{good} of size at least m vertices, with the property that there exist indistinguishable separated pairs (see Def. 2), for each of the nodes in V_{good}. We use $V_{\text{good}}^{\leq m}$ to denote the first m nodes in V_{good}, ordered by IDs. Note that the players know, in advance, the IDs in the sets V, V_{good}, A^*, B^*, as these are fixed for all possible lower bound graphs in $\mathcal{G}_{k,n}$. Furthermore, they can also pre-compute the indistinguishable separated pair (S_0^i, S_1^i) , for every $v_i \in V_{\text{good}}^{\leq m}$, as these are fully determined by fixing k, n, and the algorithm at hand.

Alice's and Bob's Simulation: Alice simulates all nodes in A^*, whereas Bob is responsible for the nodes in B^*. See Figures 3a and 3b for an example. First, Alice creates a clique on A^* and connects u_A to each node in this clique. Then, for each $i \in \text{supp}(X)$ where $v_i \in V_{\text{good}}^{\leq m}$, Alice inspects the input X_i and adds edges incident to the nodes in A^* as follows:

- If $X_i = 0$, then she includes the edges between v_i and the nodes in $S_1^i \cap A^*$.

- On the other hand, if $X_i = 1$, then Alice adds all edges connecting v_i to $S_0^i \cap A^*$.

Bob adds a clique on B^* and, analogously to Alice, includes an edge between u_B and each node in B^*. Next, we describe how Bob creates the edges between B^* and node $v_j \in V_{\text{good}}^{\leq m}$, for each $j \in \text{supp}(Y)$:

- If $Y_j = 0$, then he adds edges between v_j and $S_0^j \cap B^*$.
- If $Y_j = 1$, he connects v_j to each node in $S_1^j \cap B^*$.

Finally, Alice and Bob locally execute algorithm $\mathcal{A}^{k\text{-conn}}$ to compute the sketches produced by their simulated nodes in A^* and B^*, respectively, which they send to Charlie.

Charlie's Simulation: Charlie is responsible for computing the sketches for u_A, u_B, and all nodes in V, and he will also simulate the referee. To this end, he creates edges between u_A and each node in A^*—recall that these edges were also created by Alice for simulating the nodes in A^*—and he also connects u_A to every $v_i \in V \setminus \text{supp}(Y)$ via k parallel edges. Note that $V \setminus \text{supp}(Y)$ contains every i such that $i \in \text{supp}(X)$ or $i \notin \text{supp}(X) \cup \text{supp}(Y)$, whereby the latter case is equivalent to $X_i = Y_i = \perp$. Charlie also includes k parallel edges connecting u_B to every v_j where $j \in \text{supp}(Y) \setminus \{\sigma\}$. Similarly, he adds edges between u_B and each node in B^*. See Figure 3c.

Next, we describe how Charlie simulates the nodes in V:

- First, consider the crucial node $v_i \in V_{\text{good}}^{\leq m}$, where $\sigma = i$: Since Charlie does not know whether Alice and Bob used S_0^i or S_1^i for creating the edges between their nodes in $A^* \cup B^*$ and v_σ, he cannot hope to faithfully recreate the entire neighborhood of v_i. However, what he can do instead is to directly compute the sketch that algorithm $\mathcal{A}^{k\text{-conn}}$ produces for node v_i, in the case that $\sigma = i$. To see why this is true, recall that S_0^i or S_1^i form an indistinguishable separated pair, and Charlie knows the block $\mathcal{B}_i$ in partition $\mathcal{P}_{i,\sigma}$ that contains S_0^σ as well as S_1^σ. This ensures that v_i sends the same sketch $\pi(\mathcal{B}_i)$ given either neighborhood.
- For any node $v_j \in V_{\text{good}}^{\leq m}$ ($j \neq \sigma$), Charlie proceeds analogously as in the case $j = \sigma$, with the only difference being that he no longer uses $\mathcal{P}_{j,\sigma}$ for identifying the common block that contains S_0^j and S_1^j (and deriving v_j's sketch). Instead, if v_j is A-restricted (i.e., $j \in \text{supp}(X) \setminus \text{supp}(Y)$), then he uses $\tilde{\mathcal{P}}_{j,A^*}$ to find the common block that contains S_0^j and S_1^j and uses the associated block in $\mathcal{P}_{j,A^*}$ (see Page 9) to derive v_j's sketch. Otherwise, v_j must be B-restricted ($j \in \text{supp}(Y) \setminus \text{supp}(X)$), prompting him to find the common block in $\tilde{\mathcal{P}}_{j,B^*}$, and compute v_j's sketch accordingly.
- Charlie can directly simulate the remaining nodes in V, i.e., every

$$v_r \in (V \setminus V_{\text{good}}^{\leq m}) \cup \left\{ v_i \in V_{\text{good}}^{\leq m} \;\middle|\; i \notin (\text{supp}(X) \cup \text{supp}(Y)) \right\},$$

  since the only edges added for v_r are k parallel edges to u_A.

Charlie simulates that the referee receives the sketches sent by the nodes in V, u_A, and u_B, as well as the messages that it received from Alice and Bob for the nodes in A^* and B^*, which is sufficient for invoking $\mathcal{A}^{k\text{-conn}}$ and obtaining the decision by the referee. Charlie answers "yes" if and only if the decision was that the graph was k-edge connected.

5.1 Correctness and Complexity Bounds of the Simulation

As made evident by the above description, our simulation constructs only certain graphs in $\mathcal{G}_{k,n}$. In particular, these are graphs where all edges in the cut $E(V, W)$ are determined by separating pairs. We start by formalizing this restriction:

Definition 4. Consider any graph $G \in \mathcal{G}_{k,n}$ and recall the existence of set V_{good}, the partition (A^*, B^*), and the separating pairs (S_0^i, S_1^i) for each $v_i \in V_{\text{good}}^{\leq m}$, guaranteed by applying Corollary 1 to algorithm $\mathcal{A}^{k\text{-conn}}$. We say that G is *compatible* with a valid instance (X, Y) of $\mathsf{UniqueOverlap}_{m,s}$ if $|V_{\text{good}}| \geq m$ and, for every $i \in [m]$:

(1) If $X_i = 0$, then $E(v_i, A^*) = A^* \cap S_1^i$.
(2) If $X_i = 1$, then $E(v_i, A^*) = A^* \cap S_0^i$.
(3) If $Y_i = 0$, then $E(v_i, B^*) = B^* \cap S_0^i$.
(4) If $Y_i = 1$, then $E(v_i, B^*) = B^* \cap S_1^i$.
(5) If $i \in \text{supp}(X)$ or $i \notin (\text{supp}(X) \cup \text{supp}(Y))$, then v_i has k edges to u_A.
(6) If $i \in \text{supp}(Y) \setminus \text{supp}(X)$, then v_i has k edges to u_B.

Lemma 7. *Let G be a graph that is compatible with an instance $I = (X, Y)$ of* UniqueOverlap. *It holds that G is k-edge connected if and only if the solution to I is "yes", i.e., $X_\sigma = 0$ and $Y_\sigma = 1$.*

Proof. Since G is a compatible graph, we know that $G \in \mathcal{G}_{k,n}$. Thus, it follows from Lemma 2 that G is k-edge connected if and only condition (C1) holds, i.e., $|E(v_\sigma, A^*)| \leq k - 1$ and $|E(v_\sigma, B^*)| \geq k$. Since (S_0^σ, S_1^σ) form a separated pair, we know from Definition 1 that this is satisfied for S_1^σ. According to Definition 4, we use S_1^σ to determine the neighbors of v_σ in $A^* \cup B^*$ if and only if $X_\sigma = 0$ and $Y_\sigma = 1$, which proves the correspondence between k-edge connectivity and the solution to UniqueOverlap. □

Equipped with Definition 4, we are ready to show that the sketches produced by the simulation indeed correspond to an actual execution of $\mathcal{A}^{k\text{-conn}}$ on a suitable graph in $\mathcal{G}_{k,n}$.

Lemma 8. *Suppose that algorithm $\mathcal{A}^{k\text{-conn}}$ has a maximum sketch length of $L = o(k)$ bits. Then, Charlie provides as input to the referee in the simulation the same set of sketches that the referee receives when executing algorithm $\mathcal{A}^{k\text{-conn}}$ on a graph compatible with the instance (X, Y).*

Proof. Since the players construct a graph $G \in \mathcal{G}_{k,n}$, where $n = 2\left\lceil \frac{4m}{3} \right\rceil$, it follows from (2) (on page 6) that $|V| \geq \left\lceil \frac{4m}{3} \right\rceil$. According to Corollary 1, this implies that the set V_{good} must have a size of $c \cdot m$ for some constant $c \geq 1$, which ensures that the set $V_{\text{good}}^{\leq m} \subseteq V_{\text{good}}$ of size m exists. Recall that all players can compute V_{good}, since this is fully determined by the algorithm at hand (i.e., $\mathcal{A}^{k\text{-conn}}$) and the parameters n, k, and L. We now argue that the simulation computes the correct sketch for every type of node in graph G:

- Nodes u_A and u_B: These nodes are only simulated by Charlie who knows $\text{supp}(X)$, $\text{supp}(Y)$, and thus also σ. By the description of the simulation, Charlie creates their incident edges in a way that satisfies Points 5 and 6 of Definition 4.
- Node $w_j \in A^* \cup B^*$: Without loss of generality, assume that $w_j \in A^*$; the case $w_j \in B^*$ is symmetric. In the execution on the compatible graph G, the edges in the cut $E(V, w_j)$ are fully determined by the separating pairs $(S_0^1, S_1^1), \ldots, (S_0^m, S_1^m)$, which corresponds to how Alice creates the edges incident to w_j in the simulation. Moreover, Alice will also add an edge $\{u_A, w_j\}$, which ensures the same sketch for w_j as in the actual execution.
- Node $v_i \in V_{\text{good}}^{\leq m}$: According to the simulation, Charlie will select one of the blocks $\mathcal{B}_\sigma, \tilde{\mathcal{B}}_A, \tilde{\mathcal{B}}_B$, where $\mathcal{B}_\sigma \in P_{i,\sigma}$, $\tilde{\mathcal{B}}_A \in \tilde{P}_{i,A^*}$ and $\tilde{\mathcal{B}}_B \in \tilde{P}_{i,B^*}$, with the property that each of the blocks contains (S_0^i, S_1^i). Recall that these blocks exist due to Corollary 1. Let $\mathcal{B}_A \in P_{i,A^*}$ be the associated block of $\tilde{\mathcal{B}}_A$, and $\mathcal{B}_B \in P_{i,B^*}$ be the associated block of $\tilde{\mathcal{B}}_B$, see Page 9. If $i = \sigma$, then he computes $\pi(\mathcal{B}_\sigma)$ where $\pi(\mathcal{B}_\sigma)$ corresponds to the message sent by v_i for the inputs in $\mathcal{B}_\sigma$; otherwise, if $i \in \text{supp}(X) \setminus \text{supp}(Y)$, then he computes $\pi(\mathcal{B}_A)$. Finally, in the case that $i \in \text{supp}(Y) \setminus \text{supp}(X)$, he computes $\pi(\mathcal{B}_B)$ instead. A crucial observation is that, even

though Charlie does not know whether Alice used $S_0^i \cap A^*$ or $S_1^i \cap A^*$ to create the edges in $E(v_i, A^*)$, this does not prevent him from computing the correct sketch for v_i, since S_0^i and S_1^i are guaranteed to be in the same block in each of the partitions; a similar argument applies to the edges in $E(v_i, B^*)$.

- Node $v_i \in V \setminus V_{\text{good}}^{\leq m}$: These nodes do not have any edges to $A^* \cup B^*$. Instead, Charlie only creates k parallel edges to u_A, which matches the neighborhood of v_i in the compatible graph G. □

Lemma 9. *Suppose there exists a deterministic k-edge connectivity algorithm, with the property that every node sends a sketch of length at most $L = o(k)$ bits, when executing on $\mathcal{G}_{k,n}$. Then,* $\mathsf{UniqueOverlap}_{m,s}$ *has a deterministic communication complexity of $o\left(k \cdot \sqrt{m}\right)$ bits.*

Proof. By Lemma 8, Charlie invokes the referee with the same set of sketches as in the execution of algorithm $\mathcal{A}^{k\text{-conn}}$ on the compatible graph $G \in \mathcal{G}_{k,n}$. Thus, correctness follows directly from Lemma 7.

Next, we show the claimed bound on the communication complexity of $\mathsf{UniqueOverlap}_{m,s}$. By (15) (see page 14), we know that the size n of G is chosen such that $n = \Theta(m)$. Since every node in $A \cup B$ sends a sketch of at most L bits in the simulation, and these are all the nodes simulated by Alice and Bob, sending these concatenated sketches to Charlie requires at most

$$O(|A \cup B| \cdot L) = O(|W| \cdot L) = O\left(L \cdot \sqrt{n}\right) = o\left(k \cdot \sqrt{m}\right) \tag{16}$$

bits, where we have used (3) in the second-last equality. □

6 Combining the Pieces

We now use the results from the previous sections to prove Theorem 1.

Theorem 1 (restated). *Every deterministic algorithm that decides k-edge connectivity on n-node graphs in the distributed sketching model has a worst case message length of $\Omega(k)$ bits, for any super-constant $k = k(n) \leq \gamma\sqrt{n}$, where $\gamma > 0$ is a suitable constant.*

Let L be the maximum length of a sketch. Suppose that every node in V sends at most $o(k)$ bits, and consider an instance of $\mathsf{UniqueOverlap}_{m,s}$. Lemma 9 tells us that we can solve an instance of $\mathsf{UniqueOverlap}_{m,s}$ if Alice and Bob send a message of $o\left(k \cdot \sqrt{m}\right)$ bits. Since $k \leq O\left(\sqrt{n}\right) = O\left(\sqrt{m}\right)$, the messages sent is thus $o(m)$ bits. We obtain a contradiction to Theorem 2 which states that any deterministic algorithm for $\mathsf{UniqueOverlap}_{m,s}$ must send $\Omega(m)$ bits in the worst case.

7 Conclusion and Open Problems

A problem left open by our work is the complexity of *randomized* sketching algorithms for deciding k-edge connectivity. As outlined in Section 1, the existing lower bound approach for graph connectivity does not appear to be amenable to a straightforward generalization. This suggests that either new technical ideas are needed for proving such a bound, or perhaps there is indeed a more efficient randomized algorithm:

Open Problem 1. What is the sketch length of deciding k-edge connectivity, if we allow a small probability of error?

Our result implies that any deterministic sketching algorithm must send sketches of near-linear (in k) bits in the worst case. While the AGM sketches [1] yield an upper bound of $O(k \log^3 n)$ bits, currently there is no non-trivial deterministic upper bound:

Open Problem 2. Is there a deterministic k-edge connectivity algorithm that matches the best known randomized upper bound of $O\left(k \log^3 n\right)$ bits on the sketch length?

Finally, we point out that our techniques do not shed light on the most fundamental open problem in this setting, namely (1-edge) connectivity:

Open Problem 3. Can we solve connectivity deterministically in the distributed sketching model if messages are of length $o(n)$?

Acknowledgments

Peter Robinson was supported in part by National Science Foundation (NSF) grant CCF-2402836 Collaborative Research: AF: Medium: The Communication Cost of Distributed Computation. Ming Ming Tan was supported in part by National Science Foundation (NSF) grant CCF-2348346 CRII: AF: The Impact of Knowledge on the Performance of Distributed Algorithms.

References

[1] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. 2012. Analyzing graph structure via linear measurements. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 459–467.

[2] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. 2012. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*. 5–14.

[3] Vikrant Ashvinkumar, Sepehr Assadi, Chengyuan Deng, Jie Gao, and Chen Wang. 2023. Evaluating Stability in Massive Social Networks: Efficient Streaming Algorithms for Structural Balance. In *APPROX/RANDOM*.

[4] Sepehr Assadi. 2022. Lower Bounds for Distributed Sketching. In *11th Workshop on Advances in Distributed Graph Algorithms (ADGA)*. https://adga-workshop.org/2022/assadi.pdf

[5] Sepehr Assadi, Gillat Kol, and Rotem Oshman. 2020. Lower Bounds for Distributed Sketching of Maximal Matchings and Maximal Independent Sets. In *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*. 79–88. https://doi.org/10.1145/3382734.3405732

[6] Florent Becker, Martin Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, and Ioan Todinca. 2011. Adding a referee to an interconnection network: What can (not) be computed in one round. In *2011 IEEE International Parallel & Distributed Processing Symposium*. IEEE, 508–514.

[7] Florent Becker, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. 2014. The Simultaneous Number-in-Hand Communication Model for Networks: Private Coins, Public Coins and Determinism. In *Structural Information and Communication Complexity - 21st International Colloquium, SIROCCO 2014, Takayama, Japan, July 23-25, 2014. Proceedings*. 83–95. https://doi.org/10.1007/978-3-319-09620-9_8

[8] Sudipto Guha, Andrew McGregor, and David Tench. 2015. Vertex and hyperedge connectivity in dynamic graph streams. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 241–247.

[9] Anthony JW Hilton and Eric C Milner. 1967. Some intersection theorems for systems of finite sets. *The Quarterly Journal of Mathematics* 18, 1 (1967), 369–384.

[10] Jacob Holm, Valerie King, Mikkel Thorup, Or Zamir, and Uri Zwick. 2019. Random k-out Subgraph Leaves only O(n/k) Inter-Component Edges. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*. 896–909. https://doi.org/10.1109/FOCS.2019.00058

[11] Stasys Jukna. 2001. *Extremal combinatorics: with applications in computer science*. Vol. 29. Springer.

[12] Tomasz Jurdzinski, Krzysztof Lorys, and Krzysztof Nowicki. 2018. Communication complexity in vertex partition whiteboard model. In *International Colloquium on Structural Information and Communication Complexity*. Springer, 264–279.

[13] Tomasz Jurdzinski and Krzysztof Nowicki. 2017. Brief announcement: on connectivity in the broadcast congested clique. In *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[14] Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P Woodruff, and Mobin Yahyazadeh. 2017. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. Ieee, 475–486.

[15] Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P Woodruff, and Mobin Yahyazadeh. 2017. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. *arXiv preprint arXiv:1704.00633* (2017).

[16] Mauricio Karchmer, Ran Raz, and Avi Wigderson. 1995. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity* 5 (1995), 191–204.

[17] Eyal Kushilevitz and Noam Nisan. 1997. *Communication Complexity.* Cambridge University Press.
[18] Jiří Matoušek and Jan Vondrák. 2001. The probabilistic method. *Lecture Notes, Department of Applied Mathematics, Charles University, Prague* (2001).
[19] Andrew McGregor. 2014. Graph stream algorithms: a survey. *ACM SIGMOD Record* 43, 1 (2014), 9–20.
[20] Pedro Montealegre and Ioan Todinca. 2016. Brief announcement: deterministic graph connectivity in the broadcast congested clique. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. 245–247.
[21] Jelani Nelson and Huacheng Yu. 2019. Optimal Lower Bounds for Distributed and Streaming Spanning Forest Computation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*. 1844–1860. https://doi.org/10.1137/1.9781611975482.111
[22] Shreyas Pai and Sriram V Pemmaraju. 2020. Connectivity Lower Bounds in Broadcast Congested Clique. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
[23] Mark S Pinsker. 1964. Information and information stability of random variables and processes. *Holden-Day* (1964).
[24] Peter Robinson. 2023. Distributed Sketching Lower Bounds for k-Edge Connected Spanning Subgraphs, BFS Trees, and LCL Problems. In *37th International Symposium on Distributed Computing (DISC 2023)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 32–1.
[25] Pachara Sawettamalya and Huacheng Yu. 2025. A (Very) Nearly Optimal Sketch for k-Edge Connectivity Certificates. *CoRR* abs/2510.16336 (2025). https://doi.org/10.48550/ARXIV.2510.16336 arXiv:2510.16336
[26] Huacheng Yu. 2021. Tight Distributed Sketching Lower Bound for Connectivity. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*. 1856–1873. https://doi.org/10.1137/1.9781611976465.111

Appendix

A Proof of Lemma 3

Lemma 3 (restated). *Consider a set family W, any small $\varepsilon > 0$, and any positive integer $d \le \frac{\varepsilon|W|}{2e^{2+4/\varepsilon}}$. There exists a set family $\mathcal{S} \subseteq \binom{W}{d}$ such that, for any two distinct $S_1, S_2 \in \mathcal{S}$, we have $|S_1 \cap S_2| \le \frac{\varepsilon d}{2}$, and $|\mathcal{S}| \ge e^{d-1}$.*

Proof. The proof is similar to Lemma 6 in [14]. Fix $N = \left\lfloor \sqrt{\left(\frac{\varepsilon|W|}{2ed}\right)^{\varepsilon d/2} - 1} \right\rfloor$. We choose $S_1, \ldots, S_N$ independently and uniformly at random from $\binom{W}{d}$. We first bound the expected size of the overlap of two given subsets. Consider two distinct indices $i, j \le N$, and define indicator random variables $X_1, \ldots, X_d$ such that $X_k = 1$ if and only if the k-th element of S_j intersects with S_i, which happens with probability $\frac{d}{|W|}$, and thus $\mathbf{E}\left[|S_i \cap S_j|\right] = \sum_{k=1}^{d} \mathbf{E}\left[X_k\right] = \frac{d^2}{|W|}$. While the X_k variables are not independent, they are negatively dependent, and thus we can use a Chernoff bound to show concentration. For $\delta = \frac{\varepsilon|W|}{2d} - 1$, we get

$$\begin{aligned}
\mathbf{Pr}\left[|S_i \cap S_j| \ge (1+\delta)\,\mathbf{E}\left[|S_i \cap S_j|\right]\right] &\le \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mathbf{E}\left[|S_i \cap S_j|\right]} \\
&\le \left(\frac{e^{\frac{\varepsilon|W|}{2d}}}{\left(\frac{\varepsilon|W|}{2d}\right)^{\frac{\varepsilon|W|}{2d}}}\right)^{d^2/|W|} \\
&= \left(\frac{\varepsilon|W|}{2ed}\right)^{-\varepsilon d/2}
\end{aligned}$$

It follows by a union bound over the $\binom{N}{2} \leq \left(\frac{\varepsilon|W|}{2ed}\right)^{\varepsilon d/2} - 1$ possible pairs of subsets that a set family $\mathcal{S}$ with the required bounded intersection property exists with nonzero probability. Note that

$$\begin{aligned}
|\mathcal{S}| = \exp(\log N) &\geq \exp\left(\frac{1}{2}\left(\log\left(\left(\frac{\varepsilon|W|}{2ed}\right)^{\varepsilon d/2} - 1\right) - 2\right)\right) \\
\text{(since } d \leq \tfrac{\varepsilon|W|}{2e^{2+4/\varepsilon}}\text{)} \quad &\geq \exp\left(\frac{1}{2}\left(\log\left(\left(e^{1+4/\varepsilon}\right)^{\varepsilon d/2} - 1\right) - 2\right)\right) \\
&\geq \exp\left(\frac{1}{2}\left(\log\left(\left(e^{4/\varepsilon}\right)^{\varepsilon d/2}\right) - 2\right)\right) \\
&\geq \exp\left(\frac{1}{2}\left(\log\left(e^{2d}\right) - 2\right)\right) \\
&\geq e^{d-1}.
\end{aligned}$$

□

B Proof of Theorem 3

Theorem 3 (restated). *If $\lceil \frac{m}{3} \rceil < s$, then there exists a deterministic one-way protocol for* $\mathsf{UniqueOverlap}_{m,s}$ *in the simultaneous* 3*-party model, such that Alice and Bob send at most $s-1$ bits to Charlie.*

Proof. Consider an input vector X with $\mathrm{supp}(X) = \{i_1, i_2, \ldots, i_s\}$ where $i_1 < i_2 < \ldots < i_s$. We can obtain a binary string of length $s = |\mathrm{supp}(X)|$ from X by simply removing all $\perp$ from the vector. In other words, the binary string of X is $X_{i_1}X_{i_2}\ldots X_{i_s}$. For the rest of the proof, we use X to refer to the input vector or its binary string representation depending on the context.

We define $\pi_{i_a}(X)$ to be the mapping of the bit string of X to a $(s-1)$-length bit string that we get by removing X_{i_a}. That is,

$$\pi_{i_a}(X_{i_1}X_{i_2}\ldots X_{i_a}\ldots X_{i_s}) = X_{i_1}X_{i_2}\ldots X_{i_{a-1}}X_{i_{a+1}}\ldots X_{i_s}.$$

Our algorithm determines the message sent by a party given the input vector X to be $\pi_{i_a}(X)$, whereby i_a is chosen in a way that guarantees that Charlie can compute the output correctly. Note that, given X and $\pi_{i_a}(X)$, the only bit values of X that Charlie can not deduce with certainty is the value at index i_a, i.e., X_{i_a}. Now, consider a valid input pair $\mathrm{supp}(X) = \{i_1, \ldots, i_s\}$ and $\mathrm{supp}(Y) = \{j_1, \ldots, j_s\}$, where $\sigma = \mathrm{supp}(X) \cap \mathrm{supp}(Y)$. Upon receiving the messages $\pi_{i_a}(X)$ and $\pi_{j_b}(Y)$ sent by Alice and Bob respectively, for some values i_a and j_b, Charlie can deduce X_σ and Y_σ only if at least one of i_a or j_b is not σ.

In the following, we describe a protocol on how to choose the value $i_a \in \mathrm{supp}(X)$ so that Charlie can compute correctly: We construct a partition of $\binom{[m]}{s}$ into blocks $\mathcal{R}_1, \ldots, \mathcal{R}_m$ such that for each $i \in [m]$:

(a) For all $I \in \mathcal{R}_i$, we have $i \in I$.
(b) For all I, J in $\mathcal{R}_i$, it holds that $|I \cap J| \geq 2$.[5]

To see why Properties (a) and (b) are sufficient, consider $I = \mathrm{supp}(X)$ and $J = \mathrm{supp}(Y)$ such that $I \cap J = \{\sigma\}$. Let i and j are such that $I \in R_i$ and $J \in R_j$. The message sent by Alice and Bob given input X and Y is thus $\pi_i(X)$ and $\pi_j(Y)$, respectively. By the properties of the partition, it follows that $i \neq j$. Hence, at least one of them is not equal to σ, which allow Charlie to deduce the correct value of X_σ and Y_σ.

[5]The reader may notice that the set families $\mathcal{R}_1, \ldots, \mathcal{R}_m$ are defined analogously as in the lower bound proof in Section 4.2.

To complete the proof, we give a concrete construction of the blocks $\mathcal{R}_1, \ldots, \mathcal{R}_m$: We partition $[m]$ into $\lceil \frac{m}{3} \rceil$ intervals, each of length 3, except possibly one interval, where the length is either 1 or 2. We define $\Phi(b)$ to be the successor of b in the interval containing b, following the cyclic order. For instance, if the interval is $[b, b+1, b+2]$, then

$$\Phi(b+i) = b + ((i+1) \bmod 3),$$

for $i = 0, 1, 2$. Since we are considering cyclic order, the intervals are referred to as cycles. Note that if a cycle has only one element, say a, we set $\Phi(a)$ to be an arbitrary element different from a.

Let $\mathcal{I} = \binom{[m]}{s}$. We define $\mathcal{R}_i$ to be the set that contains all I in $\mathcal{I}$ such that both i and $\Phi(i)$ are in I, i.e., $\mathcal{R}_i = \{I \in \mathcal{I} \mid \{i, \Phi(i)\} \subseteq I\}$. To ensure $\mathcal{R}_i$ and $\mathcal{R}_j$ are disjoint, we remove any I that occurs in $\mathcal{R}_i \cap \mathcal{R}_j$ from one of them. It is straightforward to verify that the resulting $\mathcal{R}_1, \ldots, \mathcal{R}_m$ indeed satisfy Properties (a) and (b).

It remains to show that $\bigcup_{i=1}^{m} \mathcal{R}_i = \mathcal{I}$. Consider $I \in \mathcal{I}$. Recall that there are s elements in I and $\lceil \frac{m}{3} \rceil$ cycles. Given that $s > \lceil \frac{m}{3} \rceil$ from the assumption, there must exists distinct elements a and b in I such that they are in the same cycle. Hence I is in some $\mathcal{R}_i$. □

Received December 2025; accepted February 2026