Average-Case Communication Complexity of Statistical Problems

Cyrus Rashtchian CRASHTCHIAN@ENG.UCSD.EDU
UC San Diego

David P. Woodruff DWOODRUF@CS.CMU.EDU
Carnegie Mellon University

Peng Ye YEP17@MAILS.TSINGHUA.EDU.CN
Tsinghua University

Hanlin Zhu ZHUHL17@TSINGHUA.ORG.CN
Tsinghua University

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

We study statistical problems, such as planted clique, its variants, and sparse principal component analysis in the context of average-case communication complexity. Our motivation is to understand the statistical-computational trade-offs in streaming, sketching, and query-based models. Communication complexity is the main tool for proving lower bounds in these models, yet many prior results do not hold in an average-case setting. We provide a general reduction method that preserves the input distribution for problems involving a random graph or matrix with planted structure. Then, we derive two-party and multi-party communication lower bounds for detecting or finding planted cliques, bipartite cliques, and related problems. As a consequence, we obtain new bounds on the query complexity in the edge-probe, vector-matrix-vector, matrix-vector, linear sketching, and $\mathbb{F}_2$-sketching models. Many of these results are nearly tight, and we use our techniques to provide simple proofs of some known lower bounds for the edge-probe model.

Keywords: Planted clique, sketching, query complexity, communication complexity

1. Introduction

The planted clique and sparse principal component analysis problems embody an enduring interest in computational vs. statistical trade-offs. These problems have the intriguing property that it may be easy to detect the presence of planted structure in super-polynomial time. However, it is a central open problem to determine whether an efficient solution exists, even though we know that one is possible information theoretically (Abbe, 2017; Bandeira et al., 2018; Berthet and Rigollet, 2013; Jordan and Mitchell, 2015).

The planted clique problem involves distinguishing between two distributions on n-vertex graphs. In the first, the graph is generated from the Erdos-Renyi model $G(n, 1/2)$, where each edge is independently present with $1/2$ probability. The second distribution $G(n, 1/2, k)$ has a k-clique planted in a random subset of k vertices, and the remaining edges exist independently with $1/2$ probability. From an information theoretic point of view, detection is possible if $k \geq (2 + \delta) \log_2 n$ for any constant $\delta > 0$ because the largest clique in a random graph has size $(2 + o(1)) \log_2 n$ almost surely. When the clique is very large, i.e., $k \gg \sqrt{n}$, many methods can distinguish the two distributions in polynomial time (and find the planted clique). However, when $k = o(\sqrt{n})$, all known algorithms require super-polynomial time (Alon et al., 1998; Arias-Castro and Verzelen, 2014; Dekel et al., 2011; Feige and Krauthgamer, 2000; Frieze and Kannan, 2008; Kucera, 1995; Ma et al., 2015).

A natural question is to understand the complexity of statistical problems in other models. Query-based algorithms form the basis of sublinear time methods for massive graphs (Avrachenkov et al., 2014; Leskovec and Faloutsos, 2006; Maiya and Berger-Wolf, 2010; Soundarajan et al., 2017). In network monitoring applications, streaming and sketching algorithms are used for real-time data analytics when the graph is too large to fit into memory or when the edges arrive over time (Ahmad et al., 2017; Gupta et al., 2016). Typical network activity could be modeled as a distribution over edge connections, and the presence of some planted subgraph structure could signify anomalous or suspicious group behavior (Chandola et al., 2009; Huang and Kasiviswanathan, 2015). This motivates understanding the query and streaming complexity of detection problems under average-case distributions.

Rácz and Schiffer consider the edge-probe model, which measures the number of edge existence queries to solve a problem (Rácz and Schiffer, 2020); this model is also known as the dense graph model (Goldreich, 2017; Goldreich and Ron, 2009; Goldreich et al., 1998). Here, there are no computational constraints, making it feasible to study clique detection when $k = o(\sqrt{n})$. Rácz and Schiffer show that $\widetilde{\Theta}(n^2/k^2)$ edge-probe queries are necessary and sufficient to detect a planted k-clique, and they also prove similar bounds for finding the clique (Rácz and Schiffer, 2020). A more general model involves linear sketches (Woodruff, 2014). Representing an $n \times n$ matrix $\boldsymbol{A}$ as a vector $\mathsf{vec}(\boldsymbol{A})$ with n^2 entries, a query returns $\boldsymbol{u}^\top \mathsf{vec}(\boldsymbol{A})$ for a vector $\boldsymbol{u}$ with polynomially-bounded entries. A restriction of the sketching model, the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ model, returns $\boldsymbol{u}^\top \boldsymbol{M}\boldsymbol{v}$ for vectors $\boldsymbol{u}, \boldsymbol{v}$, where $\boldsymbol{M}$ is an unknown matrix (Rashtchian et al., 2020). This specializes the Mv model, which returns $\boldsymbol{M}\boldsymbol{v}$ (Sun et al., 2019). These models all generalize edge-probes.

Since there are advantages and disadvantages to these various types of queries, it is often worthwhile to understand the complexity of solving certain problems in each of the models. If an algorithm can be implemented in a more restricted model, then it may be more useful in practice. On the other hand, a lower bound for a more general model would imply the same bound for any specialized model. To this end, it is common to prove lower bounds on the communication complexity (Kushilevitz and Nisan, 2006; Rao and Yehudayoff, 2020). Then, by showing that a query-efficient or space-efficient algorithm can solve a communication problem, lower bounds can be derived for the query complexity.

1.1. Our Results

We provide a general method to encode a communication game as a statistical graph or matrix problem while retaining the input distribution. In the next subsection, we provide technical details about how to execute this approach. Here, we summarize our query complexity and communication upper and lower bounds. Throughout, we often assume that $k = o(\sqrt{n})$ because otherwise there is often an $O(1)$ query upper bound (see Section 2.2).

- In Section 3, we provide an alternate proof of the Rácz-Schiffer bound showing that detecting a planted k-clique requires $\Omega(n^2/k^2)$ edge-probe queries (Rácz and Schiffer, 2020). Then, we investigate whether stronger models are able to succeed with fewer queries. Our most technical contribution shows that $\widetilde{\Omega}(n^2/k^4)$ queries are necessary to detect a planted k-clique in the linear sketching model (and hence also in the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ model); this appears in Section 5, and it follows from an information complexity argument in Section 4. The linear sketching model is more powerful, in general, than the edge-probe model, and we leave open the question of whether the query complexity is $\widetilde{\Omega}(n^2/k^4)$ or $\widetilde{O}(n^2/k^2)$ or somewhere in between.

- We also consider detecting and finding a planted $k \times k$ bipartite clique (biclique). When we have to output the planted biclique, we provide nearly tight upper and lower bounds in the Mv model. When $k = o(\sqrt{n})$, it is easy to see that $O(\frac{n}{k})$ Mv queries suffice (Section 2.2), and we prove that $\Omega(\frac{n}{k \log n})$ queries are necessary to find a planted $k \times k$ biclique (Section B). We also obtain trade-offs in other query models, depending on the clique size, where we generally consider a planted $r \times s$ biclique. We provide an $\widetilde{\Omega}(n^2/(r^2 s^2))$ lower bound for general linear sketching. To complement this, we exhibit an algorithm in the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ model that uses only $\widetilde{O}(n^2/(r^2 s))$ queries, assuming that $r \gg \sqrt{n \log n}$. Our algorithm borrows ideas from CountSketch (Charikar et al., 2002), as high-degree planted vertices can be considered as ℓ_2 heavy hitters. Finally, we give a stronger $\widetilde{\Omega}(n^2/(rs))$ lower bound in the edge-probe model, which is tight up the logarithmic factors (Section A).

- We further uncover qualitatively different trade-offs by considering variants of the planted clique detection problem. We investigate the sandwich semi-random version of planted clique from (Feige and Krauthgamer, 2000). In this model, an adversary is allowed to remove some number of edges that are not part of the planted clique. For this variant, we prove that $\widetilde{\Theta}(n^2/k^2)$ bits are required and sufficient for a related communication game (Section C). The complexity in the linear sketching model is $\widetilde{\Theta}(n^2/k^2)$, where the upper bound follows from existing algorithms in the edge-probe model. This indicates that any improved algorithm for the usual planted clique problem would require non-trivial algorithmic techniques.

- Then, we study a *promise* variant. If the players know that the planted clique occurs in one of $O(n^2/k^2)$ edge-disjoint subgraphs, then $\widetilde{\Theta}(n^2/k^4)$ bits of communication are both necessary and sufficient for detection (Section D). This shows that the k^4 dependency is tight in this promise variant. While our motivation is technical, related promise problems have been studied for other average-case reductions (Brennan and Bresler, 2020) and for network inference when prior information has been previously obtained (Soundarajan et al., 2017).

- Finally, we also provide lower bounds for the hidden hubs problem (Kannan and Vempala, 2017) (Section E) and for sparse PCA (Berthet and Rigollet, 2013) (Section F).

- Our edge-probe lower bounds extend to the $\mathbb{F}_2$ sketching model, where querying with a vector $\boldsymbol{u}$ returns the value $\boldsymbol{u}^\mathsf{T}\mathsf{vec}(\boldsymbol{A})$ over $\mathbb{F}_2$, where again $\boldsymbol{A}$ is the adjacency matrix. While we do not know a separation between these models for finding planted structure, the $\mathbb{F}_2$ sketching model is a formal generalization of the edge-probe model (using a standard basis vector as the query). Our results also immediately provide upper and lower bounds for streaming algorithms, but we focus on communication and query complexity for brevity.

1.2. Technical Overview

LOWER BOUNDS IN THE GENERAL LINEAR SKETCHING AND $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ MODELS

We start by describing our lower bound techniques for the planted clique problem. The average case notion of our problems makes reductions from standard problems in communication complexity, such as multi-player set disjointness, non-trivial, as they do not give us instances from our desired distribution. This is unlike existing worst-case clique communication lower bounds (Braverman et al., 2018; Halldórsson et al., 2012), which reduce directly from set disjointness.

We instead use a communication complexity model that allows the players to have access to shared public randomness, as well as private randomness. Then, we consider a multi-player hypothesis testing problem, introduced in (Braverman et al., 2016), where each player either receives an independent sample from a distribution μ_0 or a distribution μ_1 and the players would like to decide which case they are in. Using a strong data processing inequality, the information cost of such a protocol was shown to be $\Omega(1)$ if $\mu_0 \geq \frac{1}{c}\mu_1$ for a constant $c > 0$, even when information is measured with respect to μ_0 alone (Braverman et al., 2016). We combine this with the information complexity framework of (Bar-Yossef et al., 2004) to prove a direct sum theorem for solving the OR of multiple copies of this problem (here, the "OR" of many instances evaluates to true whenever at least one of the component instances evaluates to true). We guarantee when the OR evaluates to 1, then exactly one copy is from μ_1. We note that Weinstein and Woodruff (Weinstein and Woodruff, 2015) prove a distributional result for *simultaneous* multi-party communication, but this would only apply to non-adaptive query algorithms, whereas our results apply even to adaptive query algorithms. Moreover, the distributions considered in (Weinstein and Woodruff, 2015) are specific, and not the same as the ones we need for our applications, which we now discuss.

The main remaining task is to choose distributions μ_0 and μ_1 so that the resulting multi-copy distribution matches that of the planted clique problem. We first use a clique partitioning scheme of (Conlon et al., 2014) which although related to results on proving worst-case clique communication lower bounds (Braverman et al., 2018; Halldórsson et al., 2012), does not seem to have been used before in this context. This gives us $\Omega(n^2/k^2)$ edge-disjoint cliques on k vertices each. We have $\binom{k}{2}$ players, and each is assigned one edge from each clique. We let μ_0 be the uniform distribution, so that if the OR of the $\Omega(n^2/k^2)$ instances above is 0, we have a graph from $G(n, 1/2)$. Otherwise the OR evaluates to 1 (i.e., the OR being true corresponds to having at least one planted k-clique). We let μ_1 be the constant distribution with value 1 (i.e., the value is always 1 in these positions), and we randomly permute vertex labels, so that in this case we have exactly one planted clique on k vertices and otherwise have a $G(n, 1/2)$ instance. By our choice of μ_0 and μ_1, this gives us an $\Omega(1)$ information cost lower bound per copy, an $\Omega(n^2/k^2)$ lower bound for the OR problem, and an $\tilde{\Omega}(n^2/k^4)$ $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ query lower bound by simulating each query across $\Theta(k^2)$ players.

From the $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ to the $\mathsf{M}\mathsf{v}$ model

While the above communication game can be applied to the $\mathsf{M}\mathsf{v}$ model, it would only give us an $\Omega(n/k^4)$ lower bound. We strengthen this to a nearly *optimal* $\widetilde{\Omega}(n/k)$ lower bound for the related planted bipartite clique (biclique) problem, and where the algorithm is promised to return a $k \times k$ biclique when it exists (later, we more generally consider $r \times s$ bicliques, but for this discussion, we let $k = r = s$). The issue is the algorithm retrieves too much information ($\Omega(n)$ bits) with each $\mathsf{M}\mathsf{v}$ query. To get around this, we only consider inputs when there actually exists a randomly planted biclique. Although the distinguishing problem is trivial now (we always have a biclique), since the algorithm must return the vertices in the biclique, it still has a non-trivial task.

Next, we fix the set of k left vertices in the biclique. They are random and form a valid input instance, but they are known to the algorithm. This might seem counterintuitive, as it only helps the algorithm. Also, each $\mathsf{M}\mathsf{v}$ query only reveals $O(k \log n)$ bits of information, and thus we will only pay an $O(k \log n)$ factor instead of an $O(k^2 \log n)$ factor per query, in our query to communication simulation. The next idea is to also fix $k-1$ of the vertices on the right in the biclique; they are again random, and so they form a valid input instance, but they are known to the algorithm. Again, this might seem counterintuitive, but it helps our analysis because now it gives us $n - (k-1) = \Omega(n)$

possible remaining vertices in the right part, and any of them can be the potential last right vertex. This gives us $\Omega(n)$ possible cliques rather than $\Omega(n/k)$ if we were to partition the right vertices into vertex-disjoint bicliques, which we would need in order to have edge-disjoint bicliques, since we have already fixed the vertices in the left of the biclique. We show the algorithm needs to reveal $\Omega(n)$ bits of information to figure out the missing right vertex. Since each player now corresponds to a row in this $k \times n$ matrix, when we do the query to communication simulation we obtain an $\widetilde{\Omega}(n/k)$ overall lower bound, losing one factor of k for the number of players, and a factor of $O(\log n)$ to transmit its dot product with the query vector. The full details are in Section B.

1.3. Related Work

Even though average-case reductions have been studied for many models, there seems to be a gap in our understanding for communication complexity. In the context of graph problems, current techniques usually construct a hard instance by identifying specific families of graphs that encode a communication game. Unfortunately, this does not answer the question of whether the complexity remains high when we are in a hypothesis testing setting; e.g., MAX-CLIQUE is NP-Hard, but distinguishing between $G(n, 1/2)$ or $G(n, 1/2, k)$ for $k \geq (2 + \delta) \log_2 n$ can be solved in quasi-polynomial time by checking all cliques of size $O(\log n)$. In the realm of communication complexity, known results for approximating the size of the maximum clique do not match the distributions for the planted clique problem (Braverman et al., 2018; Halldórsson et al., 2012). At a high level, we also use a combination of (nearly) covering the graph with subsets of vertices and then reducing to a version of set disjointness. However, we invoke a slightly different graph decomposition, ensuring that the subsets intersect in at most one vertex, and moreover, the players can use public randomness to match the input distribution of the average-case statistical problems.

The conjectured hardness of detecting/finding cliques has been used to derive statistical vs. computational trade-offs for many average-case problems (Brennan and Bresler, 2019; Berthet and Rigollet, 2013; Boix-Adserà et al., 2019; Kunisky et al., 2019). Thus, an open direction from our work is whether analogous reductions can extend our communication complexity lower bounds to other statistical problems (e.g., the stochastic block model) or property testing in the dense graph model (Goldreich, 2017). Our study of the promise planted clique problem is inspired by conjectures regarding the secret linkage of prior information of the planted clique (Brennan and Bresler, 2020). Recent work studies the query complexity of approximating the maximum clique and/or finding the clique in the $G(n, 1/2)$ model (Alweiss et al., 2021; Feige et al., 2020; Mardia et al., 2020). Lower bounds for the planted clique problem have been shown for the statistical query model (Feldman et al., 2017) and for sum-of-squares (Barak et al., 2019; Meka et al., 2015).

2. Preliminaries

Let $[n] = \{1, 2, \ldots, n\}$. We use *with high probability* to mean $1 - O(1/n^c)$ for a constant $c > 0$ and *with constant probability* to mean at least $9/10$. The notation $\widetilde{O}, \widetilde{\Omega}, \widetilde{\Theta}$ hides $\mathrm{polylog}(n)$ factors.

2.1. Problems, Games, and Query Models

For each problem, we define the null hypothesis H_0 and the alternate hypothesis H_1. The goal is to determine which hypothesis a graph or matrix has been drawn from with constant probability.

Table 1: Average-case query complexity for statistical problems (Section 2 has definitions). We suppress $\mathrm{polylog}(n)$ factors and assume $k = o(\sqrt{n})$, $k = \Omega(\log n)$ and $r = \Omega(\sqrt{n \log n})$.

	$\mathbb{F}_2$ sketch & edge-probe	Linear sketch & $\mathsf{u}^\mathsf{T}\mathsf{Mv}$	Mv	Ref.
PC	$\widetilde{\Theta}(n^2/k^2)$	$\widetilde{\Omega}(n^2/k^4)$ $\widetilde{O}(n^2/k^2)$	$\widetilde{\Omega}(n/k^4)$ $O(n/k)$	Section 3 & 5
BPC	$\widetilde{\Theta}(n^2/(rs))$	$\widetilde{\Omega}(n^2/(r^2s^2))$ $\widetilde{O}(n^2/(r^2s))$	$\widetilde{\Omega}(n/(r^2s^2))$ $\widetilde{O}(n/\min(r,s))$	Section A
FINDBPC	$\widetilde{\Theta}(n^2/k^2)$	$\widetilde{\Omega}(n^2/k^4)$ $\widetilde{O}(n^2/k^2)$	$\widetilde{\Theta}(n/k)$	Section B
SRPC	$\widetilde{\Theta}(n^2/k^2)$	$\widetilde{\Theta}(n^2/k^2)$	$\widetilde{\Omega}(n/k^2)$ $O(n/k)$	Section C
PPC	$\widetilde{\Theta}(n^2/k^2)$	$\widetilde{\Theta}(n^2/k^4)$	$\widetilde{\Omega}(n/k^4)$ $\widetilde{O}(\min(\frac{n}{k}, \frac{n^2}{k^4}))$	Section D
HH	$\widetilde{\Omega}(n^2/k^4)$ $\widetilde{O}(n^2/k)$	$\widetilde{\Omega}(n^2/k^4)$ $\widetilde{O}(n^2/k)$	$\widetilde{\Omega}(n/k^4)$ $\widetilde{O}(n/k)$	Section E
SCDC	$\Omega\left(\frac{k^2}{\theta^2}\right)$	$\widetilde{\Omega}\left(\frac{k^4}{t^2\theta^4}\right)$	$\widetilde{\Omega}\left(\frac{k^4}{t^3\theta^4}\right)$	Section F

- **Planted Clique (PC).** The input is a graph with n vertices, described as an $n \times n$ adjacency matrix $\boldsymbol{A}$. For H_0 each edge occurs with probability $1/2$, i.e., $\boldsymbol{A} \sim G(n, 1/2)$ in the Erdós-Renyi model. For H_1 there is a planted k-clique, i.e., a set R is randomly chosen over all size k subsets of $[n]$; first $\boldsymbol{A} \sim G(n, 1/2)$, then we set A_{ij} to 1 for all $i, j \in R$ with $i \neq j$.

- **Bipartite Planted Clique (BPC).** The input is a bipartite graph with n vertices on each side, described as an $n \times n$ matrix $\boldsymbol{A}$. For H_0 each edge occurs with probability $1/2$, i.e., each entry of $\boldsymbol{A}$ is sampled from $\mathtt{Bernoulli}(1/2)$. For H_1 there is an $r \times s$ planted biclique, i.e., two sets R and S are randomly chosen over all size r subsets of $[n]$ and all size s subsets of $[n]$ respectively, and then A_{ij} is set to 1 for all $i \in R$ and $j \in S$, and all the remaining entries follow $\mathtt{Bernoulli}(1/2)$ independently.

- **Semi-Random Planted Clique (SRPC).** The semi-random model was introduced by Blum and Spencer (Blum and Spencer, 1995). There are variants of the semi-random model, and we specifically consider the sandwich model (Feige and Kilian, 1998). In this model, there is an adversary which can remove arbitrary edges outside the planted clique. We describe our hypothesis testing problem as follows:

  H_0: the adversary chooses any graph G^* such that $G_{\min} \subseteq G^* \subseteq G_{\max}$, where $G_{\max}$ is a random graph drawn from $G(n, 1/2)$ and $G_{\min}$ is the empty graph.

  H_1: the adversary chooses any graph G^* such that $G_{\min} \subseteq G^* \subseteq G_{\max}$, where $G_{\max}$ is a random graph drawn from $G(n, 1/2, k)$ and $G_{\min}$ only contains the planted clique.

- **Promise Planted Clique (PPC).** There is a fixed and known collection S of subsets of k vertices such that every pair of subsets intersects in at most one vertex. For H_0, the graph is $G(n, 1/2)$ as in the PC problem. For H_1, the planted clique is chosen from S. Clearly, $|S| \leq n^2/k^2$, and if $|S| = \Theta(n^2/k^2)$, then $k \leq O(\sqrt{n})$. The motivation for the PPC problem is that we use a graph decomposition result to define S for some of our reductions (see Section 3). Thus, the PPC problem captures the relative difficulty of the problem when the set of possible cliques is known in advance. From an algorithmic point of view, this makes the problem trivial. On the other hand, from a query complexity point of view, our upper and lower bounds for sketching algorithms nearly match for the PPC problem.

- **Hidden Hubs (HH).** In the hidden hubs model $H(n, k, \sigma_0, \sigma_1)$, an $n \times n$ random matrix $\boldsymbol{A}$ is generated as follows (Kannan and Vempala, 2017). First randomly choose a subset S of k rows. Entries in rows outside S are generated from the Gaussian distribution $p_0 = \mathcal{N}(0, \sigma_0^2)$. For each row in S, choose k entries to be generated from $p_1 = \mathcal{N}(0, \sigma_1^2)$, and the other $n - k$ entries from p_0. The hypothesis testing problem (HH problem) is to distinguish H_0 and H_1, where H_0 is an $n \times n$ random matrix with all entries generated from $\mathcal{N}(0, \sigma_0^2)$, and H_1 is the model $H(n, k, \sigma_0, \sigma_1)$.

- **Sparse Component Detection Challenge (SCDC).** We consider the sub-Gaussian version of the Sparse Principal Component Analysis (SPCA) problem, using elements of a known reduction from the PC problem (Berthet and Rigollet, 2013). The empirical variance of t vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \in \mathbb{R}^d$ in direction $\boldsymbol{v}$ is defined as $\widehat{\mathsf{var}}(\boldsymbol{v}) = \frac{1}{t}\sum_{i=1}^{t}(\boldsymbol{v}^\top \boldsymbol{X}_i)^2$. Let θ and k be parameters, and let $\zeta \in (0, 1)$ be a fixed, small constant. We let $\mathcal{D}_0$ denote the set of product distributions over t i.i.d. vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \in \mathbb{R}^d$ such that for all unit vectors $\boldsymbol{v}$ we have

$$\Pr\left[|\widehat{\mathsf{var}}(\boldsymbol{v}) - 1| > 4\sqrt{\frac{\log(2/\zeta)}{d}} + 4\frac{\log(2/\zeta)}{d}\right] \le \zeta. \tag{2.1}$$

In other words, $\mathcal{D}_0 := \{\mathbf{P}_0 \mid \textit{Eq. (2.1)} \text{ holds}\}$, and $\mathcal{D}_0$ contains, e.g., isotropic distributions. We let $\mathcal{D}_1^{k,\theta}$ denote the set of product distributions over t i.i.d. vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \in \mathbb{R}^d$ such that for all unit vectors $\boldsymbol{v}$ with at most k nonzero entries ($\|\boldsymbol{v}\|_0 \le k$), we have

$$\Pr\left[(\widehat{\mathsf{var}}(\boldsymbol{v}) - (1 + \theta)) < -2\sqrt{\frac{\theta k \log(2/\zeta)}{d}} - 4\frac{\log(2/\zeta)}{d}\right] \le \zeta. \tag{2.2}$$

Similarly, $\mathcal{D}_1^{k,\theta} := \{\mathbf{P}_1 \mid \textit{Eq. (2.2)} \text{ holds}\}$. Then, for the SCDC problem, we define two hypotheses to test; the inputs $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t$ are drawn from $\mathbf{P}$ such that

$$\mathcal{H}_0 : \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \sim \mathbf{P}_0 \in \mathcal{D}_0 \qquad \text{vs.} \qquad \mathcal{H}_1 : \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \sim \mathbf{P}_1 \in \mathcal{D}_1^{k,\theta}.$$

Our goal is to distinguish which family of distributions $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t$ is sampled from. The motivation is that $\mathcal{D}_0$ contains $\mathcal{N}(0, \boldsymbol{I}_d)$ and $\mathcal{D}_1$ contains $\mathcal{N}(0, \boldsymbol{I}_d + \theta \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})$ when $\boldsymbol{u}$ is a k-sparse unit vector. Hence, this generalizes the spiked covariance model (Berthet et al., 2013; Brennan and Bresler, 2019).

Communication Games. Throughout we use *problem* to refer to the detection problems above, and we use *game* to refer to the analogous communication complexity problem. For PC, SRPC, and BPC we define the games as follows for $t \ge 2$ players. The players receive edge-disjoint subgraphs of a graph G such that the union of the edges equals the whole graph. Equivalently, the players receive $n \times n$ adjacency matrices corresponding to their subset of the edges, and they must solve the corresponding problem on the graph defined by the *sum* of the adjacency matrices (which is the adjacency matrix of the whole graph since the edge sets are disjoint). The players are promised that G is either drawn from H_0 or H_1 as in the problems defined above. To succeed, the players must determine which distribution G is drawn from with constant probability. For PPC, the only difference is that the players also all know the set S of possible locations for the planted clique. For HH, the players instead receive $n \times n$ matrices with disjoint supports, where the sum of these matrices is drawn from one of the two hypotheses. While many of these games have been defined for the union of the graphs, we also make use of an XOR variant for the 2-player version of the games. More precisely, Alice and Bob each receive adjacency matrices G_1 and G_2, which are not necessarily disjoint in the support of their entries. Then, they must solve the corresponding problem on the graph $G_1 \oplus G_2$, where an edge is present in $G_1 \oplus G_2$ if and only if it is present in exactly one of G_1 or G_2. In other words, we use the XOR of the adjacency matrices. This variant will be used for proving $\mathbb{F}_2$ sketching lower bounds (which will immediately imply the edge-probe lower bounds).

Query Models. Matrices and vectors have polynomially bounded integer entries. Let $\mathsf{vec}(\boldsymbol{A})$ denote the vectorization of an $n \times n$ matrix $\boldsymbol{A}$, i.e., n^2 entries listed in a fixed order.

- **Edge-Probe Model.** Querying position (i, j) returns A_{ij}.
- $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ **Model.** Querying with vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ returns $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}$ over $\mathbb{R}$.
- Mv **Model.** Querying with a vector $\boldsymbol{v} \in \mathbb{R}^n$ returns $\boldsymbol{A}\boldsymbol{v}$ over $\mathbb{R}$.
- $\mathbb{F}_2$ **Sketching Model.** Querying with vector $\boldsymbol{u} \in \mathbb{F}_2^{n^2}$ returns $\boldsymbol{u}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{A})$ over $\mathbb{F}_2$.
- **Linear Sketching Model.** Querying with vector $\boldsymbol{u} \in \mathbb{R}^{n^2}$ returns $\boldsymbol{u}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{A})$ over $\mathbb{R}$.

There is a relationship regarding lower bounds for the Mv model vs. lower bounds for the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ or general linear sketching model. In particular, any query in the Mv can be simulated by n queries in the linear sketching model (in fact, in the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ model by taking $\boldsymbol{u}$ to be the n standard basis vectors one at a time). We often simply state lower bounds for the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ or linear sketching models, but using this relationship, we obtain the entries in Table 1 for the Mv model (with the exception of the FINDBPC problem, where we obtain a stronger lower bound in Section B).

Finding vs. Detecting. While we mostly focus on detection problems, we also consider the variant where the algorithm should output the planted clique if there is one. We denote this by adding Find before the problem name (e.g., for the FindBPC problem/game, the algorithm/protocol should output the planted $r \times s$ biclique). For many of the models we study, it is straightforward to find the clique by using only a factor of $\mathrm{polylog}(n)$ more queries than for detection. We describe the upper bounds in Section 2.2, and we prove a communication lower bound for the FindBPC game in Section B, which implies a lower bound for matrix-vector queries.

2.2. Algorithms for Detecting and Finding

We review algorithms in the query models listed above. We start with the PC problem, where $k \geq 10 \log n$ for simplicity. Previous work on the edge-probe model presents a simple sampling algorithm using $O((n/k)^2 \log^2 n)$ queries: choose a subset B of $100(n/k) \log n$ vertices uniformly at random, query all pairs in B, and compute the largest clique in this induced subgraph (Rácz and Schiffer, 2020). If there is no planted clique, then the largest induced clique has size at most $3 \log n$ with high probability; otherwise, there is an induced clique B' of size at least $4 \log n$ with high probability. To actually find the clique, the next step is to query all neighbors of B', which reveals the whole planted clique using a total of $O((n/k)^2 \log^2 n + n \log n)$ edge-probe queries. The same general idea leads to algorithms for the SRPC, BPC, and PPC problems as well (for detecting and finding).

We mention two improvements to the edge-probe algorithm in the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ and Mv models. For both models, the query vectors may have bit-complexity $O(\log n)$ in each entry. In the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ model, we can query all pairs in B by using only $O((n/k)^2 \log n)$ queries, saving a $\log n$ factor (use exponentially increasing entries to simulate $O(\log n)$ edge-probe queries with one $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ query). In the Mv model, we can query with an indicator vector to receive all neighbors of a vertex. Again by using exponentially increasing entries, we can query $O(\log n)$ vertices at a time. Therefore, we can query all pairs in B with $O(n/k)$ queries; we can also find the planted k-clique with an additional $O(k)$ queries by looking at the shared neighborhood of B'.

We also note that a single query suffices when $k \geq c\sqrt{n}$ for a large enough constant $c > 1$ in the $\mathsf{u}^\mathsf{T}\mathsf{Mv}$, Mv, and general linear sketching models. We can use a single query to detect a planted k-clique with constant probability by counting the edges (i.e., ones in the matrix). Indeed, the total number of edges is at least $\frac{1}{2}\binom{n}{2} + \binom{k}{2} \geq n^2/2 + c'n$ when there is a planted clique. Otherwise, it is at most $\frac{1}{2}\binom{n}{2} + c''n$ with constant probability for some $c'' < c'$, allowing us to distinguish the two cases. In light of this, we focus on the case of $k = o(\sqrt{n})$ for the remainder of the paper.

2.3. Communication Complexity Preliminaries

We consider a multi-player communication model, where $t \geq 2$ players communicate via a publicly shared blackboard (i.e., all players see all messages). The total number of bits written on the blackboard is the measure of communication. This model generalizes both point-to-point and broadcast models, and hence, our lower bounds hold for both the message passing and broadcast settings. We let Π denote the collection of all messages written on the blackboard. Abusing notation slightly, we use Π for both the protocol and the transcript $\Pi \in \{0,1\}^*$ in bits. The communication cost is the length of Π, which we denote as $|\Pi|$, in the worst case over the support of the input distribution. In other words, we consider the *randomized communication complexity* (see, e.g., (Rao and Yehudayoff, 2020)). At the termination of the communication protocol, one of the players must output the answer using a function of Π with no constraints on the computation time (e.g., in the games defined above, the player should output which of the two distributions the input has been sampled from). We consider the success probability of randomized protocols (players have access to both public, shared random bits and private random bits). Throughout, the exact success probability will not be important, and we consider the randomized communication complexity of solving a problem with constant success probability, e.g., 9/10.

We use a standard $\Omega(n)$ lower bound on the 2-player UNIQUE DISJOINTNESS game (Kalyanasundaram and Schintger, 1992; Razborov, 1992). Two players each have a bitstring $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$. They are promised that one of the following two cases holds: either (i) for all $i \in [n]$ either $x_i = 0$ or $y_i = 0$ or both, or (ii) there is a unique $i \in [n]$ such that $x_i = y_i = 1$ and for all $i' \neq i$, either $x_i = 0$ or $y_i = 0$ or both. The UNIQUE DISJOINTNESS game is to communicate and determine which case they are in.

3. Warm-up: Lower Bound for PC in the Edge-Probe Model

We present a simple proof demonstrating the main ideas of our reduction method. We first explain the graph decomposition, and then we use this to prove a communication lower bound for the XOR version of the PC game in Theorem 4. As a consequence, in Corollary 5, we provide an alternate proof of the Rácz-Schiffer lower bound of $\Omega(n^2/k^2)$ edge-probe queries for the PC problem (Rácz and Schiffer, 2020).

The key aspect of our communication lower bounds is using a graph decomposition into a set of edge-disjoint cliques.[1] By ensuring that the cliques are edge-disjoint, while covering most of the graph, we can partition edges among the players while preserving the input distribution.

Lemma 1 (Lemma 6.6 in (Conlon et al., 2014)) *Let $k \geq 2$ and n be positive integers and let $f(n,k)$ denote the minimum number of cliques, each on at most k vertices, needed to clique partition the complete graph K_n. If $n > k$, then $f(n,k) = \Theta\left(\max\left\{(n/k)^2, n\right\}\right)$.*

Remark 2 (Number of uncovered edges) *First, recall that we are interested in the case when $k = o(\sqrt{n})$. In this regime, the above lemma can be strengthened to show that $f(n,k) = (1 + o(1))\frac{n^2}{k(k-1)}$, which is essentially best possible (Conlon et al., 2014). In such a clique partition, there are $\Omega(n^2)$ edges belonging to cliques of size $\Omega(k)$. Indeed, assume there are m_1 cliques of size $\Omega(k)$ and m_2 cliques of size $o(k)$, and observe that $m_1 + m_2 = \Theta\left((n/k)^2\right)$. Now if there were only $o(n^2)$ edges belonging to cliques of size $\Omega(k)$, then the total number of edges would be $o(n^2) + m_2 \cdot o(k^2) \leq o(n^2) + \Theta\left((n/k)^2\right) \cdot o(k^2) = o(n^2)$, a contradiction. Thus, $m_1 = \Omega(n^2)/\Theta(k^2) = \Omega\left(n^2/k^2\right)$.*

Remark 3 (Size of cliques) *The above lemma only guarantees cliques of size at most k. However, by slightly changing constants, we can guarantee $\Theta(n^2/k^2)$ cliques of size exactly k. Indeed, by a standard counting argument, a constant fraction of the cliques must have size at least αk for a constant $\alpha \in (0,1)$. Therefore, we apply lemma with $k' = k/\alpha$, and then find $\Theta(n^2/k^2)$ cliques of size exactly k by restricting to the subcliques of the cliques of size k' if necessary.*

Theorem 4 *Any protocol that solves the XOR version of the PC game with constant success probability must communicate at least $\Omega(n^2/k^2)$ bits.*

Proof We reduce to the 2-player UNIQUE DISJOINTNESS game with input length $\ell = \Theta(n^2/k^2)$. Let Alice and Bob have inputs $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^\ell$, respectively. We use $\boldsymbol{x}, \boldsymbol{y}$ to build a random input graph G as follows. First, randomly permute the vertex labels. Then, use Lemma 1 to obtain a collection S of $\Theta(n^2/k^2)$ edge-disjoint cliques with k vertices; for each edge not covered by S, choose each of them with probability $1/2$ independently, call this graph G', and give it to Alice (see Remark 2 and Remark 3 for details about the number of uncovered edges and the clique size, respectively). Index the subgraphs as $S = \{Z_1, \ldots, Z_\ell\}$. We repeat the following process independently for each $i \in [\ell]$. Alice and Bob will receive graphs G_1^i and G_2^i based on x_i and y_i, and these graphs will be supported on the vertices of Z_i. Color all edges of a k-clique K_k^i with four colors uniformly at random using public randomness. Then,

- $\boldsymbol{x}_i = 0 \implies$ add all edges in K_k^i with colors 1 or 3 to G_1^i
- $\boldsymbol{x}_i = 1 \implies$ add all edges in K_k^i with colors 1 or 2 to G_1^i
- $\boldsymbol{y}_i = 0 \implies$ add all edges in K_k^i with colors 1 or 4 to G_2^i

[1] More formally, a set of *edge-disjoint* cliques is a collection of subsets of vertices $V_1, \ldots, V_\ell$ such that for all $i \neq j$ the subsets V_i and V_j intersect in at most one vertex.

- $\boldsymbol{y}_i = 1 \implies$ add all edges in K_k^i with colors 3 or 4 to G_2^i

Define $G = G' \cup \left(\bigcup_{i=1}^{\ell} G_1^i \oplus G_2^i\right)$. We claim that if $(x_i, y_i) \neq (1,1)$ for all $i \in [\ell]$, then G is distributed according to H_0. Each possible edge is included with probability 1/2 either because of the random coloring or it is in G'. Indeed, for each of the three combinations $(0,0), (1,0), (0,1)$, exactly two colors of edges end up in $G_1^i \oplus G_2^i$. Otherwise, if $(x_i, y_i) = (1,1)$ for some i, then all four colors of edges appear in $G_1^i \oplus G_2^i$, and hence this is the planted clique. By randomly permuting the vertices at the beginning (with public randomness), each k-clique is equally likely. A protocol solving the XOR version of the PC game also solves UNIQUE DISJOINTNESS on ℓ bits and must communicate $\Omega(\ell) = \Omega(n^2/k^2)$ bits. ■

Let G_1, G_2 denote the adjacency matrices for Alice and Bob, respectively. Assume there is a q query algorithm in the $\mathbb{F}_2$ sketching model that solves the PC problem with constant probability. This can be implemented by having Alice compute her q sketches on G_1 and then she sends these q bits to Bob. Then, Bob can complete the execution of the algorithm $G_1 \oplus G_2$ locally, and hence, solve the XOR version of the PC game. Therefore, $q = \Omega(n^2/k^2)$ by Theorem 4, and we get the following.

Corollary 5 *For the PC problem, $\Omega(n^2/k^2)$ queries in the $\mathbb{F}_2$ sketching model are required to distinguish H_0 and H_1 with constant probability.*

4. Parameter Estimation Game and Multi-player Communication

We next prove communication lower bounds that imply query lower bounds for the general linear sketching model. While similar results have appeared before (e.g., (Bar-Yossef et al., 2004; Braverman et al., 2016; Weinstein and Woodruff, 2015)), we are unaware of any results that suffice for the distributional lower bounds that we need for our reductions. The entropy of X is $H(X) = -\sum_x p_x \log_2 p_x$. The mutual information is $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X)$.

Definition 6 (Hellinger Distance) *Consider two probability distributions $f, g : \Omega \to \mathbb{R}$. The square of the Hellinger distance between f and g is $h^2(f,g) := \frac{1}{2}\int_\Omega \left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx$.*

We consider a version of the multi-party UNIQUE DISJOINTNESS game, where n players each receive an m-dimensional binary vector, and they determine whether there is some coordinate such that every player's vector has a one in this coordinate. We also define an input distribution where the vectors are either uniformly random, or there is a planted coordinate that is all ones.

Parameter Estimation (PE) game. Let $B \in \{0,1\}$ be a binary variable, and let $\boldsymbol{V} \in \{0,1\}^m$ be a random binary vector (the distribution of $\boldsymbol{V}$ will depend on B). When $B = 0$, then $\boldsymbol{V}$ is the all zeros vector. When $B = 1$, then there is exactly one entry of $\boldsymbol{V}$ equal to 1, and the entry is chosen uniformly at random. We define two distributions: $\mu_0 = \texttt{Bernoulli}(1/2)$ and $\texttt{Bernoulli}(1)$. Now suppose $\boldsymbol{V} = \boldsymbol{v}$, and the n players each obtain a vector $\boldsymbol{X}^{(i)} \in \{0,1\}^m$, where $X_j^{(i)} \sim \mu_{v_j}$. They need to communicate with each other to determine the value of B with error probability at most δ. We assume that m and n are comparable, i.e., $m = O(n^\alpha)$ for some constant $\alpha > 0$. To set notation, let $\mathcal{Z} \subset \left(\mathscr{X}^{(1)}\right)^m \times \left(\mathscr{X}^{(2)}\right)^m \times \cdots \times \left(\mathscr{X}^{(n)}\right)^m$ be the set of inputs. The PE game corresponds to computing $f : \mathcal{Z} \to \{0,1\}$, which outputs B on inputs $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(n)}$, where the inputs are drawn from the distribution described above (depending on B). For convenience, let $\boldsymbol{X} = (\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(n)})$, and let $\boldsymbol{X}_j = (X_j^{(1)}, X_j^{(2)}, \ldots, X_j^{(n)})$. Also, let $\Pi \in \{0,1\}^*$ be a randomized protocol, where $\Pi(\boldsymbol{X})$ is the transcript when the players have $\boldsymbol{X}$ as inputs, and $|\Pi(\boldsymbol{X})|$ denotes its length in bits. We consider a function $g : \{0,1\}^* \to \{0,1\}$, such that when $B = 0$, then $g(\Pi(\boldsymbol{X})) = 0$ with probability at least $1 - \delta$, and when $B = 1$, then $g(\Pi(\boldsymbol{X})) = 1$ with probability at least $1 - \delta$, where the randomness is from both the input $\boldsymbol{X}$ and the protocol Π, i.e., the players have shared public and also private randomness. In other words, g is the estimator for the parameter estimation problem. We provide a lower bound on the information and communication complexity of solving the PE game, which will be the basis of several of our results.

Next, we recall a standard communication lower bound (see, e.g., (Rao and Yehudayoff, 2020)).

Proposition 7 *For a protocol* Π *and distribution* μ *of inputs,* $\max_{\boldsymbol{X}' \in \mathsf{supp}(\mu)} |\Pi(\boldsymbol{X}')| \geq I(\boldsymbol{X}; \Pi)$.

4.1. Direct Sum and Communication Lower Bound

For distributions μ_0, μ_1 over the same sample space, we write $\mu_1 \leq c \cdot \mu_0$ if the point-wise density of μ_0 is at most c times larger than μ_1 for $c > 0$. For μ_0, μ_1 defined above, we have $c = 2$ and that only a one-sided guarantee is possible (as μ_1 has no mass on 0). We use the distributed strong data processing inequality (Distributed SDPI). Let $\beta(\mu_0, \mu_1)$ denote the *SDPI* constant, which is the infimum over real $\beta \geq 0$ such that $I(B; \Pi) \leq \beta \cdot I(\boldsymbol{X}; \Pi)$ where $B \to \boldsymbol{X} \to \Pi$ forms a Markov chain. This inequality holds with $\beta = 1$, which is the *data processing inequality*. For our results, it suffices to take $\beta = 1$, but for completeness, we state the stronger version of the following theorem.

Theorem 8 (Theorem 3.1 in (Braverman et al., 2016)) *Suppose* $\mu_1 \leq c \cdot \mu_0$ *and* $\beta(\mu_0, \mu_1) = \beta$. *Then,* $c'(c+1)\beta \cdot I(X; \Pi \mid B = 0) \geq h^2(\Pi|_{B=0}, \Pi|_{B=1})$, *where* $c' > 0$ *is an absolute constant. The same holds conditioned on* $B = 1$ *instead of* $B = 0$.

The challenge is to lower bound the information, conditioning on the distribution when $B = 0$, which is the utility of the above theorem. When the protocol is correct with constant probability, the Hellinger distance is also a constant (via a standard connection with total variation distance), and when $\beta = \Theta(1)$, then Theorem 8 provides an $\Omega(1)$ lower bound on the information. This suffices for our purposes because we use a direct sum over many instances and only need an $\Omega(1)$ lower bound on the information to achieve the communication lower bound. We can decompose the PE game on m coordinates to a single coordinate, which follows from standard properties of mutual information, such as subadditivity, since conditioned on $B = 0$, all $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m$ are independent (see e.g. (Bar-Yossef et al., 2004; Braverman et al., 2016)).

Lemma 9 *Fix the input distribution of* $\boldsymbol{X}$ *when* $B = 0$. *Then,*

$$I(\boldsymbol{X}; \Pi | B = 0) \geq \sum_{j=1}^{m} I(\boldsymbol{X}_j; \Pi | B = 0).$$

Proof By definition, $I(\boldsymbol{X}; \Pi | B = 0) = H(\boldsymbol{X} | B = 0) - H(\boldsymbol{X} | \Pi, B = 0)$. Observe that we have $H(\boldsymbol{X} | B = 0) = \sum_{j=1}^{m} H(\boldsymbol{X}_j | B = 0)$ since given $B = 0$, all $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m$ are independent. Also, by subadditivity, $H(\boldsymbol{X} | \Pi, B = 0) \leq \sum_{j=1}^{m} H(\boldsymbol{X}_j | \Pi, B = 0)$. Putting these together, we have that $I(\boldsymbol{X}; \Pi | B = 0) \geq \sum_{j=1}^{m} H(\boldsymbol{X}_j | B = 0) - \sum_{j=1}^{m} H(\boldsymbol{X}_j | \Pi, B = 0) = \sum_{j=1}^{m} I(\boldsymbol{X}_j; \Pi | B = 0)$. ■

Theorem 10 *Assume that $m = \mathrm{poly}(n)$. The communication complexity of the n-player* PE *game on m coordinates with error probability δ is $\Omega(m)$ assuming that $\delta \leq 1/10$.*

Proof The single coordinate PE game is that for a specific $j \in [m]$ and $V_j = v_j \in \{0, 1\}$, each of the n players receives an instance of the variable $X_j^{(i)} \sim \mu_{v_j}$ for $i \in [n]$. Their task is to communicate with each other to determine the value of V_j with error probability at most δ. Let Π be a randomized protocol that solves the n-player PE game on m coordinates with error probability δ. Our goal is to show that

$$I(\boldsymbol{X}_j; \Pi | B = 0) = \Omega(1) \quad \text{for all } j \in [m]. \tag{4.1}$$

By Proposition 7 and Lemma 9, we lower bound the communication by

$$I(\boldsymbol{X}; \Pi | B = 0) \geq \sum_{j=1}^{m} I(\boldsymbol{X}_j; \Pi | B = 0) = m \cdot \Omega(1) = \Omega(m).$$

To show Eq. (4.1), we use Theorem 8. We consider the single coordinate PE game on coordinate j. We construct a protocol $\Pi'(\boldsymbol{X}_j)$ to solve the single coordinate PE game. Our method is to construct another random matrix $\boldsymbol{X}'$ as follows. Using public randomness, players choose a uniformly random $j' \in [m]$, and let $\boldsymbol{X}'_{j'} = \boldsymbol{X}_j$. For $\ell \neq j'$, let $\boldsymbol{X}'_\ell$ be a random vector where each entry is an independent $\texttt{Bernoulli}(1/2)$ variable, sampled by each player independently using private randomness. Since $m = \mathrm{poly}(n)$, the probability that any $\boldsymbol{X}'_\ell$ is an all ones vector is exponentially small. Then, let $\Pi'(\boldsymbol{X}_j) = \Pi(\boldsymbol{X}')$. By this construction, when $B = V_j$, we have that Π has the same distribution as $\Pi'(\boldsymbol{X}_j)$. Since Π could determine the value of B with error probability δ, $\Pi'(\boldsymbol{X}_j)$ can also determine the value of V_j with error δ. Thus, by Theorem 8, $I(\boldsymbol{X}_j; \Pi | B = 0) = I(\boldsymbol{X}_j; \Pi'(\boldsymbol{X}_j) | V_j = 0) = \Omega(1)$ for all $j \in [m]$, where we use $\beta = 1$ and $c = 2$ and that the squared Hellinger distance is $\Theta(1)$ since the success probability is a constant. ■

5. Planted Clique Lower Bound for Linear Sketching

Theorem 11 *For $k = n^\gamma$ where $0 < \gamma < \frac{1}{2}$, any protocol with $\Theta(k^2)$ players that solves the* PC *game with constant success probability must communicate $\Omega\left(n^2/k^2\right)$ bits.*

Proof We reduce from the PE game (Section 4) to the PC game to get hardness of PC from hardness of PE. Given a complete graph with n vertices, by Lemma 1, we can partition most of the edges (or equivalently, vertex pairs) by $\Theta\left(n^2/k^2\right)$ cliques of size k. We consider the PE game with $\binom{k}{2}$ players, each having inputs with $\Theta\left(n^2/k^2\right)$ coordinates (i.e., each player is responsible for one edge in each potential clique). Each V_j corresponds to a clique of size k, and the indicator vector for its $\Theta(k^2)$ edges corresponds to the binary vector $\boldsymbol{X}_j$ (using an arbitrary indexing of the edges). The uncovered edges can be sampled with a public coin to appear with probability $1/2$, and they can be given to any player without loss of generality (see Remark 2 and Remark 3 for details about the number of uncovered edges and the clique size, respectively). Using public randomness, the players randomly relabel all vertices, so that the location of the planted clique is random). By this construction, we have the PE to PC translation: $B = 0$ corresponds to $G(n, 1/2)$ and

$B = 1$ corresponds to $G(n, 1/2, k)$. Hence, the $\Theta(k^2)$ players can solve the PE game by detecting the planted clique. The randomized communication complexity of the PC game is $\Omega\left(n^2/k^2\right)$. ■

The communication lower bound of the PC game with $\Theta(k^2)$ players is $\Omega(n^2/k^2)$. A single query in the general linear sketching model can be simulated with $O(k^2 \log n)$ bits of communication since there are $\Theta(k^2)$ players. Thus, any algorithm that solves the PC problem with constant success probability must use $\widetilde{\Omega}(n^2/k^4)$ queries, and we get the following.

Corollary 12 *Let $k = n^\gamma$ where $0 < \gamma < 1/2$. Then, $\widetilde{\Omega}(n^2/k^4)$ general linear sketching queries are necessary to solve the PC problem with constant success probability.*

6. Conclusion

Motivated by understanding statistical-computational trade-offs, we addressed a variety of related average-case communication complexity problems. To this end, we developed a generic reduction technique that preserves the distribution of graph problems that can be defined in terms of planted subgraphs. Specifically, we proved new lower bounds for the planted clique problem and three variants: the bipartite version, the semi-random version, and the promise version. For the $\mathbb{F}_2$ sketching model (and edge-probe model as a special case), we obtained tight bounds on the query complexity. For the more general linear sketching model, we also proved new lower bounds for these problems, and we demonstrated a lower bound for the hidden hubs problem. Finally, we provided lower bounds for a variant of the SPCA problem.

Looking forward, our techniques may be useful for developing a more general theory of average-case communication complexity. Indeed, the next step could be to explore the natural analogues of other statistical problems that have been reduced to planted clique (Berthet and Rigollet, 2013; Brennan and Bresler, 2019, 2020; Kunisky et al., 2019). A more concrete direction is to close the gaps in Table 1. For example, in the linear sketching model we establish that the query complexity of the PC problem is between $\widetilde{\Omega}(n^2/k^4)$ and $\widetilde{O}(n^2/k^2)$. Similarly, for the planted $r \times s$ biclique problem (BPC), the complexity is between $\widetilde{\Omega}(n^2/(rs)^2)$ and $\widetilde{O}(n^2/(r^2 s))$ when $r \gg \sqrt{n}\log n$. Another direction could be to determine a non-linear query model where an upper bound of $\Theta(n/k^2)$ can be derived when $k = o(\sqrt{n})$ and when the query only reveals $O(\log n)$ bits (compared to the Mv model, which reveals $O(n \log n)$ bits).

Acknowledgments

D. Woodruff would like to thank NSF grant No. CCF-181584, Office of Naval Research (ONR) grant N00014-18-1-256, and a Simons Investigator Award.

References

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134 – 147, 2017.

Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466, 1998.

Ryan Alweiss, Chady Ben Hamida, Xiaoyu He, and Alexander Moreira. On the subgraph query problem. *Combinatorics, Probability and Computing*, 30(1):1–16, 2021.

Ery Arias-Castro and Nicolas Verzelen. Community detection in dense random networks. *Annals of Statistics*, 42(3):940–969, 06 2014.

K. Avrachenkov, N. Litvak, L. Ostroumova Prokhorenkova, and E. Suyargulova. Quick detection of high-degree entities in large directed networks. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, USA, 2014. IEEE Computer Society.

Afonso S Bandeira, Amelia Perry, and Alexander S Wein. Notes on computational-to-statistical gaps: predictions using statistical physics. *Portugaliae Mathematica*, 75(2):159–187, 2018.

Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.

Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735, 2019.

Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Conference on Learning Theory*, pages 1046–1066, 2013.

Quentin Berthet, Philippe Rigollet, et al. Optimal detection of sparse principal components in high dimension. *The Annals of Statistics*, 41(4):1780–1815, 2013.

A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204 – 234, 1995. ISSN 0196-6774. doi: https://doi.org/10.1006/jagm.1995.1034.

Enric Boix-Adserà, Matthew Brennan, and Guy Bresler. The Average-Case Complexity of Counting Cliques in Erdös-Rényi Hypergraphs. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1256–1280. IEEE, 2019.

Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1011–1020, 2016.

Vladimir Braverman, Zaoxing Liu, Tejasvam Singh, NV Vinodchandran, and Lin F Yang. New bounds for the clique-gap problem using graph decomposition theory. *Algorithmica*, 80(2):652–667, 2018.

Matthew Brennan and Guy Bresler. Average-case lower bounds for learning sparse mixtures, robust estimation and semirandom adversaries. *arXiv preprint arXiv:1908.06130*, 2019.

Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP '02, page 693–703, Berlin, Heidelberg, 2002. Springer-Verlag.

David Conlon, Jacob Fox, and Benny Sudakov. Short proofs of some extremal results. *Combinatorics, Probability & Computing*, 23(1):8–28, 2014.

Yael Dekel, Ori Gurel-Gurevich, and Yuval Peres. Finding hidden cliques in linear time with high probability. In *2011 Proceedings of the Eighth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 67–75. SIAM, 2011.

Uriel Feige and Joe Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, FOCS '98, page 674. IEEE Computer Society, 1998.

Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms*, 16(2):195–208, 2000.

Uriel Feige, David Gamarnik, Joe Neeman, Miklós Z Rácz, and Prasad Tetali. Finding cliques using few probes. *Random Structures & Algorithms*, 56(1):142–153, 2020.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2):1–37, 2017.

Alan Frieze and Ravi Kannan. A new approach to the planted clique problem. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

Chao Gao, Zongming Ma, Harrison H Zhou, et al. Sparse cca: Adaptive estimation and computational barriers. *The Annals of Statistics*, 45(5):2074–2101, 2017.

Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.

Oded Goldreich and Dana Ron. Algorithmic aspects of property testing in the dense graphs model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 520–533. Springer, 2009.

Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

Arpit Gupta, Rüdiger Birkner, Marco Canini, Nick Feamster, Chris Mac-Stoker, and Walter Willinger. Network monitoring as a streaming analytics problem. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, HotNets '16, page 106–112, New York, NY, USA, 2016. Association for Computing Machinery.

Magnús M Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. Streaming and communication complexity of clique approximation. In *International Colloquium on Automata, Languages, and Programming*, pages 449–460. Springer, 2012.

Hao Huang and Shiva Prasad Kasiviswanathan. Streaming anomaly detection using randomized matrix sketching. *Proc. VLDB Endow.*, 9(3):192–203, November 2015.

Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

Ravindran Kannan and Santosh Vempala. The hidden hubs problem. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1190–1213, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

Ludek Kucera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995.

Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. *arXiv preprint arXiv:1907.11636*, 2019.

Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 2006.

Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 631–636, 2006.

Zongming Ma, Yihong Wu, et al. Computational barriers in minimax submatrix detection. *The Annals of Statistics*, 43(3):1089–1116, 2015.

Arun S. Maiya and Tanya Y. Berger-Wolf. Sampling community structure. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 701–710, 2010.

Jay Mardia, Hilal Asi, and Kabir Aladin Chandrasekher. Finding planted cliques in sublinear time. *arXiv preprint arXiv:2004.12002*, 2020.

Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 87–96, 2015.

Miklós Z Rácz and Benjamin Schiffer. Finding a planted clique by adaptive probing. *ALEA Latin American Journal of Probability and Mathematical Statistics*, 17:775–790, 2020.

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

Cyrus Rashtchian, David P Woodruff, and Hanlin Zhu. Vector-Matrix-Vector Queries for Solving Linear Algebra, Statistics, and Graph Problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

AA Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106:385–390, 1992.

Sucheta Soundarajan, Tina Eliassi-Rad, Brian Gallagher, and Ali Pinar. ε-wgx: Adaptive edge probing for enhancing incomplete networks. In *Proceedings of the 2017 ACM on Web Science Conference*, page 161–170. Association for Computing Machinery, 2017.

Xiaoming Sun, David P Woodruff, Guang Yang, and Jialin Zhang. Querying a matrix through matrix-vector products. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, 2019.

Omri Weinstein and David P Woodruff. The simultaneous communication of disjointness with applications to data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 1082–1093. Springer, 2015.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

A. Bipartite Planted Clique Detection

For the BPC problem, we first state the bipartite version of the graph decomposition lemma.

Lemma 13 *Given n, r, and s, we can use $\lceil n/r \rceil \cdot \lceil n/s \rceil$ edge-disjoint bicliques to cover an $n \times n$ complete bipartite graph. Moreover, $\lfloor n/r \rfloor \cdot \lfloor n/s \rfloor$ of them are of size $r \times s$.*

Proof Let $a = \lceil n/r \rceil$ and $b = \lceil n/s \rceil$. We can partition the vertices on the left side into a sets $U_1, \ldots, U_a$ so that $|U_1| = \cdots = |U_{a-1}| = r$ and $|U_a| = n - (a-1)r$. Also we can partition the vertices on the right side into b sets $V_1, \ldots, V_b$ so that $|V_1| = \cdots = |V_{b-1}| = s$ and $|V_b| = n - (b-1)s$. The $a \cdot b$ bicliques formed by U_i and V_j for all i, j can cover the whole bipartite graph. ■

Now we can prove a lower bound using the same strategy as Theorem 11.

Theorem 14 *For the BPC problem, suppose $rs \leq n$. Then $\widetilde{\Omega}(n^2/(rs)^2)$ general linear sketching queries are required to distinguish H_0 and H_1 with constant probability.*

Proof We use Lemma 13 to randomly partition a complete bipartite graph. Then we consider the PE game for rs players, where each player receives $\lfloor n/r \rfloor \cdot \lfloor n/s \rfloor = \Theta(n^2/(rs))$ coordinates. We reindex the edges (regardless if they exist or not) in each biclique from 1 to rs. The i-th edge is present in the j-th biclique if and only if the value player i holds at coordinate j is one. For the negligible amount of edges that are not covered by these bicliques (e.g., the subgraphs with size other than $r \times s$), we let the graph contain each of them with probability $1/2$ using public randomness. The communication lower bound of the PE game with $\Theta(rs)$ players on $\Theta(n^2/(rs))$ coordinates is $\Omega(n^2/(rs))$, which implies the same lower bound for the BPC game. A single query in the general linear sketching model can be simulated with $O(rs \log n)$ bits of communication since there are $\Theta(rs)$ players. Thus, any algorithm that solves the BPC problem with constant success probability must use $\widetilde{\Omega}(n^2/(r^2s^2))$ queries. ■

We also design an algorithm when r is larger than $C\sqrt{n \log n}$ (w.l.o.g. we suppose $r > s$) for some constant C, to further close the gap between the upper bound and the lower bound.

Theorem 15 *For the BPC problem, suppose $r \geq C\sqrt{n \log n}$ for $C > 16$. Then there exists an algorithm which can distinguish H_0 and H_1 with high probability, using $\widetilde{O}(n^2/(r^2s))$ $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ queries.*

Proof The algorithm is as follows. We first randomly sample $\widetilde{O}(n/s)$ columns. If there is a planted biclique, then with high probability at least one column that belongs to the planted column will be sampled. We then partition the columns into groups with size $r^2/(16n \log n)$ each, and compute $\sum_{i \in S} x_i$ for each group S where x_i is the sum of column i. The algorithm returns 1 (i.e., there is a planted biclique) if there is an S such that $\sum_{i \in S} x_i \geq n/2 \cdot |S| + r/4$. And it returns 0 if there is no such S. Since the sum of a group can be computed using one $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ query, our algorithm will use $\widetilde{O}(n^2/(r^2s))$ $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ queries in total.

We now prove that our algorithm succeeds with high probability. We consider a set S of $r^2/(16n \log n)$ columns. If S does not contain any columns in the planted biclique, by Hoeffding's inequality, we have $\Pr\left(\sum_{i \in S} x_i - n/2 \cdot |S| \geq r/4\right) \leq \exp(-\frac{2r^2/4^2}{n \cdot |S|}) = e^{-2 \log n} = 1/n^2$, where we consider $\sum_{i \in S} x_i$ as the sum of $n|S|$ random variables with $|S| = r^2/(16n \log n)$. Therefore, $\sum_{i \in S} x_i$ is less than $n/2 \cdot |S| + r/4$ with probability at least $1 - 1/n^2$.

Now consider the case that S contains at least one planted column. Let U denote the planted biclique. Note that the expected value of x_i is $n/2 + r/2$ for $i \in U$. Thus, applying Hoeffding's inequality again we have

$$\Pr\left(\left|\sum_{i \in S} x_i - n/2 \cdot |S| - r/2 \cdot |S \cap U|\right| \leq -r/4\right) \leq \exp\left(-\frac{2r^2/4^2}{n \cdot |S| - r \cdot |S \cap U|}\right) \leq 1/n^2.$$

Thus, $\sum_{i \in S} x_i$ will be greater than $n/2 \cdot |S| + r/2 \cdot |S \cap U| - r/4 \geq n/2 \cdot |S| + r/4$ with probability at least $1 - 1/n^2$. A union bound implies that if there is not a planted clique, with high probability the sum of each group will be smaller than $n/2 \cdot |S| + r/4$ simultaneously. Otherwise with high probability we can find a group whose sum is larger than $n/2 \cdot |S| + r/4$. Thus our algorithm can output the correct answer with high probability. ■

Considering lower bounds on the the query complexity in the $\mathbb{F}_2$ sketching model (and hence the edge probe model), we obtain a better lower bound for the XOR version of the BPC game.

Theorem 16 *Any protocol solving the XOR version of the* BPC *game with constant success probability must communicate $\Omega(n^2/(rs))$ bits, and hence, $\Omega(n^2/(rs))$ queries are required to solve the* BPC *problem in the $\mathbb{F}_2$ sketching model.*

Proof We again use Lemma 13 to randomly partition a complete bipartite graph. Now we consider the two-player UNIQUE DISJOINTNESS game with input length $\lfloor n/r \rfloor \cdot \lfloor n/s \rfloor$. For each edge in an $r \times s$ biclique we uniformly randomly assign a color among 4 colors. Then we consider the bitstrings Alice and Bob hold, and construct graphs G_1 and G_2 as follows:

- If Alice has a 0, add edges with color 1 or 3 in the corresponding biclique to G_1.
- If Alice has a 1, add edges with color 1 or 2 in the corresponding biclique to G_1.
- If Bob has a 0, add edges with color 1 or 4 in the corresponding biclique to G_2.
- If Bob has a 1, add edges with color 3 or 4 in the corresponding biclique to G_2.

Finally, we construct graph $G = G_1 \oplus G_2$, namely, an edge occurs in G if and only if it occurs in exactly one of G_1 and G_2. It can be verified that if Alice and Bob both have a 1 on the same position, G will contain the corresponding biclique. Otherwise G will randomly contain each edge with probability $1/2$. Therefore, we finish the reduction and obtain an $\Omega(n^2/rs)$ lower bound. ■

B. Lower Bound for Finding a Planted Biclique

We consider the FindBPC game, where there may be a planted $r \times s$ biclique in an $n \times n$ bipartite graph, and the goal is to output all vertices of the biclique if it exists. Considering the case when $r = s = k$, the algorithm in Section 2.2 uses $\widetilde{O}(n/k)$ queries to solve the FindBPC problem in the Mv model. We provide a nearly-matching lower bound, showing that $\widetilde{\Omega}(n/k)$ queries are necessary. In fact, we can use a similar strategy to obtain both a communication lower bound for the FindBPC game and a query complexity lower bound for the FindBPC problem in the Mv model. We combine both results in the following theorem.

Theorem 17 *Let r and s be parameters that satisfy $3 \log n \leq r \leq s \leq n/2$.*

- *Any r-player protocol that solves the* FindBPC *game with constant success probability must communicate $\Omega(n)$ bits.*
- *Any algorithm that solves the* FindBPC *problem with constant success probability must use $\Omega(n/(r \log n))$ queries in the* Mv *model.*

Proof For the first part of the theorem, we reduce the FindBPC game to a "promise" variant of the PE game defined in Section 4. We refer to this variant as FindPE, where the parameter V is always set to one, but the players must output the index of the coordinate that is all ones (which is promised to exist when $V = 1$). For consistency with the FindBPC formulation, we let r denote the number of players and n denote the length of the vectors given to each player. We assume that $r \geq 3 \log n$ so that with high probability the only all ones coordinate is the planted coordinate.

First, note that the communication lower bound for the original PE game implies a lower bound for FindPE. To see this, we show how a FindPE protocol can solve the PE game with a negligible increase in communication. Given a PE instance, the players run the FindPE algorithm (even in the case of $V = 0$, as long as the protocol aborts if it uses more communication than it would on a $V = 1$ instance). If it outputs the index of a column, the players can sample $O(1)$ bits from this column to determine if it is all ones or random. If the FindPE algorithm outputs anything else, then we know that $V = 0$. This requires only $O(1)$ extra bits to succeed with constant probability, implying FindPE requires $\Omega(n)$ bits of communication.

Now we explain the connection to $\mathsf{FindBPC}$. We begin by constructing a random $n \times n$ matrix. We choose $n - r$ rows uniformly at random, and we independently sample the entries of these rows from $\texttt{Bernoulli}(1/2)$. For the remaining r rows, we assign one row to each of the r players. Among the n entries of the rows, we choose $s - 1$ at random, and set all entries to be one in each of these chosen rows (e.g., we plant $s - 1$ all ones columns in the $r \times n$ submatrix). Overall, we have defined the whole matrix except for $n - s + 1$ entries in each of the r rows. By using public randomness, we can assume that the $n^2 - r(n - s + 1)$ entries are known to all players.

We embed an instance of the r-player FindPE game in the unset entries, where each player has an input vector of size $n - s + 1$. Since FindPE is a promise variant, we are guaranteed that one of the coordinates is one in all r vectors. In particular, the full $n \times n$ matrix corresponds to the adjacency matrix of a bipartite graph with an $r \times s$ planted biclique (e.g., $r \times s$ all ones submatrix). Using this construction, the players can then execute a protocol for $\mathsf{FindBPC}$. By doing so, they reveal the location of the all ones coordinate from the FindPE instance. Therefore, since $s \leq n/2$, we have that $n - s + 1 = \Omega(n)$, and the players must communicate $\Omega(n)$ bits, which provides the desired lower bound for solving the $\mathsf{FindBPC}$ game.

Moving on to the second part of the theorem, we can also use the same construction to prove a lower bound on the query complexity in the Mv model. The r players build the $n \times n$ matrix in the same way as before, and the connection to the FindPE game is also the same. The difference is that they will now use a protocol for $\mathsf{FindBPC}$ that we derive from a query algorithm in the Mv model. Recall that the inputs of the r players correspond to an $r \times n$ submatrix (and the rest of the matrix is known to all the players). Therefore, each query in the Mv model can be simulated by communicating $O(r \log n)$ bits because the players simply need to evaluate the matrix-vector product on the $r \times n$ submatrix (each player handles one row). If the query algorithm uses q queries to solve the $\mathsf{FindBPC}$ problem, then this gives rise to a protocol for this construction that solves the $\mathsf{FindBPC}$ game (and hence the FindPE game) by communicating $q \cdot O(r \log n)$ bits. Thus, $q = \Omega(n/(r \log n))$ queries are needed to solve the $\mathsf{FindBPC}$ problem in the Mv model. ■

C. Semi-Random Planted Clique

For the SRPC problem, since the adversary only removes edges outside the planted clique, we can use the existing edge-probe upper bound (Theorem 1, (Rácz and Schiffer, 2020)) to obtain the following: suppose $k \geq (2 + \varepsilon) \log n$ for some constant $\varepsilon > 0$, then there exists an algorithm which can distinguish H_0 and H_1 using $\widetilde{O}(n^2/k^2)$ edge-probe queries (see Section 2.2; the algorithm is the same as the standard planted clique problem). We also provide a nearly matching lower bound for the corresponding communication game. The key observation is that we can reduce to the 2-player UNIQUE DISJOINTNESS game (instead of k^2 players) because we now have more flexibility to remove edges that are not in the planted clique.

Theorem 18 *Let $k = n^\gamma$ for any $\gamma \in (0, 1/2)$. Any 2-player protocol that solves the* SRPC *game with constant success probability needs to communicate $\Omega(n^2/k^2)$ bits. Hence, $\Omega(n^2/(k^2 \log n))$ queries are required in the general linear sketching model to solve the* SRPC *problem.*

Proof By Lemma 1, we can partition the complete graph K_n into $\Theta(n^2/k^2)$ edge-disjoint cliques each with size $\Theta(k)$. Then we randomly color each edge in these cliques with red or blue with equal probability. Now we consider the 2-player UNIQUE DISJOINTNESS game with input strings of length $\Theta(n^2/k^2)$. Construct a graph G as follows: for each bit, let G contain the red edges in the clique if Alice has a 1, and let G contain the blue edges in the clique if Bob has a 1. For those edges outside the cliques, each of them occurs with probability $1/2$. If there is at most a single 1 in every position, then G can be viewed as an instance under H_0. If there is a unique position such that both Alice and Bob have a 1, then G can be viewed as an instance under H_1. Thus we reduce the 2-player set UNIQUE DISJOINTNESS game (Section 2.3) to our semi-random planted clique, and therefore we get an $\Omega(n^2/k^2)$ lower bound for the SRPC game. For the query lower bound, we can simulate a single linear sketch query with $O(\log n)$ bits of communication, which implies that any algorithm succeeding with constant success probability must use $\Omega(n^2/(k^2 \log n))$ queries. ∎

D. Promise Planted Clique

Recall that the PPC problem is a promise variant of the planted clique problem. Here, there is a set S of $\Theta(n^2/k^2)$ possible subsets of vertices that may contain the clique. This information is known beforehand, and the goal is determine whether the graph is random or a k-clique has been planted in one of the subgraphs in S. We provide nearly matching upper and lower bounds.

Theorem 19 *If $|S| = \Theta(n^2/k^2)$ for the* PPC *problem, then $\widetilde{\Theta}(n^2/k^4)$ queries are sufficient and necessary for constant success probability in the general linear sketching and* $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ *models.*

Proof For the lower bound, observe that the proof of Theorem 11 already uses a set S of $\Theta(n^2/k^2)$ possible subsets of vertices that may contain the clique (via the graph decomposition result Lemma 1 and Remark 2). Hence, we can prove a lower bound using the $\Theta(k^2)$-player version of the PE game as before, which requires $\Omega(n^2/k^2)$ bits of communication. We can simulate each linear sketching or $\mathsf{u}^\mathsf{T}\mathsf{M}\mathsf{v}$ query with $O(k^2 \log n)$ bits since there are $\Theta(k^2)$ players. Thus, $\Omega(n^2/(k^4 \log n))$ queries are necessary.

For the upper bound, we sketch the idea of a randomized algorithm using the knowledge of S. First, randomly choose a subset S' of $\frac{k^2}{81 \log n}$ subgraphs uniformly from S. The subgraphs in S' contain a total of $m = \frac{k^2}{81 \log n} \cdot \binom{k}{2} = \Theta(k^4/\log n)$ edges. We let a_i be 1 if the i-th edge exists and be 0 if it does not exist. Then, if S' does not contain the planted clique, by Hoeffding's inequality, we have

$$\Pr\left(\sum_{i=1}^{m} a_i - m/2 \geq k^2/9\right) \leq \exp(-\frac{k^4}{81m}) \leq 1/n^2.$$

If S' contains the planted clique, we have

$$\Pr\left(\sum_{i=1}^{m} a_i - \binom{k}{2} - (m - \binom{k}{2})/2 \leq -k^2/9\right) \leq \exp\left(-\frac{k^4}{81(m - \binom{k}{2})}\right) \leq 1/n^2.$$

Note that $\binom{k}{2} + (m - \binom{k}{2})/2 - k^2/9 > m/2 + k^2/9$. As a consequence, we can identify whether there is a planted clique in S' by counting the edges in S' using one $\mathsf{u^T Mv}$ query. Repeating this process $\Theta(n^2/k^2)/\Theta(k^2/\log n) = \widetilde{\Theta}(n^2/k^4)$ times, we have that a union bound ensures that one of the subsets $S' \subseteq S$ contains the planted clique with high probability if there is one in the graph. ■

E. Hidden Hubs

Our techniques give an $\Omega(n^2/k^2)$ communication lower bound for a corresponding k^2-player game, which implies a lower bound of $\widetilde{\Omega}(n^2/k^4)$ general linear sketching queries for the HH problem (Kannan and Vempala, 2017).

Theorem 20 *Suppose $\sigma_1 \leq \sigma_0 \leq c\sigma_1$ for some constant $c > 0$. Any algorithm that solves the HH game with constant success probability requires $\widetilde{\Omega}(n^2/k^4)$ queries in the general linear sketching and $\mathsf{u^T Mv}$ models.*

Proof We sketch the slight modification of the proof of Theorem 11 for the HH problem (which in turn needs a slight modification of the proof of Theorem 10). First, we let $w = \lfloor n/k \rfloor$, and we randomly choose w disjoint subsets $R_1, \ldots, R_w$ of k rows each (discard the remaining rows if k does not divide n). Then, for row r in set R_i, we again randomly choose w disjoint sets of k entries $T_{r1}, \ldots, T_{rw}$ and let $U_{ij} = \bigcup_{r \in R_i} T_{rj}$. In particular, we have that $|U_{ij}| = k^2$.

Now we consider a k^2-player communication game that is a modified "Gaussian version" of the PE game from Section 4, where the players have inputs of size w^2 each. The modification is that instead of binary variables, we consider the distributions $\mu_0' = \mathcal{N}(0, \sigma_0^2)$ and $\mu_1' = \mathcal{N}(0, \sigma_1^2)$, where σ_0 and σ_1 are the parameters of the HH problem. In this way, we construct matrix $\boldsymbol{A}$ so that the entries in U_{ij} for $i, j \in [k]$ are the values that the k^2 players hold. For the entries not in any U_{ij} we generate them according to μ_0'. Thus we reduce to the HH game from this k^2-player communication game. The key step of the proof of Theorem 10 provides a lower bound of $\Omega(1)$ for the information complexity for each coordinate. We can prove the same bound, again by Theorem 8. We continue to use $\beta = 1$, e.g., the standard data processing inequality.

The coordinates in the modified game are drawn from the distributions μ_0' and μ_1'. Then, we let $f_0(x) = \frac{1}{\sqrt{2\pi}\sigma_0}\exp(-\frac{x^2}{2\sigma_0^2})$ and $f_1(x) = \frac{1}{\sqrt{2\pi}\sigma_1}\exp(-\frac{x^2}{2\sigma_1^2})$ be the probability density function of μ_0' and μ_1', respectively. Thus $\frac{f_1(x)}{f_0(x)} = \frac{\sigma_0}{\sigma_1} \cdot \exp(-\frac{x^2}{2}(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_0^2}))$. Note that $\frac{\sigma_0}{\sigma_1} \leq c$, and the exponential term is less than 1 since $\sigma_1 \leq \sigma_0$, and so $\mu_1' \leq c\mu_0'$. Thus, the direct sum argument holds (Lemma 9), and we also have an $\Omega(1)$ lower bound on the information complexity for each coordinate. Hence, the overall proof strategy implies a lower bound of $\Omega(w^2) = \Omega(n^2/k^2)$ for the k^2-player game.

For the query lower bound, we note that each general linear sketching or $\mathsf{u^T Mv}$ query can be simulated with $O(k^2 \log n)$ bits of communication as there are k^2 players. Therefore, this shows that $\widetilde{\Omega}(n^2/k^4)$ queries are necessary to solve the HH problem. ■

The algorithm from (Kannan and Vempala, 2017) can be simulated in the query models that we consider. The main idea is to randomly sample $\widetilde{\Theta}(n/k)$ entries in each row. Then, using these, it is known how to distinguish the two hypotheses from the HH problem as long as $\sigma_1^2 > 2\sigma_0^2$. Hence, in the edge-probe model, we can sample the entries with $\widetilde{\Theta}(n^2/k)$ queries. The same upper bound trivially holds for the linear sketching and $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ models. In the Mv model, we can sample with $\widetilde{\Theta}(n/k)$ queries.

For the HH problem, we leave open the question of tightening the bounds. We also note that there is a bound of $\widetilde{\Theta}(n^2/k^2)$ in the statistical query model, depending on σ_0 and σ_1 (Kannan and Vempala, 2017).

F. Sparse Principal Component Analysis

We consider the sub-Gaussian version of the SPCA problem, using elements of a known reduction from the PC problem (Berthet and Rigollet, 2013). The empirical variance of t vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \in \mathbb{R}^d$ in direction $\boldsymbol{v}$ is defined as

$$\widehat{\mathsf{var}}(\boldsymbol{v}) = \frac{1}{t}\sum_{i=1}^{t}(\boldsymbol{v}^\top \boldsymbol{X}_i)^2.$$

Let θ and k be parameters, and let $\zeta \in (0,1)$ be a fixed, small constant. We let $\mathcal{D}_0$ denote the set of product distributions over t i.i.d. vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \in \mathbb{R}^d$ such that for all unit vectors $\boldsymbol{v}$ we have

$$\Pr\left[|\widehat{\mathsf{var}}(\boldsymbol{v}) - 1| > 4\sqrt{\frac{\log(2/\zeta)}{d}} + 4\frac{\log(2/\zeta)}{d}\right] \le \zeta. \tag{F.1}$$

In other words, $\mathcal{D}_0 := \{\mathbf{P}_0 \mid \textit{Eq. (F.1)} \text{ holds}\}$, and $\mathcal{D}_0$ contains, e.g., isotropic distributions.

We let $\mathcal{D}_1^{k,\theta}$ denote the set of product distributions over t i.i.d. vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \in \mathbb{R}^d$ such that for all unit vectors $\boldsymbol{v}$ with at most k nonzero entries ($\|\boldsymbol{v}\|_0 \le k$), we have

$$\Pr\left[(\widehat{\mathsf{var}}(\boldsymbol{v}) - (1+\theta)) < -2\sqrt{\frac{\theta k \log(2/\zeta)}{d}} - 4\frac{\log(2/\zeta)}{d}\right] \le \zeta. \tag{F.2}$$

Similarly, $\mathcal{D}_1^{k,\theta} := \{\mathbf{P}_1 \mid \textit{Eq. (F.2)} \text{ holds}\}$.

SCDC problem. Define two hypotheses to test; the inputs $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t$ are drawn from $\mathbf{P}$ such that

$$\mathcal{H}_0 : \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \sim \mathbf{P}_0 \in \mathcal{D}_0 \qquad \text{vs.} \qquad \mathcal{H}_1 : \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t \sim \mathbf{P}_1 \in \mathcal{D}_1^{k,\theta}.$$

Our goal is to distinguish between the possible family of distributions that $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_t$ is sampled from. The motivation for this problem is that $\mathcal{D}_0$ contains $\mathcal{N}(0, \boldsymbol{I}_d)$ and $\mathcal{D}_1$ contains $\mathcal{N}(0, \boldsymbol{I}_d + \theta \boldsymbol{u}\boldsymbol{u}^\mathrm{T})$ when $\boldsymbol{u}$ is a k-sparse unit vector. In other words, this hypothesis testing problem is a generalization of the spiked covariance matrix detection problem. Intuitively, θ corresponds to the signal strength, and k corresponds to the sparsity of the unknown "high variance" direction of the alternate hypothesis distribution. Nonetheless, many known algorithms for the spiked covariance problem also hold for this more general problem (Berthet and Rigollet, 2013). We note that reductions between planted clique and the spiked covariance version are known (Gao et al., 2017), but we do not know how to implement these reductions efficiently in our query or communication models. Instead, we describe a reduction for the above formulation of the problem.

Let $\mathbb{G}_m$ denote the set of graphs on m vertices. With the following reduction, we obtain our main theorem for SCDC. We provide the key details and verify that the reduction holds in our query models, and we refer to (Berthet and Rigollet, 2013) for the full details about the distributional relationships.

Reduction from PC (Berthet and Rigollet, 2013). For any $\gamma \in (0,1)$ and a fixed tolerance $\delta \in (0, 1/3)$ (e.g., $\delta = 5\%$), given $(d,t,k) \in R_\gamma$, where

$$R_\gamma = R_0 \cap \{k \geq t^\gamma\} \cap \{t < d\}$$

and

$$R_0 = \left\{ (d,t,k) \in \mathbb{N}_+^3 : 15\sqrt{\frac{k\log(6ed/\delta)}{t}} \leq 1, k \leq d^{0.49} \right\}$$

where 0.49 can be replaced by any constant $C < 0.5$, the randomized reduction $\mathtt{bl}_{d,t,k,m,\kappa} : \mathbb{G}_{2m} \mapsto \mathbb{R}^{d\times t}$ is a procedure defined as follows, where m, κ are positive integers such that $t \leq m < d$ and $k \leq \kappa \leq m$.

For a $(2m)$-vertex graph $G = (V, E)$, which is an instance of PC problem with a potential clique of size κ, we first choose m uniformly random vertices $V_{\mathtt{left}}$ among $2m$ vertices, and then choose t uniformly random vertices $V_{\mathtt{right}}$ among the remaining m vertices that are not in $V_{\mathtt{left}}$. Make it a bipartite graph by restricting its edges in $E \cap \{V_{\mathtt{left}} \times V_{\mathtt{right}}\}$. Then, add $(d-m)$ new vertices to $V_{\mathtt{left}}$ and place an edge between every old vertex in $V_{\mathtt{right}}$ and each new vertex in $V_{\mathtt{left}}$ independently with probability $1/2$. We relabel the left (resp. right) vertices by a random permutation of $\{1, 2, \dots, d\}$ (resp. $\{1, 2, \dots, t\}$). Let $G' = (\{1, 2, \dots, d\} \times \{1, 2, \dots, t\}, E')$ denote the resulting bipartite graph, and let $\boldsymbol{B}$ denote the $d \times t$ adjacency matrix of G'. Also, let $\eta_1, \eta_2, \dots, \eta_t \in \{-1, 1\}$ be t i.i.d. Rademacher random variables that are independent of all previous random variables. Define

$$\boldsymbol{X}_i^{(G)} = \eta_i(2\boldsymbol{B}_i - 1) \in \{-1, 1\}^d, \tag{F.3}$$

where $\boldsymbol{B}_i$ is the i-th column of $\boldsymbol{B}$. By all above steps, we finish the reduction

$$\mathtt{bl}_{d,t,k,m,\kappa}(G) = (\boldsymbol{X}_1^{(G)}, \boldsymbol{X}_2^{(G)}, \dots, \boldsymbol{X}_t^{(G)}) \in \mathbb{R}^{d\times t}.$$

Now that we have described the reduction bl, we explain how to simulate the algorithm. In the query models, we will use the public randomness of the algorithm for this randomized reduction. We next state a simple, yet general, result, which identifies matrix operations that can be simulated in the query models.

Lemma 21 *Let $\boldsymbol{X}$ be a matrix that is a transformation of a matrix $\boldsymbol{Y}$ after applying one or more of the following operations:*

(i) insert a row or column into $\boldsymbol{Y}$,

(ii) permute the rows or columns of $\boldsymbol{Y}$, or

(iii) for field elements a, b, apply $\phi(y) = ay + b$ to all entries in a row or column of $\boldsymbol{Y}$, replacing each entry y with $\phi(y)$.

Then, for any query to $\boldsymbol{X}$ in the $\mathsf{u}^{\mathsf{T}}\mathsf{Mv}$*,* Mv*, edge-probe, or linear sketching models, there is a deterministic way to perform a single query to $\boldsymbol{Y}$ and simulate the original query to $\boldsymbol{X}$.*

Proof We first explain the simulation for the general linear sketching model; at the end, we mention the differences for the other models. Say that the query algorithm wants to learn $\boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{X})$. Our goal is to design a vector $\boldsymbol{w}$ such that $\boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{X})$ can be computed directly from $\boldsymbol{w}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Y})$.

For (i), assume $\boldsymbol{X}$ is $\boldsymbol{Y}$ after inserting a row (the column case is analogous). The idea is that we can compute the contribution from the insertion and add this after querying $\boldsymbol{Y}$. More precisely, let $\boldsymbol{Z}$ be a matrix with a single non-zero row in the position of the inserted row to $\boldsymbol{Y}$ with the same entries. Let $\boldsymbol{w}$ be the vector obtained from $\boldsymbol{v}$ by deleting the positions in $\boldsymbol{v}$ corresponding to the inserted row. As the algorithm knows $\boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Z})$, we have that $\boldsymbol{w}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Y})$ suffices to compute $\boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{X}) = \boldsymbol{w}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Y}) + \boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Z})$.

For (ii), we can permute the entries of $\boldsymbol{v}$ to obtain $\boldsymbol{w}$. Precisely, we can determine the permutation matrix $\mathbf{P}$ such that $\boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{X}) = \boldsymbol{v}^{\mathrm{T}}\mathbf{P}\mathsf{vec}(\boldsymbol{Y}) = (\mathbf{P}^{\mathrm{T}}\boldsymbol{v})^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Y})$ and use the query vector $\boldsymbol{w} = \mathbf{P}^{\mathrm{T}}\boldsymbol{v}$.

Finally, for (iii), first multiply entries in $\boldsymbol{v}$ by a for each position corresponding to the modified row or column. Let the resulting query vector be $\boldsymbol{v}_a$. Then, calculate the sum z of entries in $\boldsymbol{v}_a$ that overlap with the positions in the modified row or column, and add zb to the result of the $\boldsymbol{v}_a$ query. Overall, by construction we have that $\boldsymbol{v}_a^{\mathrm{T}}\mathsf{vec}(\boldsymbol{Y}) + zb = \boldsymbol{v}^{\mathrm{T}}\mathsf{vec}(\boldsymbol{X})$.

If multiple of these operations are used to transform $\boldsymbol{Y}$ into $\boldsymbol{X}$, then we can iteratively apply the above procedures, i.e., we can simulate any query with a single other query.

We have described the simulation for the general linear sketching model. The same strategy works for the $\mathbb{F}_2$ sketching model. For the edge-probe model, note that a single entry in $\boldsymbol{X}$ depends on only a single entry of $\boldsymbol{Y}$ even after any of the three allowed operations. In the $\mathsf{u}^{\mathsf{T}}\mathsf{Mv}$ model, the difference for (iii) is that we can rescale rows via the left query vector (or columns via the right query vector) by multiplying the relevant entry by a. For the Mv model, the main difference is that when we modify a row of $\boldsymbol{Y}$, we have to compute the contribution to $\boldsymbol{X}\boldsymbol{v}$ and add this to obtain the correct query, similar to (iii) above. Columns in (iii) can be handled by updating the query vector. ■

Proposition 22 *There exists a constant $\delta \in (0, 1)$ such that the following holds. If there is a query algorithm $\mathcal{A}$ for the* SCDC *problem that makes q queries and has success probability $1 - \delta$, then there is a query algorithm $\mathcal{A}'$ for the* PC *problem that makes q queries and has success probability $1 - \Theta(\delta)$. This holds for algorithms in the* $\mathsf{u}^{\mathsf{T}}\mathsf{Mv}$*,* Mv*, edge-probe, and general linear sketching models.*

Proof Let $\mathcal{A}$ be an algorithm for SCDC in a query model that we consider. Given an instance of planted clique, we have a graph G. We can apply the randomized reduction to produce a $d \times t$ matrix $\mathtt{bl}(G)$ composed of columns $(\boldsymbol{X}_1^{(G)}, \boldsymbol{X}_2^{(G)}, \ldots, \boldsymbol{X}_t^{(G)})$. We claim that we can use Lemma 21 to take the algorithm $\mathcal{A}$ for SCDC and derive an algorithm $\mathcal{A}'$ for the PC problem on G via the procedure $\mathtt{bl}(G)$. Here, $\boldsymbol{Y}$ corresponds to the adjacency matrix of G and $\boldsymbol{X}$ corresponds to $\mathtt{bl}(G)$, the SCDC input. First, even though $\mathtt{bl}$ is a randomized procedure, we have used the randomness of the query algorithm, and hence we know the transformation. The linear transformation in Eq. (F.3) is covered by part (iii) of Lemma 21. Then, in the procedure $\mathtt{bl}(G)$, we are sampling vertices, permuting vertex labels, and adding edges connected to new vertices in the graph, which are covered by the three parts of the lemma.

It remains to show that if $\mathcal{A}$ has constant success probability for SCDC, then $\mathcal{A}'$ has constant success probability for PC. We sketch this argument, since it follows from Theorem 7 and Lemma 8 in (Berthet and Rigollet, 2013). They work in a more general model, where they consider families of statistical tests $\psi = \{\psi_{d,t,k}\}$ for SCDC and tests $\xi = \{\xi_{m,\kappa}\}$ for PC on $2m$-vertex graphs. In our models, this corresponds to the query algorithms by considering a test to be a query algorithm that queries the matrix and then outputs a binary variable, where 0 corresponds to the null hypothesis, and 1 corresponds to the alternate hypothesis.

To state their result, fix $\alpha \in [1,2), \gamma \in (0, \frac{1}{4-\alpha})$, and define $a = 2\gamma, b = 1-(2-\alpha)\gamma$. Their result says that for any $\tau > 0$, there exists a constant $L > 0$, such that the following holds. For $(d,t,k) \in R_\gamma$, there exist κ, m such that $(2m)^{a/2} \le \tau\kappa \le (2m)^{b/2}$, a random transformation $\mathtt{bl} = \{\mathtt{bl}_{d,t,k,m,\kappa}\}$, $\mathtt{bl}_{d,t,k,m,\kappa} : \mathbb{G}_{2m} \mapsto \mathbb{R}^{d\times t}$, and distributions $\mathbf{P}_0 \in \mathcal{D}_0$, $\mathbf{P}_1 \in \mathcal{D}_1^{k,L\theta_\alpha}$ such that the following holds. For shorthand, we use the notation $\mathbf{P}_0(f=1)$ for a test f to mean the probability that the output is 1 when an instance is drawn from $\mathbf{P}_0$, i.e., the error probability of NO instances (and analogously for $\mathbf{P}_1(f=0)$). We also use $\vee$ to mean $\max$. Then, their Theorem 7 says that there exists a constant δ such that for any family of tests $\psi = \{\psi_{d,t,k}\}$, we have

$$\mathbf{P}_0^{\otimes t}(\psi_{d,t,k}=1) \vee \mathbf{P}_1^{\otimes t}(\psi_{d,t,k}=0) \ge \mathbf{P}_0^{(G)}(\xi_{m,\kappa}(G)=1) \vee \mathbf{P}_1^{(G)}(\xi_{m,\kappa}(G)=0) - \frac{\delta}{5},$$

where

$$\xi_{m,\kappa} = \psi_{d,t,k} \circ \mathtt{bl}_{d,t,k,m,\kappa} \text{ and } \theta_\alpha = \sqrt{\frac{k^\alpha}{t}}.$$

■

Using Proposition 22, we see that if we have any query algorithm for SCDC with error probability at most δ', then we can derive a query algorithm for PC with error probability at most $\delta' + \delta/5$. In other words, a constant success probability query algorithm for SCDC implies one for PC with the same query complexity. This allows us to derive the query complexity lower bounds in the following theorem.

Theorem 23 *Given any $\alpha \in [1,2)$, $\gamma \in (0, \frac{1}{4-\alpha})$ and any $(d,t,k) \in R_\gamma$, for the SCDC problem with input matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_t]$ when $d = \Theta(t)$, $k = \Theta(t^\gamma)$ and $\theta = \Theta\left(\sqrt{\frac{k^\alpha}{t}}\right)$, any algorithm that succeeds with constant success probability requires*

- $\widetilde{\Omega}\left(\frac{k^4}{t^2\theta^4}\right)$ *queries in the general linear sketching or* $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ *models,*
- $\widetilde{\Omega}\left(\frac{k^4}{t^3\theta^4}\right)$ *queries in the* Mv *model,*
- $\Omega(k^2/\theta^2)$ *queries in the edge-probe or* $\mathbb{F}_2$ *sketching models.*

Proof By using the reduction above and applying Proposition 22, we can use an algorithm for SCDC to solve the PC problem on m-vertex graphs. Translating between the parameters via the reduction, plugging in $k = \Theta(t^\gamma)$, $\kappa = O\left(t^{\frac{1}{2} - \frac{(2-\alpha)\gamma}{2}}\right)$, $\theta = \Theta\left(\sqrt{\frac{k^\alpha}{t}}\right)$ and $t = \Theta(m) = \Theta(d)$, yields the following query lower bounds:

- Using the lower bound of $\widetilde{\Omega}(m^2/\kappa^4)$ queries for the PC problem in the general linear sketching and $\mathsf{u}^\mathsf{T}\mathsf{Mv}$ models when $\kappa = o(\sqrt{m})$ from Corollary 12, we get a query lower bound for SCDC of

$$\widetilde{\Omega}\left(\frac{m^2}{\kappa^4}\right) = \widetilde{\Omega}\left(\frac{m^2}{t^{2-2(2-\alpha)\gamma}}\right) = \widetilde{\Omega}\left(t^{2(2-\alpha)\gamma}\right) = \widetilde{\Omega}\left(\frac{t^{4\gamma}}{t^{2\alpha\gamma}}\right) = \widetilde{\Omega}\left(\frac{k^4}{k^{2\alpha}}\right) = \widetilde{\Omega}\left(\frac{k^4}{t^2\theta^4}\right),$$

  since $m^2 = \Theta(t^2)$ and $t^\gamma = \Theta(k)$, and in the final equality, we use that $\frac{t^2}{k^{2\alpha}} = \Theta\left(\frac{1}{\theta^4}\right)$.

- Using the lower bound of $\widetilde{\Omega}(m/\kappa^4)$ queries for the PC problem in the Mv model when $\kappa = o(\sqrt{m})$ (direct corollary of Corollary 12), we get a query lower bound for SCDC of

$$\widetilde{\Omega}\left(\frac{m}{\kappa^4}\right) = \widetilde{\Omega}\left(\frac{mt}{t\kappa^4}\right) = \widetilde{\Omega}\left(\frac{m^2}{t\kappa^4}\right) = \widetilde{\Omega}\left(\frac{k^4}{t^3\theta^4}\right),$$

  using the above calculations and the fact that $m = \Theta(t)$.

- Using the lower bound of $\Omega(m^2/\kappa^2)$ for the PC problem in the edge-probe and $\mathbb{F}_2$ sketching models when $\kappa = o(\sqrt{m})$ from Corollary 5, we get a query lower bound for SCDC of

$$\Omega\left(\frac{m^2}{\kappa^2}\right) = \Omega\left(\frac{t^2}{\kappa^2}\right) = \Omega\left(\frac{t^2}{t^{1-(2-\alpha)\gamma}}\right) = \Omega\left(\frac{t \cdot t^{2\gamma}}{t^{\alpha\gamma}}\right) = \Omega\left(\frac{t \cdot k^2}{k^\alpha}\right) = \Omega\left(\frac{k^2}{\theta^2}\right),$$

  where the final equality uses that $\frac{t}{k^\alpha} = \Theta\left(\frac{1}{\theta^2}\right)$.

■