Distributed Sketching Lower Bounds for k-Edge Connected Spanning Subgraphs, BFS Trees, and LCL Problems

Peter Robinson
School of Computer & Cyber Sciences, Augusta University, GA, USA

Abstract

We investigate graph problems in the distributed sketching model, where each node sends a single message to a referee who computes the output. We define a class of graphs and give a framework for proving lower bounds for certain embeddable problems, which leads to several new results: First, we present a tight lower bound of $\Omega(n)$ bits for the message size of computing a breadth-first search (BFS) tree. For locally-checkable labeling (LCL) problems, we show that verifying whether a given vertex labeling forms a weak 2-coloring requires messages of $\Omega(n^{1/3}\log^{2/3} n)$ bits, and the same lower bound holds for verifying whether a subset of nodes forms a maximal independent set. We also prove that computing a k-edge connected spanning subgraph (k-ECSS) requires messages of size at least $\Omega\big(k\log^2(n/k)\big)$, which is tight up to a logarithmic factor. To show these results, we define a simultaneous multiparty (SMP) model of communication complexity, where the players obtain certain subgraphs as their input, and develop a generic embedding argument that allows us to prove lower bounds for the graph sketching model by using reductions from the SMP model. We point out that these results also extend to single-round algorithms in the broadcast congested clique.

We also (nearly) settle the space complexity of the k-ECSS problem in the streaming model by extending the work of Kapralov, Nelson, Pachoki, Wang, and Woodruff (FOCS 2017): We prove a communication complexity lower bound for a direct sum variant of the $\mathsf{UR}^{\subset}_k$ problem and show that this implies $\Omega(k\,n\log^2(n/k))$ bits of memory for computing a k-ECSS. This is known to be optimal up to a logarithmic factor.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed graph algorithm, graph sketching, streaming

Digital Object Identifier 10.4230/LIPIcs.DISC.2023.32

1 Introduction

Understanding the amount of communication that is required for solving fundamental graph problems has been at the forefront of research in distributed computing. In this work, we consider the *distributed graph sketching model* (SKETCH), introduced in [5]. In SKETCH there are n nodes and each node starts out knowing its neighborhood of the input graph. After observing its initial state and the shared randomness, each node sends a single message to the referee, who does not get any input and is responsible for computing the output by inspecting the received messages. As elaborated in [16, 3, 28], the distributed sketching model is equivalent to the *single-round broadcast congested clique* (BCC_1), where each node sends a single message of β bits, where β denotes the *link bandwidth*, and these messages are received by all nodes simultaneously at the end of the round. Consequently, the results of our work apply to both models.

We assume that the nodes are assigned unique IDs from the set $[n]$. In addition, we equip the nodes with some amount of initial knowledge of the input graph, namely, each node knows not only its own ID but also the IDs of all of its neighbors. This is known as the KT_1 *assumption*, which has turned out to be a key ingredient for achieving communication-efficiency in distributed algorithms (see [19, 14, 13, 4]). We point out that KT_1 knowledge presents a significant obstacle when proving lower bounds, due to fact that each edge is

part of the input of *both* of its endpoints. Consequently, we cannot independently modify the input of a player (i.e. node) without affecting other nodes. This is a crucial difference between the models assumed in our work and, for instance, the edge-partition models, where each player obtains a subset of edges as their input [25, 27].

At a first glance, it may seem that the non-interactive computation of one-round algorithms presents a severe handicap to solving *any* interesting problem with a distributed algorithm in this setting, despite the initial KT_1 knowledge. However, the breakthrough results of Ahn, and Guha, and McGregor [2] (see also the work of Kapron, King, and Mountjoy [18]) introduced a linear sketching technique that opened up the possibility of communication-efficient solutions in SKETCH and BCC_1 for several fundamental graph problems, including computing spanning trees and deciding graph connectivity, while requiring messages (also called "sketches") of only polylogarithmic length.

1.1 Our Contributions and Related Work

A Lower Bound Technique for the Distributed Graph Sketching Model. We present an embedding approach for proving lower bounds in the distributed sketching model (SKETCH) and, equivalently, in the single-round broadcast congested clique (BCC_1). This technique generalizes an approach that was pioneered by Nelson and Yu [22], who proved an $\Omega(\log^3 n)$ lower bound in this setting for computing a spanning forest. In a subsequent breakthrough, Yu [28] extended this work by showing that this is a tight lower bound even for the easier problem of graph connectivity.

Our approach differs from previous works by defining a *simultaneous multiparty (*SMP*) model* as an intermediate step, where some of the players may get an entire subgraph as their input rather than just the neighborhood of a single node. A technical challenge is that the inputs of different players overlap with each other, which rules out using simple product distributions for the lower bound. We specify a class of fairly generic lower bound graphs and introduce the notion of *embeddable problem*, which captures a broad range of intuitive properties, making it applicable to seemingly unrelated problems such as computing a k-edge connected spanning subgraph and verifying a weak 2-coloring. For embeddable problems that have unique outputs for a given input (e.g., decision problems), we obtain a reconstruction procedure that succeeds with sufficiently high probability in recovering the output, while omitting the transcript of some players. For general embeddable problems, which may not have uniquely determined outputs, we use Pinsker's inequality to argue that omitting the transcript of some players does not significantly skew the probability distribution of certain important cut sets. We point out that the reconstruction mechanism for unique output problems has a significantly improved error probability compared to using Pinsker's inequality as in [22, 28], which may be useful for other applications.

In more detail, we choose the class of lower bound graphs such that there is a large set of nodes V with the property that all nodes in V have neighborhoods that are "similar", i.e., are identically distributed. We show that, for solving an embeddable problem, the referee needs to obtain a sufficient amount of information about the neighborhood of one specific important node $v_\sigma \in V$. However, since the index σ is not given to the algorithm, the neighbors of the nodes in V do not know which one of their own neighbors is v_σ and consequently end up sending messages of large size to ensure a small probability of error. For instance, when computing a BFS tree, the node v_σ is chosen to be the only node in V for which all of its incident edges are part of any BFS tree. Due to the lack of knowledge of σ thus effectively requires the referee to learn about the neighborhoods of all nodes in V.[1]

[1] The author would like to thank the anonymous DISC 2023 reviewer for suggesting this intuition.

Computing a BFS Tree. Similarly to computing a spanning tree, computing a BFS tree has a small output size of $\Theta(n \log n)$ bits, and hence one might expect that the sketching technique of [2], which allows recovering an incident edge for each vertex, would lead to a solution using only sketches of length $O(\text{poly} \log n)$. We show that this intuition is misleading by presenting a tight bound of $\Omega(n)$ on the message size for computing a BFS tree in BCC_1 and SKETCH. This reveals a near-linear gap to the problem of computing a spanning tree, which requires only messages of $O(\log^3 n)$ bits. For the proof of this result, we only need to use the generic lower bound graph construction and do not require the full machinery of the embedding argument. With the right lower bound construction in place, the result readily follows from a reduction to the index problem in two-party communication.

▶ **Theorem 1.** *Any public coin constant-error randomized algorithm that computes a BFS tree rooted at a designated node of an n-node graph, requires a worst case message length of $\Omega(n)$ bits in the distributed sketching model* (SKETCH) *and the one-round broadcast congested clique* (BCC_1).

Verifying Symmetry Breaking Problems. We apply the embedding technique to locally-checkable labeling (LCL) problems [21], which have been studied extensively in the distributed computing literature and, roughly speaking, are graph problems that can be verified locally in the sense that each node only needs to check the consistency of the assigned labels in its $O(1)$-neighborhood. Here, we focus on verifying a weak 2-coloring, which is a vertex coloring of the graph with two colors, with the only restriction being that each non-isolated vertex has at least one neighbor with a different color. Since a weak 2-coloring can be computed from the output of other symmetry breaking problems, it comes as no surprise that more difficult LCL problems such as verifying a maximal independent set adhere to the same lower bound as weak 2-coloring. While the work of Assadi, Kol, and Oshman [3] shows a lower bound of $\Omega(n^{1/2-\epsilon})$ bits on the message size for *computing* an MIS in the distributed sketching model, it is unclear whether their result has any implications for the verification problem, due to the fundamentally different nature of computation and verification of symmetry breaking problems. We instantiate the embedding technique to prove the following result:

▶ **Theorem 2.** *Any $\frac{1}{25}$-error randomized algorithm that verifies if a labeling of a subset of vertices forms a weak 2-coloring of an n-node input graph, requires a worst case message length of $\Omega\left(n^{1/3} \log^{2/3} n\right)$ bits in* SKETCH *and* BCC_1. *The same bound holds for deciding whether a subset of nodes forms a maximal independent set.*

Computing a k-Edge Connected Spanning Subgraph. By applying the embedding technique, we obtain the first lower bounds for computing a k-edge connected spanning subgraph. Prior to our work, the only known lower bound for this problem was the one for spanning tree construction (i.e., $\Omega(\log^3 n)$ bits, see [22]), which does not scale with k. In particular, for $k = O(\log n)$, the lower bound of [22] for computing a spanning forest immediately implies an $\Omega(\log^3 n)$ lower bound, since the referee can recover a spanning tree from a k-ECSS.

▶ **Theorem 3.** *Any public coin randomized algorithm that computes a k-edge connected spanning subgraph of an n-node graph in* SKETCH *or* BCC_1 *with probability at least $1 - o(1)$, has a worst case message length of $\Omega\left(k \log^2 \frac{n}{k}\right)$ bits, for any $k = o\left(\frac{n^{1/4}}{\log^{1/2} n}\right)$.*

We point out that Theorem 3 is tight up to a logarithmic factor, since the algorithm for deciding k-edge connectivity of Ahn, Guha, and McGregor [2] also computes a "witness", i.e., a k-edge connected subgraph. It is straightforward to implement their technique in SKETCH using messages of $O(k \log^3 n)$ bits.

In Section 6, we consider the k-ECSS problem in the dynamic data streaming setting where the graph is represented as a stream of edge arrivals and departures. To prove a lower bound, we introduce a new communication complexity problem called ℓ-fold $\mathsf{UR}_k^{\subset}$, which essentially consists of ℓ instances of the $\mathsf{UR}_k^{\subset}$ problem, defined by Kapralov, Nelson, Pachoki, Wang, and Woodruff [17]. In the $\mathsf{UR}_k^{\subset}$ problem, there are two players, Alice and Bob. Alice gets a set S, whereas Bob gets a proper subset $T \subset S$. After Alice sends a single message to Bob, he must output k elements in $S \setminus T$. It was shown in [17] that the $\mathsf{UR}_k^{\subset}$ problem requires $\Omega\big(k \log^2 \frac{n}{k}\big)$ bits in the one-way two-party model. The ℓ-fold $\mathsf{UR}_k^{\subset}$ problem is a direct-sum variant of the $\mathsf{UR}_k^{\subset}$ problem and, by a simple extension of the lower bound technique of [17], we prove that ℓ-fold $\mathsf{UR}_k^{\subset}$ requires $\Omega(k\,\ell \log^2 \frac{n}{k})$ bits. This in turn gives rise to a lower bound on the required memory:

▶ **Theorem 4.** *Any Monte Carlo data structure for computing a k-edge connected spanning subgraph of an n-node graph requires $\Omega\big(k\,n \log^2 \frac{n}{k}\big)$ space in the one-pass fully dynamic turnstile model.*

1.2 Additional Related Work

Closely related to k-ECSS is the problem of computing a spanning forest of the input graph in the distributed sketching model. As mentioned above, Nelson and Yu [22] prove a lower bound of $\Omega(\log^3 n)$ bits and this is known to be optimal due to the graph sketching approach of [2], which relies on access to shared randomness. Holm, King, Thorup, Zamir, and Zwick [15] show that a spanning tree can be computed with a message length of $O(\sqrt{n} \log n)$ bits, *without* access to shared randomness. Currently, there are no lower bounds known for the distributed sketching model if nodes only have access to private random bits.

While our results only apply to single-round algorithms in the BCC_1 model, several other works have studied multi-round lower bounds in this setting: Drucker, Kuhn, and Oshman [10] show round lower bounds for subgraph detection problems, whereas Chen and Grossman [7] prove a lower bound for the directed planted clique problem. Pai and Pemmaraju [23] give round complexity lower bounds depending on the per-round bandwidth for graph connectivity and finding connected components in BCC_1. The work of [12] considers so called hybrid models resulting from combining the broadcast congested clique with other distributed computing models.

Several other works show lower bounds for one-round algorithms in the related CONGEST model [24], which differs from the congested clique by assuming that the input graph corresponds to the actual communication network. Fischer, Gonen, Kuhn, and Oshman [11] show that one-round randomized algorithms for triangle detection require nodes to send messages of at least $\Omega(\Delta)$ bits, where Δ is the maximum degree of the graph. Previously, Abboud, Censor-Hillel, Khoury, and Lenzen [1] showed that a slightly stronger bound of $\Omega(\Delta \log n)$ bits for deterministic algorithms based on their novel fooling views framework. We point out that the proof of [1] assumes that all three nodes must detect that they are part of a triangle (if one exists), rather than just at least node as in [11]. A related question is the minimum link bandwidth necessary for obtaining a solution in a certain number of rounds, which is also called bandwidth complexity in [6].

2 A Lower Bound Technique for Embeddable Problems

In this section, we present a generic technique for showing lower bounds for problems that satisfy certain "embeddability" properties. We first define a general class of graphs that we will use for all our lower bounds in Sections 3, 4 and 5, albeit with somewhat different

parameters. On these graphs, we define the simultaneous multiparty (SMP) model, and show that embeddable problems have specific properties that enable us to compute a solution while omitting the messages of some player.

2.1 The Lower Bound Graph $\mathcal{G}_\ell$

For a positive integer parameter ℓ, we define a class of graphs $\mathcal{G}_\ell$ that contains all graphs G defined as follows. The vertices of G consist of sets U, V, and W, whereby $|V| = \ell$, and we further partition U into vertex sets $U_1, \dots, U_\ell$. Each v_i is connected to a subset of the vertices in U_i and W. We use E_i to denote the edges in the cut $E(v_i, W)$. Figure 1 on page 19 shows the general structure of the graphs in $\mathcal{G}_\ell$. We will fix the precise cardinalities of U and W as well as the edges $E(U, V)$ and $E(V, W)$ when we introduce the specific input distributions in the subsequent sections. In the problems that we consider, the output will depend on the neighborhood of a particular vertex v_σ, where $\sigma \in [\ell]$ is called the *embedding index*.

We give each vertex a unique integer as its *ID*. In addition to an ID, we also assume that each vertex in W has a *label*. For instance, in the context of verifying a weak 2-coloring, a label of a vertex indicates its color. For k-edge connected spanning subgraphs, on the other hand, we simply omit the labels. The crucial difference between IDs and vertex labels will become apparent when considering the SKETCH model: Every node knows only its own label, but knows the IDs of all nodes in its neighborhood.

2.2 The Simultaneous Multiparty (SMP) Model

In our lower bound constructions, we use the following simultaneous multiparty model as an intermediate step: There are $\ell + 2$ players $\text{Alice}_1, \dots, \text{Alice}_\ell$, Bob, and Charlie. When revealing the *neighborhood of a vertex* u to a specific player, the player learns the ID and the label of u, as well as the IDs of all of u's neighbors in G. The inputs of the players are defined as follows; see Figure 2 on page 19: For each $v_i \in V$, player Alice_i knows the neighborhood of vertex v_i, whereas Bob knows the neighborhoods of all vertices in W. In other words, Bob knows the entire cut $E(V, W)$, including the labels of W. Charlie gets as input the neighborhoods of all nodes in U, the index σ, and the IDs and labels of the nodes in W.

$\text{Alice}_1, \dots, \text{Alice}_\ell$ and Bob each send a single message to Charlie who must output the solution. Apart from these messages there is no other communication between the players. However, we assume that they have access to an infinite string R of random bits when considering randomized algorithms.

Random Variables and Notation

Let Π_i denote the message sent by Alice_i and let Π_B denote Bob's message. We use random variable C to denote Charlie's output. By a slight abuse of notation, we assume that U and W also denote the IDs of the vertex sets U and W, respectively. Furthermore, we use $\mathcal{L}_W$ to denote the labels of the nodes in W. We define the abbreviation $\Pi_{(\leqslant j)} := (\Pi_1, \dots, \Pi_j)$ and define $\Pi_{\geqslant j}$ analogously. Observe that Charlie computes C based on his initial knowledge, the received messages Π_B, $\Pi_{(\leqslant \ell)}$, and the shared randomness R, i.e., $C := C(R, U, W, E(U, V), \mathcal{L}_W, \Pi_{(\leqslant \ell)}, \Pi_B, \sigma)$. To shorten the notation, we define

$$Z := (U, W, E(U, V), \mathcal{L}_W),$$

and point out that Charlie's input is Z and σ. We use the indicator random variable $\mathbf{1}_{\mathsf{Succ}}$ for the event that the protocol succeeds.

Throughout this section, we make use of basic notions from information theory. We refer the reader to Appendix A for the formal definition of these quantities and pointers to further references. For random variables X, Y, and Z, we use $\mathbf{H}[X \mid Y]$ to denote the *conditional entropy* of X conditioned on Y, which, intuitively speaking, captures the expected remaining uncertainty of X's value after revealing Y. We use $\mathbf{I}[X : Y \mid Z]$ for the *conditional mutual information* between X and Y conditioned on Z, which is the expected amount of information X reveals about Y (and vice versa) after revealing Z.

2.3 Embeddable Problems

We say that a problem P is *embeddable* if there is an input distribution $\mathcal{D}$ on graphs in $\mathcal{G}_\ell$ that satisfies the following two properties:

(P1) **Independence of the embedding index σ**: Random variable σ is sampled uniformly from $[\ell]$, and is independent of the edges, labels, and vertex IDs.

(P2) **Independence of cut sets under conditioning**: Random variables $E_1, \dots, E_\ell$ are mutually independent conditioned on Z.

Intuitively speaking, Property (P1) guarantees that the specific value of the index σ does not bias the distribution of the transcripts of the players $\text{Alice}_1, \dots, \text{Alice}_\ell$, and Bob. Property (P2) ensures that knowing some of the cut sets does not leak information about the remaining cut sets, in particular E_σ. As we will see in Lemma 6 below, Properties (P1) and (P2) are sufficient for obtaining a probability distribution on E_σ that is close to the one that Charlies has access to when computing his output, even though it does not take into account Bob's transcript. For problems that also satisfy the following property (P3), which avoids dependencies between Charlie's output and parts of the graph that are unrelated to E_σ, we give a bound on the probability of directly reconstructing Charlie's output in Lemma 5 below. Note that (P3) is a natural property of decision problems, where E_σ and the labels of its neighbors fully determines the output of the algorithm.

(P3) **Unique Output**: Conditioned on Charlie's input Z, σ, the cut set E_σ, the shared randomness R, and the event that Charlie's output C correctly solves problem P, it holds that C is a deterministic function of E_σ, i.e., $\mathbf{H}[C \mid E_\sigma, R, Z, \sigma, \mathbf{1}_{\mathsf{Succ}} = 1] = 0$.

In the next lemma, we formalize a crucial property of embeddable problems: We can compute a solution with sufficiently large probability just by inspecting Charlie's input and the transcripts of $\text{Alice}_1, \dots, \text{Alice}_\ell$.

▶ **Lemma 5** (Existence of Reconstruction Protocol). *Consider an embeddable problem* P *with input distribution* $\mathcal{D}$ *on* $\mathcal{G}_\ell$ *that satisfies (P1), (P2), and (P3). Suppose that there is a public coin randomized* SMP *protocol that solves* P *with error* $\delta \leqslant \min\left\{\frac{1}{2}, \frac{1}{|C|^2}\right\}$. *Then there exists a reconstruction protocol* $\mathcal{R}(R, Z, \sigma, \Pi_{(\leqslant \ell)})$ *that returns Charlie's output* C *with probability at least* $2^{-\left(\frac{|\Pi_B|}{\ell} + 3\sqrt{\delta}\right)}$.

To gain some intuition for applying Lemma 5, suppose that Charlie just outputs a single bit, i.e., $|C| \leqslant 1$ and that $\delta \leqslant \frac{1}{25}$, which means that $\sqrt{\delta} \leqslant \frac{1}{5}$. Now assume that Bob sends a message of at most $\ell/5 \leqslant \sqrt{\delta}\ell$ bits, which means that, on average, his message can reveal only a $(\frac{1}{5})$-fraction of a bit of information for each of the ℓ cut sets E_i between V and

W. Then, Lemma 5 tells us that we can recover Charlie's output with probability at least $\frac{1}{2^{1/5+3/5}} \approx 0.57$ without Bob's message Π_B. Moreover, if we consider protocols that succeed with high probability, i.e., $\delta \leqslant \frac{1}{\ell}$, and restrict the length of Bob's message to at most $\sqrt{\ell}$ bits, we get a recovery protocol that succeeds with probability at least $\frac{1}{2^{(4/\sqrt{\ell})}} = 1 - o(1)$.

For random variables X and Y, consider the probability distributions $\mu(X)$ and $\mu(X \mid Y{=}y)$. We define $|\mu(X) - \mu(X \mid Y{=}y)|_{TV}$ to be the *total variation distance*, which is the maximum difference in the probability of any event $\mathcal{E}$ on X for these two distributions. We use parts of the techniques developed in the proof of Lemma 5 to show the following result:[2]

▶ **Lemma 6.** *Consider an algorithm for an embeddable problem that satisfies (P1) and (P2). Then, it holds that*

$$\mathbf{E}\big[\big|\mu(E_\sigma \mid Z, \sigma, \Pi_{\leqslant \ell}) - \mu(E_\sigma \mid Z, \sigma, \Pi_{\leqslant \ell}, \Pi_B)\big|_{TV}\big] \leqslant 2\sqrt{|\Pi_B|/\ell},$$

where the expectation is taken over Z, σ, $\Pi_{\leqslant \ell}$, and Π_B.

Note that, strictly speaking, Lemma 6 does not give a concrete reconstruction protocol, but instead only an upper bound on the statistical distance between the distribution of E_σ, conditioned on Charlie's input and $\Pi_{\leqslant \ell}$, and the distribution of E_σ where we also condition on Π_B. However, this turns out to be sufficient for obtaining a concrete reconstruction protocol, as we demonstrate in Section 5.

2.4 Proof of Lemma 5

High-Level Overview. Recalling that our goal is to obtain a protocol that recovers the output C without seeing Bob's message Π_B, we start by deriving an upper bound on how much information his message may contain about C. We show that this is roughly equivalent to the amount of information that Π_B conveys about the cut set E_σ (see Lemma 7). In particular, since Bob does not know σ, the amount of information that Π_B contains about E_σ is only a $\frac{|\Pi_B|}{\ell}$-fraction on average (see Lemma 8). In other words, if Bob's message is short compared to the number of cut sets ℓ, then it cannot convey a significant amount of information about E_σ. In Lemma 9, we combine these observations to show that we can guess Charlie's output with a probability of at least $2^{-\frac{|\Pi|}{\ell}}$, where we have omitted some error terms that depend on the success probability of the original protocol.

We now give the detailed argument. Observe that $\Pi_{(\leqslant \ell)}$, Π_B, R, Z, and σ fully determine C, and thus

$$\mathbf{I}\big[C : \Pi_{(\leqslant \ell)}, \Pi_B \;\big|\; R, Z, \sigma\big] = \mathbf{H}[C \mid R, Z, \sigma]. \tag{1}$$

Therefore, by the chain rule, we have that

$$\begin{aligned}\mathbf{I}\big[C : \Pi_{(\leqslant \ell)} \;\big|\; R, Z, \sigma\big] &= \mathbf{I}\big[C : \Pi_{(\leqslant \ell)}, \Pi_B \;\big|\; R, Z, \sigma\big] - \mathbf{I}\big[C : \Pi_B \;\big|\; R, Z, \Pi_{(\leqslant \ell)}, \sigma\big] \\ &\quad \text{(by (1))} \quad = \mathbf{H}[C \mid R, Z, \sigma] - \mathbf{I}\big[C : \Pi_B \;\big|\; R, Z, \Pi_{(\leqslant \ell)}, \sigma\big]. \end{aligned} \tag{2}$$

The next lemma shows that we can upper-bound the amount of information that Bob's transcript reveals about Charlie's output in terms of the information that the transcript reveals about the cut set E_σ, assuming that the protocol succeeds.

▶ **Lemma 7.** $\mathbf{I}\big[C : \Pi_B \;\big|\; R, Z, \Pi_{(\leqslant \ell)}, \sigma\big] \leqslant \mathbf{I}\big[E_\sigma : \Pi_B \;\big|\; R, Z, \Pi_{(\leqslant \ell)}, \sigma\big] + 3\sqrt{\delta}.$

[2] Omitted proofs are presented in the full version of the paper.

Next, we show that Bob's message reveals the same amount of information about any cut set (on average), which holds for E_σ in particular.

▶ **Lemma 8.** $\mathbf{I}\big[E_\sigma : \Pi_B \;\big|\; R, Z, \Pi_{(\leqslant \ell)}, \sigma\big] \leqslant \frac{|\Pi_B|}{\ell}$ *and also* $\mathbf{I}\big[E_\sigma : \Pi_B \;\big|\; Z, \Pi_{(\leqslant \ell)}, \sigma\big] \leqslant \frac{|\Pi_B|}{\ell}$.

Plugging the bound of Lemma 8 into Lemma 7, we obtain

$$\mathbf{I}\big[C : \Pi_B \;\big|\; R, Z, \Pi_{(\leqslant \ell)}, \sigma\big] \leqslant \frac{|\Pi_B|}{\ell} + 3\sqrt{\delta}.$$

Returning to (2), we get

$$\mathbf{I}\big[C : \Pi_{(\leqslant \ell)} \;\big|\; R, Z, \sigma\big] \geqslant \mathbf{H}[C \mid R, Z, \sigma] - \frac{|\Pi_B|}{\ell} - 3\sqrt{\delta}. \tag{3}$$

Intuitively speaking, (3) says that the transcript of $\mathrm{Alice}_1, \ldots, \mathrm{Alice}_\ell$ reveal all except a fraction of a bit of the information contained in Charlie's output C, in expectation. However, we cannot directly use the assumed SMP protocol $\mathcal{P}$ for reconstructing Charlie's output, because $\mathcal{P}$ is only guaranteed to work given the transcripts of *all* players. Nevertheless, the next lemma shows that there exists a simple reconstruction protocol, which completes the proof of Lemma 5.

▶ **Lemma 9.** *There exists a protocol $\mathcal{R}$ that takes R, Z, $\Pi_{(\leqslant \ell)}$, and σ as input, and correctly computes Charlie's output C with probability at least $2^{-\left(\frac{|\Pi_B|}{\ell} + 3\sqrt{\delta}\right)}$.*

Proof. Protocol $\mathcal{R}$ works as follows: Given the input $\{\Pi_1 = \pi_1, \ldots, \Pi_\ell = \pi_\ell, R = r, Z = z, \sigma = i\}$, it returns the output of Charlie that maximizes the probability, which is

$$\arg\max_c \mathbf{Pr}[C = c \mid \pi_1, \ldots, \pi_\ell, r, z, i].$$

Let $Y = (\Pi_{(\leqslant \ell)}, R, Z, \sigma)$, and let $p_{c|y} = \mathbf{Pr}[C = c \mid Y = y]$. We observe that

$$\begin{aligned}
\mathbf{Pr}[\mathcal{R} \text{ succeeds}] &= \sum_y \mathbf{Pr}[Y = y] \max_c p_{c|y} \\
&= \mathop{\mathbf{E}}_y\big[2^{\log \max_c p_{c|y}}\big] \\
\text{(by Jensen's inequality)} \quad &\geqslant 2^{\mathbf{E}[\log \max_c p_{c|y}]} \\
&= 2^{\mathbf{E}\left[\log\left(\max_c(p_{c|y}) \cdot \sum_c p_{c|y}\right)\right]} \\
&\geqslant 2^{\mathbf{E}\left[\log\left(\sum_c p_{c|y}^2\right)\right]}.
\end{aligned} \tag{4}$$

For a fixed y, we define the random variable $P_y(c) := p_{c|y}$, which is a function of c. In this notation, the exponent on the right-hand side becomes

$$\begin{aligned}
\mathop{\mathbf{E}}_y\left[\log\left(\sum_c p_{c|y} P_y(c)\right)\right] &= \mathop{\mathbf{E}}_y\Big[\log\Big(\mathop{\mathbf{E}}_c[P_y(c)]\Big)\Big] \\
\text{(by Jensen's inequality)} \quad &\geqslant \mathop{\mathbf{E}}_y\Big[\mathop{\mathbf{E}}_c[\log P_y(c)]\Big] \\
&= -\mathop{\mathbf{E}}_y[\mathbf{H}[C \mid Y = y]] \\
&= -\mathbf{H}[C \mid Y]
\end{aligned}$$

Returning to (4), we get

$$\mathbf{Pr}[\mathcal{R} \text{ succeeds}] \geqslant 2^{-\mathbf{H}[C \mid Y]} = 2^{-\mathbf{H}\left[C \mid \Pi_{(\leqslant \ell)}, R, Z, \sigma\right]}.$$

Since $\mathbf{H}[C \mid \Pi_{(\leqslant \ell)}, R, Z, \sigma] = \mathbf{H}[C \mid R, Z, \sigma] - \mathbf{I}\left[C : \Pi_{(\leqslant \ell)} \mid R, Z, \sigma\right]$, it follows that

$$\begin{aligned}\mathbf{Pr}[\mathcal{R} \text{ succeeds}] &\geqslant 2^{-\left(\mathbf{H}[C \mid R,Z,\sigma] - \mathbf{I}\left[C : \Pi_{(\leqslant \ell)} \mid R,Z,\sigma\right]\right)} \\ \text{(by (3))} \quad &\geqslant 2^{-\left(\frac{|\Pi_B|}{\ell} + 3\sqrt{\delta}\right)}.\end{aligned}$$

◀

3 A Lower Bound for Computing a BFS Tree

As a warm-up, we instantiate the generic class of lower bound graphs defined in Section 2.1 to show a tight bound on the message length for computing a breadth-first search (BFS) tree, where a fixed node s starts out knowing that it is designated as the source and the goal for the referee is to output a BFS tree rooted at s.

▶ **Theorem 1** (restated). *Any public coin constant-error randomized algorithm that computes a BFS tree rooted at a designated node of an n-node graph, requires a worst case message length of $\Omega(n)$ bits in the distributed sketching model* (SKETCH) *and the one-round broadcast congested clique* (BCC_1).

Proof. We are able to obtain this lower bound via a direct reduction from the Index_N problem in the two-party one-way setting, where there are two players, Diane and Edward. Diane starts with a binary vector $\mathbf{x}$ of length N and Edward gets an index $i \in [N]$. Diane can send a single message to Edward who must output the i-th bit of $\mathbf{x}$.

As discussed in Section 1, the models SKETCH and BCC_1 are equivalent and we will focus on the former out of convenience. Suppose that there is a SKETCH algorithm $\mathcal{A}$ that computes a BFS tree rooted at any given source node. We describe how Diane and Edward can simulate $\mathcal{A}$ to solve the Index_{ℓ^2} problem. Based on the lower bound graph class $\mathcal{G}_\ell$ that we described in Section 2.1, they sample a graph as follows: All the IDs of the nodes are fixed and the cardinalities of the vertex sets are defined as $|U| = |V| = |W| = \ell$. Moreover, there is a fixed perfect matching between U and V known to both players, i.e., we have edges $(u_1, v_1), \ldots, (u_\ell, v_\ell)$. Assume that Diane gets input $\mathbf{x}$, which is a binary vector of length ℓ^2. Diane interprets her input $\mathbf{x}$ as the characteristic vector of the ℓ^2 possible edges between the sets V and W, for the fixed ordering ρ of $V \times W$ where $\rho = ((v_1, w_1), \ldots, (v_1, w_\ell), (v_2, w_1), \ldots, (v_2, w_\ell), \ldots, (v_\ell, w_\ell))$. That is, Diane adds the i-th edge of ρ to the graph if and only if $\mathbf{x}_i = 1$. As a result, Diane knows the neighborhoods of all nodes in $V \cup W$. She simulates $\mathcal{A}$ on each one of them and sends the resulting messages to Edward.

Edward gets as input some index $i \in [\ell^2]$. Since he knows the ordering ρ, he computes the index $\sigma \in [\ell]$ such that $v_\sigma \in V$ is incident to the i-th (potential) edge in ρ, and adds the edge (s, u_σ). Figure 3 in the attached full paper shows the resulting graph. Then, he simulates $\mathcal{A}$ on s and each vertex in U, whereby s is designated as the source node of the tree. Upon receiving Diane's message, he simulates the referee and obtains the BFS tree assuming that $\mathcal{A}$ succeeded. If the i-th edge is included in the BFS-edges leading from v_σ to W, he outputs 1, otherwise he answers 0. Correctness follows since the BFS tree rooted at s must contain all the edges in the cut (v_σ, W).

It was shown in [20] that the Index_{ℓ^2} problem requires $\Omega(\ell^2)$ bits in the one-way two-party model, for achieving constant probability of success. Therefore, Diane's simulation produces a message of length $\Omega(\ell^2)$ bits, and thus one of the 2ℓ vertices simulated by her must have sent a message of size $\Omega(\ell)$ bits. The result follows since the lower bound graph has $n = 3\ell + 1$ vertices in total. ◀

4 A Lower Bound for Verifying Symmetry Breaking Problems

We now turn our attention to the problem of verifying whether a given labeling of the vertices is a weak 2-coloring of the input graph, which means that each non-isolated vertex has at least one differently-colored neighbor.[3]

High-level Overview. As we plan to employ Lemma 5, we start by defining the EI_m problem in the SMP model and show that it is an embeddable problem satisfying Properties (P1), (P2), and (P3) with a suitable input distribution on the graphs in $\mathcal{G}_\ell$. Next, we show how to simulate a given EI_m algorithm in the one-way two-party communication complexity model for solving set disjointness. From this, we derive a lower bound on the length of Alice_σ's message. We obtain the sought lower bound by showing that a protocol for 2-weak coloring can be used to solve the EI_m problem in the SMP model.

4.1 The Edge Intersection Problem EI_m

Here, in addition to vertex IDs, we assume that each vertex in W is labeled with a bit indicating its color, and we define $W_b \subseteq W$ to be the b-labeled vertices, for $b \in \{0,1\}$. As defined in Section 2, random variable $\mathcal{L}_W$ represents the label assignment for nodes in W. We consider the simultaneous multiparty (SMP) model with the input assignments as described in Section 2.2, i.e., Alice_i knows the neighborhood of v_i, Bob knows all nodes (and labels) in W as well as their neighbors, and Charlie knows the IDs of U, W, the labels $\mathcal{L}_W$, and the embedding index σ. Charlie receives a message from $\text{Alice}_1, \dots, \text{Alice}_\ell$ and Bob, and then computes his answer. The goal is to determine whether an edge in E_σ "intersects" with (i.e., has an endpoint in) the 1-labeled nodes in W. Thus, to correctly solve the EI_m problem, it must hold for Charlie's output that

$$C = \begin{cases} \text{"yes"} & \text{if } E_\sigma \cap W_1 \neq \emptyset; \\ \text{"no"} & \text{otherwise.} \end{cases} \tag{5}$$

4.2 The Hard Input Distribution $\mathcal{D}_{\mathsf{EI}_m}$

Let $\ell = \lceil m^3 \log m \rceil$. We define the following distribution $\mathcal{D}_{\mathsf{EI}_m}$ on the class $\mathcal{G}_\ell$. We fix the IDs of all vertices in advance, i.e., they are the same for all graphs sampled from $\mathcal{D}_{\mathsf{EI}_m}$. In particular, we specify that $|W| = m^2$ and we assign the set $[m^2]$ as the IDs of the vertices in W. The sets U_i are singletons, i.e., $U_i = \{u_i\}$, and there is a perfect matching $\{u_1, v_1\}, \dots, \{u_\ell, v_\ell\}$ between U and V.

We will choose the edges in the cut sets $E_1, \dots, E_\sigma$ and the labels of W by sampling the input from the product distribution on certain set families for which set disjointness is known to be hard:

▶ **Lemma 10** (follows from Lemmas 1 and 2 in [9]). *There exist set families $\mathcal{X}, \mathcal{Y} \subseteq \binom{[m^2]}{m}^4$ such that (a) $|\mathcal{X}| \geqslant 2^{(m \log m)/4}$, (b) $|\mathcal{Y}| \leqslant \frac{1}{4} m \log m$. Moreover, for all distinct $X, X' \in \mathcal{X}$, it holds that (c) $|X \cap X'| \leqslant \frac{m}{4}$, and (d) there exists $Y \in \mathcal{Y}$ such that Y has a nonempty intersection with either X or X'.*

---

[3] The weak 2-coloring problem was introduced in the seminal work of [21].

[4] $\binom{[N]}{m}$ denotes the family of all m-element subsets of $[N]$.

We make use of the set families guaranteed by Lemma 10 to sample a graph G from $\mathcal{D}_{\mathsf{EI}_m}$ as follows:

1. Sample σ uniformly from $[\ell]$.
2. For each v_i, sample a random set $X \in \mathcal{X}$ and connect v_i to each $w \in W$ that has an ID in X.
3. Randomly pick a set $Y \in \mathcal{Y}$ and label the nodes in W according to the output of the resulting indicator function on W: That is, for $j \in [m^2]$, the label of w_j is 1 if $j \in Y$ and 0 otherwise.

Figure 4a shows a graph sampled from $\mathcal{D}_{\mathsf{EI}_m}$.

▶ **Lemma 11.** *Problem* EI_m *is embeddable with input distribution* $\mathcal{D}_{\mathsf{EI}_m}$*, and satisfies Properties (P1), (P2), and (P3) (as defined in Section 2).*

4.3 A Lower Bound for the EI_m Problem

To prove that the EI_m problem requires a large transcript length, we use a reduction from set disjointness.

▶ **Lemma 12** (implicit in Theorem 4 in [9])**.** *Solving set disjointness in the one-way two-party model with a public coin randomized protocol that succeeds with probability* $\frac{1}{2} + \epsilon$*, for some constant* $\epsilon > 0$*, has a communication complexity of* $\Omega(m \log m)$ *bits, when Diane's input is sampled uniformly from* $\mathcal{X}$ *and Edward's input is sampled uniformly from* $\mathcal{Y}$*.*

▶ **Lemma 13.** *Consider a public coin randomized protocol* $\mathcal{P}$ *that solves the* EI_m *problem with error* $\delta \leqslant \frac{1}{25}$*. If* $|\Pi_B| \leqslant \frac{1}{16} m^3 \log m$*, then* $|\Pi_\sigma| = \Omega(m \log m)$*.*

Proof. We show a reduction from the set disjointness problem [26] in the one-way two-party model, where there are two players, Diane and Edward that are given subsets X and Y respectively. Diane sends a single message to Edward who must decide whether $X \cap Y = \emptyset$.

Given an instance of set disjointness, Diane and Edward will simulate the assumed EI_m protocol $\mathcal{A}$ on a graph sampled from $\mathcal{D}_{\mathsf{EI}_m}$ by embedding the set disjointness instance into the neighborhood of node v_σ. Suppose that Diane has input $X \in \mathcal{X}$ and Edward has input $Y \in \mathcal{Y}$, both of which were sampled uniformly. As required, they choose the cardinalities $|U| = \ell$ and $|W| = m^2$, and make each U_i a singleton set. Moreover, they fix the IDs of all nodes in advance such that the set $[m^2]$ defines the IDs of the nodes in W. Note that this also determines the perfect matching between U and V. In the simulation, Diane will simulate only Alice_σ, whereas Edward simulates Alice_i $(i \neq \sigma)$ and Charlie. Note that, in addition to the edges, the players also need to assign binary vertex labels to the vertices in W:

1. Using public randomness, they uniformly sample an index σ from $[\ell]$.
2. Diane uses her input X to define the IDs of the neighbors of v_σ in W.
3. Similarly, for each $v_i \in V$ $(i \neq \sigma)$, Edward uniformly samples a random $X_i \in \mathcal{X}$ and connects v_i to W by connecting v_i to each $w \in W$ with an ID in X_i.
4. Edward uses his input Y to assign the labels of the nodes in W. That is, for each index $j \in Y$, the label of w_j is 1, and he labels all w_k $(k \notin Y)$ with 0.

It is straightforward to verify that the sampling procedure executed by Diane and Edward results in an input assignment to players $\text{Alice}_1, \ldots, \text{Alice}_\ell$, and Charlie that is the same as in distribution $\mathcal{D}_{\mathsf{EI}_m}$. Therefore, Diane can simulate the EI_m protocol for Alice_σ and send the resulting message to Edward who, in turn, is able to simulate Alice_i $(i \neq \sigma)$. Once Edward receives Diane's message, he knows Z, $\Pi_{(\leqslant \ell)}$, and σ, and hence he also knows Charlie's

input. Since $|\Pi_B| \leqslant \frac{1}{16}m^3 \log m$ and $\ell = \lceil m^3 \log m \rceil$ in $\mathcal{D}_{\mathsf{EI}_m}$, it follows that $\frac{|\Pi_B|}{\ell} \leqslant \frac{1}{16}$. Thus, Edward invokes the reconstruction protocol $\mathcal{R}$ guaranteed by Lemma 5 to recover Charlie's output. He answers that X and Y are disjoint if and only if Charlie's output is "no". Since $\mathcal{R}$ succeeds with probability at least $2^{-\left(\frac{|\Pi_B|}{\ell}+3\sqrt{\delta}\right)} \geqslant 2^{-(1/16+3/5)} > 0.63$, applying Lemma 12 completes the proof of Lemma 13. ◀

4.4 Proof of Theorem 2

▶ **Theorem 2** (restated). *Any $\frac{1}{25}$-error randomized algorithm that verifies if a labeling of a subset of vertices forms a weak 2-coloring of an n-node input graph, requires a worst case message length of $\Omega\left(n^{1/3}\log^{2/3} n\right)$ bits in* SKETCH *and* BCC_1*. The same bound holds for deciding whether a subset of nodes forms a maximal independent set.*

We prove the theorem via a reduction from the EI_m problem in the SMP model. Let G be an input graph sampled from $\mathcal{D}_{\mathsf{EI}_m}$ and let $\mathcal{Q}$ be a protocol that satisfies the premise of the theorem. The players will simulate $\mathcal{Q}$ on a graph H, which extends G with some edges and adds a vertex coloring, as we describe in more detail below. Each player Alice$_i$ simulates $\mathcal{Q}$ for node v_i, while assigning color 0 to v_i. Bob simulates all nodes in W and adds an edge between some arbitrary node in $w \in W_1$ and every node in W_0. For the simulation, the nodes in W_b ($b \in \{0,1\}$) are colored with color b. See Figure 4b for an example of the resulting graph. Charlie, on the other hand, simulates the referee and all nodes in U, where he colors u_σ with 0 and the nodes in $U \setminus \{u_\sigma\}$ with 1. Moreover, he adds an edge between u_σ and some arbitrary u_j ($j \neq \sigma$). Note that Charlie knows which node is u_σ since the index σ is part of his input. The edges added by Bob and Charlie ensure that every node (with the possible exception of v_σ) has a differently-colored neighbor by construction. It follows that the output of the EI_m protocol verifies whether the given coloring of G is valid:

▶ **Observation 14.** *The coloring is a weak 2-coloring if and only if $E_\sigma \cap W_1$ is nonempty. An analogous property holds for the question whether the vertices with color 1 form a maximal independent set or a minimal dominating set.*

Now, assume towards a contradiction that the worst case sketch length produced by protocol $\mathcal{Q}$ is at most $\frac{1}{16}m \log m$. This ensures that Bob sends a message of at most $\frac{1}{16}|W|m \log m = \frac{1}{16}m^3 \log m$ bits in the simulation. Since this satisfies the premise of Lemma 13, it follows that the node v_σ simulated by Alice$_\sigma$ sends a sketch of length $\Omega(m \log m)$ in the worst case. The total number of nodes in H is $n = |U| + |V| + |W| = 2\lceil m^3 \log m \rceil + m^2 = \Theta\left(m^3 \log m\right)$. Hence it follows that $\log m = \Omega(\log n)$ and thus $m = \Omega\left((n/\log n)^{1/3}\right)$. We conclude that Alice$_\sigma$'s sketch must have a length of $\Omega(n^{1/3}\log^{2/3} n)$ bits. By Observation 14, the same result holds for verifying a maximal independent set or a minimal dominating set.

5 A Lower Bound for k-ECSS in Sketching Model

In this section, we will apply Lemma 6 for showing a lower bound of $\Omega\left(k \log^2 \frac{n}{k}\right)$ on the message size for computing a k-edge connected spanning subgraph (k-ECSS).

High-level Overview. We first define an embeddable problem, the $\mathsf{ER}_{k,m}$ problem, and a suitable input distribution on the lower bound graphs $\mathcal{G}_\ell$ (see Section 2.1). We consider the $\mathsf{ER}_{k,m}$ problem in the SMP model, where each vertex in V has m neighbors and the goal is to find a subset of k edges in the cut E_σ of the input graph. Subsequently, we show that it is

in fact an embeddable problem (see Sec. 2) that satisfies Properties (P1) and (P2), and thus Lemma 6 applies. We then simulate the assumed $\mathsf{ER}_{k,m}$ protocol in the one-way two-party communication model and use it to solve the $\mathsf{UR}^{\subset}_{k}$ problem defined in [17], which implies a lower bound on the length of Alice$_\sigma$'s message. The final step is to simulate a given k-ECCS protocol $\mathcal{P}$ designed for the SKETCH model to solve the $\mathsf{ER}_{k,m}$ problem in the SMP model, and this will yield the sought lower bound on the worst case sketch size of $\mathcal{P}$.

5.1 The Edge Recovery Problem $\mathsf{ER}_{k,m}$

We consider the simultaneous multiparty model and the graph class $\mathcal{G}_\ell$, where each vertex in V has exactly m neighbors. A protocol solves the $\mathsf{ER}_{k,m}$ problem if, after receiving the messages from Alice$_1, \ldots,$ Alice$_\ell$, and Bob, player Charlie outputs a subset of k edges in the cut E_σ.

5.2 The Hard Input Distribution $\mathcal{D}_{\mathsf{ER}_{k,m}}$

Our distribution is similar to the one used in [22], albeit with some crucial differences. Let $\ell := \lceil m^2 \log^2 \frac{m}{k} \rceil$ and let $\Gamma = \lceil m^2/k \rceil$. To sample a graph $G \in \mathcal{G}_\ell$ from $\mathcal{D}_{\mathsf{ER}_{k,m}}$, we fix the IDs of nodes in V to be the set $[\ell]$ and perform the following steps:

1. Uniformly sample σ from $[\ell]$.
2. Fix the size of the sets $U_1, \ldots, U_\ell$, and W to be Γ. Sample $(\ell+1)$ disjoint random subsets $A_1, \ldots, A_{\ell+1}$, each of size Γ from $F_0 := [\ell^2] \setminus [\ell]$, and use A_i to assign the IDs to the nodes in U_i, whereas, for the nodes in W, we use $A_{\ell+1}$.
3. For each $v_i \in V$, we choose a random m-element set $S \subseteq [\Gamma]$ and a uniformly random $T \subset S$ such that $|S \setminus T| \geqslant k$ and $|T| \geqslant k$. We connect v_i to $|S \setminus T|$ random vertices from W and $|T|$ random vertices in U_i.

Figure 5a shows an instance of a graph sampled from $\mathcal{D}_{\mathsf{ER}_{k,m}}$.

▶ **Lemma 15.** *The total number of nodes in graph G is $n = O\big(m^4 \log^2(m/k)\big)$.*

▶ **Lemma 16.** *The $\mathsf{ER}_{k,m}$ problem is embeddable with distribution $\mathcal{D}_{\mathsf{ER}_{k,m}}$, i.e., satisfies properties (P1) and (P2).*

5.3 A Lower Bound for the $\mathsf{ER}_{k,m}$ Problem

We use Lemma 6 to show the following:

▶ **Lemma 17.** *Consider a deterministic protocol $\mathcal{P}$ that solves the $\mathsf{ER}_{k,m}$ problem with probability at least $1 - o(1)$ on inputs sampled from $\mathcal{D}_{\mathsf{ER}_{k,m}}$, and suppose that $|\Pi_B| = o(\ell)$. Then, there exists a deterministic protocol $\mathcal{R}$ that succeeds with probability at least $1 - o(1)$ on inputs from $\mathcal{D}_{\mathsf{ER}_{k,m}}$, just by inspecting Charlie's input and the transcripts of Alice$_1, \ldots,$ Alice$_\ell$, i.e., while omitting Bob's transcript Π_B.*

The $\mathsf{UR}^{\subset}_{k}$ Problem. There are two players, Diane and Edward. For some integer $N > 0$, Diane is given a set $S \subseteq [N]$ and Edward starts with a subset $T \subset S$. To solve the $\mathsf{UR}^{\subset}_{k}$ problem, Diane sends a single message to Edward, who in turn must output k elements in $S \setminus T$. Theorem 3 in [17] shows a worst case communication complexity lower bound of $\Omega(k \log^2(N/k))$ bits for algorithms that succeed with constant probability. For the purpose of our reduction, we need a slightly more specific result:

In the full version of the paper, we show how to adapt Theorem 3 in [17] to obtain the following result:

▶ **Lemma 18.** *Consider a universe of size N. Let $m = \lfloor\sqrt{Nk}\rfloor$ and suppose that $k \leqslant N/2^{10}$. Suppose that Diane's set S is chosen from $\binom{[N]}{m}$ and Edward's input set is any proper subset $T \subset S$ under the restriction that $|S \setminus T| \geqslant k$ and $|T| \geqslant k$. Then, any $\mathsf{UR}_k^{\subset}$ protocol $\mathcal{P}$ that errs with small constant probability in the one-way 2-party model with public coins, has a worst case transcript length of $\Omega(k \log^2(\frac{N}{k}))$ bits.*

In the proof of the next lemma, we use the hardness of $\mathsf{UR}_k^{\subset}$ to show a lower bound for the $\mathsf{ER}_{k,m}$ problem by simulating the SMP model in the 2-party setting. We point out that the approach is similar to Section 3 of [22] and postpone the full proof to the full version.

▶ **Lemma 19.** *Consider a protocol $\mathcal{P}$ that solves the $\mathsf{ER}_{k,m}$ problem with error at most $o(1)$ on inputs sampled from $\mathcal{D}_{\mathsf{ER}_{k,m}}$. If $|\Pi_B| = o(\ell)$, where $\ell = \lceil m^2 \log^2 \frac{m}{k} \rceil$, then $|\Pi_\sigma| = \Omega\big(k \log^2 \frac{m}{k}\big)$.*

5.4 Proof of Theorem 3

▶ **Theorem 3** (restated). *Any public coin randomized algorithm that computes a k-edge connected spanning subgraph of an n-node graph in SKETCH or BCC_1 with probability at least $1 - o(1)$, has a worst case message length of $\Omega\big(k \log^2 \frac{n}{k}\big)$ bits, for any $k = o\Big(\frac{n^{1/4}}{\log^{1/2} n}\Big)$.*

We first describe the hard input distribution $\mathcal{D}_{k\text{-ECSS}}$ for computing a k-connected spanning subgraph in the SKETCH model, which is a simple extension of $\mathcal{D}_{\mathsf{ER}_{k,m}}$:

1. Sample a graph G from $\mathcal{D}_{\mathsf{ER}_{k,m}}$.
2. Graph H contains all edges of G; in addition, we make the subgraph induced by each U_i ($i \in [\ell]$) a clique, and we also add a clique on W.

Let $\mathcal{A}$ be a deterministic algorithm that computes a k-connected spanning subgraph in the SKETCH model on inputs sampled from $\mathcal{D}_{k\text{-ECSS}}$ with probability at least $1 - o(1)$. Note that the result immediately extends to randomized algorithms by a simple application of Yao's lemma. Given a graph G sampled from the hard input distribution $\mathcal{D}_{\mathsf{ER}_{k,m}}$, we add the necessary edges to G according to $\mathcal{D}_{k\text{-ECSS}}$ and then simulate $\mathcal{A}$ on the resulting graph H to solve the $\mathsf{ER}_{k,m}$ problem in the SMP model. Figure 5b shows an example of this graph.

Observe that every U_i and W consist of $\lceil m^2/k \rceil > k$ vertices and hence there are k edge-disjoint paths between any two vertices that lie within such a set. Moreover, we sample the neighborhoods of v_i such that $|E(v_i, W)| \geqslant k$ and $|E(v_i, U_i)| \geqslant k$ (see Step 3 of $\mathcal{D}_{\mathsf{ER}_{k,m}}$). This ensures that there are at least k edge-disjoint paths between all pairs of vertices of H:

▶ **Observation 20.** *Graph H is k-edge connected.*

For each $i \in [\ell]$, player Alice$_i$ simulates $\mathcal{A}$ for node v_i and sends the corresponding sketch to Charlie. Bob, on the other hand, sends Charlie the concatenated sketches of the nodes in W, which he computes by simulating $\mathcal{A}$ given their respective neighborhood in H as an input. Finally, Charlie, simulates $\mathcal{A}$ for all nodes in U, and he also simulates the referee. It follows immediately from the input assignment of the $\mathsf{ER}_{k,m}$ problem that the players have the necessary information to perform the simulation. Observe that every k-edge connected subgraph of H must include k edges in the cut $E(v_\sigma, W)$, and hence the simulation solves the $\mathsf{ER}_{k,m}$ problem with the same probability of success.

Let L be the worst case sketch length of protocol $\mathcal{A}$. In our simulation, Bob's message is of length $|\Pi_B| \leqslant L\,|W| \leqslant O\Big(\frac{L\,m^2}{k^2}\Big)$. Assume that $L = o\big(k \log^2 \frac{m}{k}\big)$, as otherwise we are done. By Lemma 15, we know that $\log \frac{m}{k} = \Omega\big(\log \frac{n}{k}\big)$, which implies that $|\Pi_B| = o\big(m^2 \log^2 \frac{n}{k}\big) = o(\ell)$. By applying Lemma 19, we conclude that Alice$_\sigma$ must send a message of $\Omega\big(k \log^2 \frac{m}{k}\big) = \Omega\big(k \log^2 \frac{n}{k}\big)$ bits in the worst case.

6 A Streaming Lower Bound for k-ECSS

In this section, we consider the data streaming setting, where the algorithm learns about the input graph as a stream of edges. That is, in the *fully dynamic turnstile model*, the algorithm observes the stream entries sequentially. Each entry of the stream refers to two vertices u and v and indicates whether the edge $\{u, v\}$ is added or removed from the current graph, and the algorithm needs to react to this update. The main objective is to minimize the amount of memory used by the algorithm while taking (preferably) only a single pass over the data stream.

We show a memory lower bound for computing a k-ECSS in the fully dynamic turnstile model by extending the work of [17].

▶ **Theorem 4** (restated). *Any Monte Carlo data structure for computing a k-edge connected spanning subgraph of an n-node graph requires $\Omega\big(k\,n\log^2\frac{n}{k}\big)$ space in the one-pass fully dynamic turnstile model.*

In our proof of Theorem 4, we use a reduction from the 2-party communication complexity, where we need to solve multiple instances of $\mathsf{UR}^{\subset}_k$ in parallel. We first recall the definition of the $\mathsf{UR}^{\subset}_k$ problem from Section 5: We are given the universe $[N]$ and there are two players called Alice and Bob. Alice obtains a set $S \subseteq [N]$ and Bob has a subset $T \subset S$. Alice sends a message to Bob who must then output k elements in $S \setminus T$.[5] We define the *ℓ-fold $\mathsf{UR}^{\subset}_k$ problem*, where Alice and Bob obtain ℓ independently sampled instances of $\mathsf{UR}^{\subset}_k$ (on the same universe) and they need to solve all of them, again, assuming that Alice can send only a single message to Bob.

▶ **Lemma 21.** *Consider any $k = \Omega(\log N)$ and a universe of size $N > k$. Any one-way communication protocol that solves the ℓ-fold $\mathsf{UR}^{\subset}_k$ problem with error at most δ requires $\Omega\big((1-\delta)k\,\ell\,\log^2\big(\frac{N}{k}\big)\big)$ bits.*

We now show how the lemma implies Theorem 4: Suppose that there exists an algorithm $\mathcal{A}$ that maintains a k-edge connected spanning subgraph in the turnstile model. We simulate $\mathcal{A}$ in the 2-party model. Our simulation is similar to the one used for showing a lower bound on the memory needed for maintaining a spanning forest in Lemma 1 of [22]. Consider a graph G with vertex sets X and Y, each of size ℓ, for some $\ell > k$. The IDs of the vertices in X are given by $[\ell]$, whereas the neighborhood of the i-th vertex in Y will correspond to the i-th instance of $\mathsf{UR}^{\subset}_k$. Recall that Alice starts with the input $S_1, \ldots, S_\ell$ and Bob has input $T_1, \ldots, T_\ell$, where (S_i, T_i) is the i-th instance of $\mathsf{UR}^{\subset}_k$. Alice and Bob will perform edge insertions/removals and execute the streaming algorithm $\mathcal{A}$ accordingly. Alice first inserts edges such that X forms a clique. Then, for each set S_i and each $x \in S_i$, Alice adds an edge $\{x, y_i\}$ to G. Subsequently, Alice sends the memory state of $\mathcal{A}$ to Bob who, in turn, for each T_i and each $x' \in T_i$, continues to simulate $\mathcal{A}$ by removing $\{x', y_i\}$ from G. Recall that $|S_i \setminus T_i| \geqslant k$, which guarantees that the degree of each node in Y is at least k after the last update. Moreover, the nodes in X form a clique of size greater than k, and hence it follows that G is k-edge connected. Finally, Bob returns the output of $\mathcal{A}$ which must include k edges incident to each $y_i \in Y$ with probability at least $1 - \delta$ to ensure k-connectivity. Each one of these edges corresponds to an element in $S_i \setminus T_i$, and hence the simulation solves ℓ-fold $\mathsf{UR}^{\subset}_k$ with precisely the same probability. Since Lemma 21 tells us that ℓ-fold $\mathsf{UR}^{\subset}_k$ has a communication complexity of $\Omega(k\,n\log^2(n/k))$ for $\ell = n$, it follows that $\mathcal{A}$ must use at least $\Omega(k\,n\log^2(n/k))$ bits of memory. This completes the proof of Theorem 4.

[5] Here, we restrict ourselves to the case where the inputs satisfy that $|S \setminus T| \geqslant k$.

7 Future Work and Open Problems

Our results reveal insights into the communication complexity of distributed graph verification problems. However, we are not aware of any communication-efficient 1-round verification algorithm for these type of symmetry breaking problems.

▶ **Open Problem 1.** Is there an algorithm that verifies an LCL problem in just a single round of the broadcast congested clique while sending $o(m)$ bits on graphs with m edges?

While we showed a lower bound on the memory for maintaining a k-edge connected spanning subgraph in the turnstile model, the more fundamental question regarding the space required to solve graph connectivity (i.e., $k = 1$) has yet to be answered, as pointed out in [28]:

▶ **Open Problem 2.** Is there a lower bound of $\Omega(n \log^3 n)$ memory for solving graph connectivity in the turnstile model?

References

1 Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Christoph Lenzen. Fooling views: a new lower bound technique for distributed computations under congestion. *Distributed Comput.*, 33(6):545–559, 2020. doi:10.1007/s00446-020-00373-4.

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

3 Sepehr Assadi, Gillat Kol, and Rotem Oshman. 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*, pages 79–88, 2020. doi:10.1145/3382734.3405732.

4 Baruch Awerbuch, Oded Goldreich, David Peleg, and Ronen Vainish. A trade-off between information and communication in broadcast protocols. *J. ACM*, 37(2):238–256, 1990. doi:10.1145/77600.77618.

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

6 Matthias Bonne and Keren Censor-Hillel. Distributed detection of cliques in dynamic networks. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 132:1–132:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.132.

7 Lijie Chen and Ofer Grossman. Broadcast congested clique: Planted cliques and pseudorandom generators. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 248–255, 2019.

8 T. Cover and J.A. Thomas. *Elements of Information Theory, second edition.* Wiley, 2006.

9 Anirban Dasgupta, Ravi Kumar, and D Sivakumar. Sparse and lopsided set disjointness via information theory. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 517–528. Springer, 2012.

10 Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 367–376, 2014. doi:10.1145/2611462.2611493.

11 Orr Fischer, Tzlil Gonen, Fabian Kuhn, and Rotem Oshman. Possibilities and impossibilities for distributed subgraph detection. In *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, SPAA 2018, Vienna, Austria, July 16-18, 2018*, pages 153–162, 2018. doi:10.1145/3210377.3210401.

12 Pierre Fraigniaud, Pedro Montealegre, Pablo Paredes, Ivan Rapaport, Martín Ríos-Wilson, and Ioan Todinca. Computing power of hybrid models in synchronous networks. *arXiv preprint arXiv:2208.02640*, 2022.

13 Mohsen Ghaffari and Fabian Kuhn. Distributed MST and broadcast with fewer messages, and faster gossiping. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 30:1–30:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.DISC.2018.30.

14 Robert Gmyr and Gopal Pandurangan. Time-message trade-offs in distributed algorithms. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 32:1–32:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.DISC.2018.32.

15 Jacob Holm, Valerie King, Mikkel Thorup, Or Zamir, and Uri Zwick. 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*, pages 896–909, 2019. doi:10.1109/FOCS.2019.00058.

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

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

18 Bruce M. Kapron, Valerie King, and Ben Mountjoy. Dynamic graph connectivity in polylogarithmic worst case time. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1131–1142, 2013. doi:10.1137/1.9781611973105.81.

19 Valerie King, Shay Kutten, and Mikkel Thorup. Construction and impromptu repair of an MST in a distributed network with o(m) communication. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 71–80. ACM, 2015. doi:10.1145/2767386.2767405.

20 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

21 Moni Naor and Larry J. Stockmeyer. What can be computed locally? *SIAM J. Comput.*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

22 Jelani Nelson and Huacheng Yu. 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*, pages 1844–1860, 2019. doi:10.1137/1.9781611975482.111.

23 Shreyas Pai and Sriram V Pemmaraju. 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, 2020.

24 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000. doi:10.1137/1.9780898719772.

25 Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. *SIAM J. Comput.*, 45(1):174–196, 2016. doi:10.1137/15M1007525.

26 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

27 David P. Woodruff and Qin Zhang. When distributed computation is communication expensive. *Distributed Computing*, 30(5):309–323, 2017. doi:10.1007/s00446-014-0218-3.

28 Huacheng Yu. 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*, pages 1856–1873, 2021. doi:10.1137/1.9781611976465.111.

A Tools from Information Theory

We give the definitions of some basic notions from information theory and restate some facts (without proofs) that we use throughout the paper. We refer the reader to [8] for additional details. Throughout this section, we assume that X, Y, Z, etc. are discrete random variables. We use capitals to denote random variables and corresponding lowercase characters for values, unless stated otherwise. When computing expected values, we sometimes use the subscript notation $\mathbf{E}_x$ to make it explicit that the expectation is taken over the distribution of a specific random variable X.

▶ **Definition 22.** *The* entropy of X *is defined as*

$$\mathbf{H}[X] = \sum_x \mathbf{Pr}[X=x] \log_2(1/\mathbf{Pr}[X=x]). \tag{6}$$

The conditional entropy of X conditioned on Y *is given by*

$$\begin{aligned}\mathbf{H}[X \mid Y] &= \mathop{\mathbf{E}}_y[\mathbf{H}[X \mid Y=y]] \\ &= \sum_y \mathbf{Pr}[Y=y]\, \mathbf{H}[X \mid Y=y].\end{aligned} \tag{7}$$

▶ **Definition 23.** *Let X and Y be discrete random variables. The* mutual information between X and Y *is defined as*

$$\mathbf{I}[X:Y] = \sum_{x,y} \mathbf{Pr}[x,y] \cdot \log\left(\frac{\mathbf{Pr}[x,y]}{\mathbf{Pr}[x]\,\mathbf{Pr}[y]}\right) \tag{8}$$

▶ **Definition 24.** *Let X, Y, and Z be discrete random variables. The* conditional mutual information of X and Y *is defined as*

$$\mathbf{I}[X:Y \mid Z] = \mathbf{H}[X \mid Z] - \mathbf{H}[X \mid Y,Z]. \tag{9}$$

▶ **Lemma 25.** $\mathbf{I}[X:Y \mid Z] \leqslant \mathbf{H}[X \mid Z] \leqslant \mathbf{H}[X]$.

▶ **Lemma 26** (Theorem 6.1 in [26])**.** *Consider any random variable X. Every encoding of X has expected length at least $\mathbf{H}[X]$.*

▶ **Lemma 27** (Theorem 6.12 in [26])**.** *Let $X_1, \ldots, X_k$ be independent random variables, and let B be jointly distributed. Then,*

$$\sum_{i=1}^{k} \mathbf{I}[X_1 : B] \leqslant \mathbf{I}[X_1, \ldots, X_k : B].$$

▶ **Lemma 28** (Data Processing Inequality, see Theorem 2.8.1 in [8]). *If random variables X, Y, and Z form the Markov chain $X \to Y \to Z$, i.e., the conditional distribution of Z depends only on Y and is conditionally independent of X, then*

$$\mathbf{I}[X : Y] \geqslant \mathbf{I}[X : Z].$$

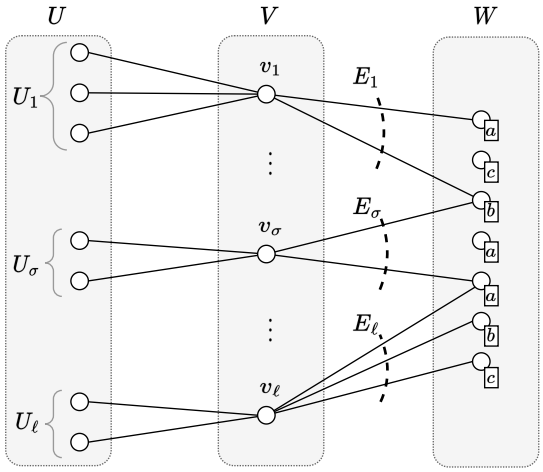
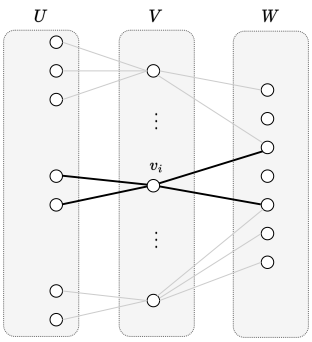
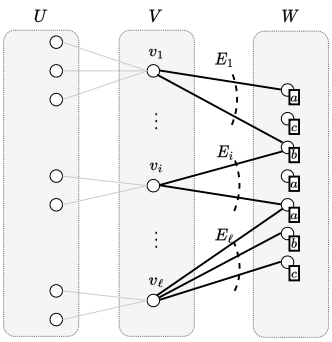


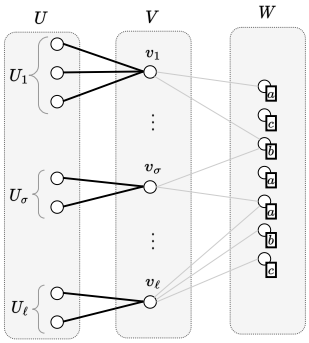
Figure 1 The general structure of the lower bound graphs in $\mathcal{G}_\ell$. Each v_i is connected to a subset of the vertices in U_i and to a subset of the vertices in W. Note that the cardinalities of the sets $U_1, \ldots, U_\ell$, and W, as well as the edges $E(U, V)$ and $E(V, W)$ depend on the hard input distribution, which is problem-specific. In this example, the labels of the nodes in W are chosen from $\{a, b, c\}$.



(a) Alice$_i$'s input: the entire neighborhood of v_i.



(b) Bob's input: $E(V, W)$ and $\mathcal{L}_W$.



(c) Charlie's input: $E(U, V)$, σ, and $\mathcal{L}_W$.

Figure 2 The input assignment in the SMP model.

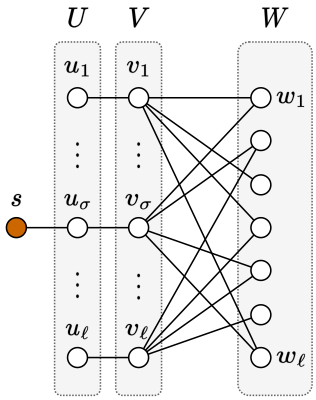
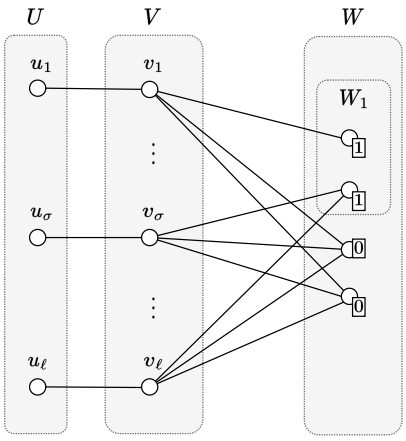
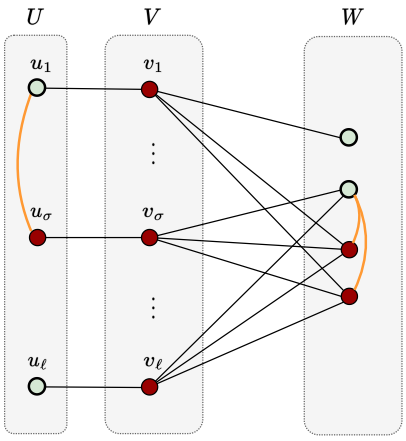


Figure 3 The lower bound graph G for proving the hardness of computing a BFS tree in the distributed sketching model and the one-round broadcast congested clique. The BFS tree rooted at s must include all edges in the cut $E(v_\sigma, W)$. Note that we sample the edges in the cut $E(V, W)$ according to the hard input distribution of the Index_{ℓ^2} problem.

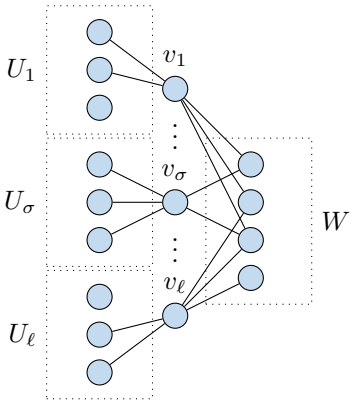


(a) A graph G sampled from distribution $\mathcal{D}_{\mathsf{EI}_m}$. To solve the EI_m problem in the simultaneous multiparty model, Charlie must output some edge in $E_\sigma \cap W_1$ if it exists.

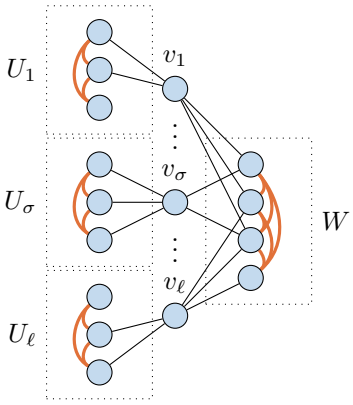


(b) The graph used in the simulation argument. The players add the thick orange edges to the graph sampled from $\mathcal{D}_{\mathsf{EI}_m}$ (see Figure 4a). Red corresponds to color 0 and green to color 1. The given vertex coloring forms a weak 2-coloring if and only if v_σ has a green-colored neighbor in W.

Figure 4 The lower bound graph construction used in the proof of Theorem 2.



(a) A graph sampled from the lower bound distribution $\mathcal{D}_{\mathsf{ER}_{k,m}}$. The distribution ensures that, for all $i \in [\ell]$, the cuts $E(U_i, v_i)$ and $E(v_i, W)$ have at least $k = 2$ edges.



(b) A k-edge connected graph G that we use to prove a lower bound for the k-ECSS problem, for $k = 2$. To simulate a SKETCH algorithm, the players sample G from the lower bound distribution $\mathcal{D}_{\mathsf{ER}_{k,m}}$ and then add the thick orange edges to form cliques of size greater than k.

Figure 5 The lower bound graph construction of Theorem 3.