A Unified Approach to Memory-Sample Tradeoffs for Detecting Planted Structures

Sumegha Garg* Jabari Hastings† Chirag Pabbaraju‡ Vatsal Sharan§

Abstract

We present a unified framework for proving memory lower bounds for multi-pass streaming algorithms that detect planted structures. Planted structures — such as cliques or bicliques in graphs, and sparse signals in high-dimensional data — arise in numerous applications, and our framework yields multi-pass memory lower bounds for many such fundamental settings. We show memory lower bounds for the planted k-biclique detection problem in random bipartite graphs and for detecting sparse Gaussian means. We also show the first memory-sample tradeoffs for the sparse principal component analysis (PCA) problem in the spiked covariance model. For all these problems to which we apply our unified framework, we obtain bounds which are nearly tight in the low, $O(\log n)$ memory regime. We also leverage our bounds to establish new multi-pass streaming lower bounds, in the vertex arrival model, for two well-studied graph streaming problems: approximating the size of the largest biclique and approximating the maximum density of bounded-size subgraphs.

To show these bounds, we study a general distinguishing problem over matrices, where the goal is to distinguish a null distribution from one that plants an *outlier* distribution over a random submatrix. Our analysis builds on a new distributed data processing inequality that provides sufficient conditions for memory hardness in terms of the likelihood ratio between the *averaged* planted and null distributions. This result generalizes the inequality of [Braverman et al., STOC 2016] and may be of independent interest. The inequality enables us to measure information cost under the null distribution – a key step for applying subsequent direct-sum-type arguments and incorporating the multi-pass information cost framework of [Braverman et al., STOC 2024]. Finally, to instantiate our framework in concrete settings, we derive bounds on the likelihood ratio between the planted and null distributions using careful truncations.

*Rutgers University. Email: sumegha.garg@rutgers.edu
†Stanford University. Email: jabarih@stanford.edu
‡Stanford University. Email: cpabbara@stanford.edu
§University of Southern California. Email: vsharan@usc.edu

Contents

1 Introduction

Many statistical estimation tasks involve discovering certain hidden structures in the data distribution. A well-known instance of this is the planted clique problem [Jer92, Kuč95], where one is given a random Erdős–Rényi graph $G(n, 1/2)$ (each edge exists with probability 1/2) but with a clique added on a uniformly randomly chosen subset of k vertices. The goal is to recover this planted clique. Several variants of this problem, such as finding the densest subgraph within a graph [CX16] or finding the presence of certain community structure in the graph [Abb18] share a similar "planted" flavor. Planted structures arise not only in combinatorial problems such as clique detection, but also in classical statistical settings – for instance, estimating the mean of a high-dimensional Gaussian when the mean vector is known to be k-sparse [BGM+16] or performing dimensionality reduction through sparse principal component analysis (PCA) [ZHT06]. Since modern settings often involve large amounts of high-dimensional data with many irrelevant attributes, problems with sparse planted structures capture key challenges in statistical estimation in such settings. Other examples include sparse linear regression [SD15], sub-matrix detection [MW15b] and testing almost k-wise independence [AAK+07].

These problems with planted structure also offer a fertile ground to understand average-case complexity, and the interaction of computational and statistical resources. In many of these settings, there is believed to be a gap between what is information-theoretically optimal, and what is possible under computational constraints. The planted clique problem and the sparse PCA problem are among the problems which have been central objects of study in this line of work. For the planted clique problem, if the clique size $k = \Omega(\sqrt{n})$ then polynomial-time algorithms are known for recovering the clique [AKS98, FK00, McS01]. It is widely conjectured that if $k < O(n^{1/2-\delta})$ for some $\delta > 0$, then no polynomial-time algorithms exist for approximately recovering (or detecting) the planted clique. Hardness of planted clique implies hardness of a number of problems with planted structure including testing almost k-wise independence [AAK+07], community detection [BB20], sub-matrix detection [MW15b], as well as sparse PCA [BR13]—pointing at its fundamental nature for understanding statistical-computational gaps and average-case complexity. Similarly, the sparse PCA problem which adds a sparsity constraint to the usual PCA problem (discussed further in Section 1.4) has played a central role in understanding computational-statistical tradeoffs in statistical settings. Its hardness has been studied from the perspective of sum of squares relaxations [MW15a, HKP+17], low-degree likelihood ratio tests [DKWB24], statistical query algorithms [BBH+21], robustness to adversarial perturbations [dKNS20], failure of approximate message passing [LKZ15, BMR20] and methods from statistical physics [LKZ17, AWZ23].

Our goal in this work is to understand statistical-computational gaps for detecting planted clique, sparse PCA and other problems with planted structures. We consider the streaming model of computation, where the algorithm gets one or more passes over an input drawn from some data-generating distribution. Here, the memory usage of the algorithm is the main metric of computational cost. The streaming model over stochastic inputs is widely studied [GM07, AMOP08, KMM12, KKS14, CMVW16, Raz18, SSV19, BGW20, BGL+24], and it captures many modern settings involving massive computation on large graphs or datasets. In addition to its practical relevance, investigating the role of memory in detecting planted structures offers a complementary vantage point to understand the computational hardness of statistical inference [Sha14, SVW16, DH24, MSSV24] and, as we show, also yields new streaming lower bounds for approximation problems on worst-case graphs.

In this work, we develop a general framework for proving memory hardness of detecting planted structures in data, and apply it to several canonical settings ranging from graph problems to learning tasks. Our first application establishes unconditional statistical-computational tradeoffs for the *planted biclique problem* – a bipartite generalization of the planted clique problem – previously studied by [FGR+17] in the context of statistical query hardness for planted clique detection. In this problem, the goal is to distinguish whether a uniformly random bipartite graph has a $(k \times k)$ biclique planted on a uniformly chosen set of vertices. The problem is at least as hard as the planted clique problem and has been used as a cryptographic primitive [ABW10]. Moreover, most known algorithms and bounds for the planted clique problem naturally extend to the bipartite version [AV11, FP16, KLP22, BKS23]. In the streaming model, at each time-step the algorithm observes a uniformly random left vertex together with its adjacency list. [FGR+17] studied the distributional version of the planted biclique problem defined on such adjacency-list vectors.

Problem 1.1. *Fix an integer k, $1 \le k \le n$, and a uniformly random subset of k indices $S \subseteq [n]$. The input distribution D_S on vectors $x \in \{0,1\}^n$ is defined as follows: with probability $1 - (k/n)$, x is uniform over $\{0,1\}^n$; and with probability k/n, x is such that its k coordinates from S are set to 1, and the remaining coordinates are uniform in $\{0,1\}$. Given m independent samples, the distributional planted k-biclique problem is to distinguish between samples drawn from D_S and samples drawn uniformly from $\{0,1\}^n$.*

We show that any p-pass streaming algorithm solving the distributional planted k-biclique problem with m samples requires

$$\Omega\left(\frac{n}{m} \cdot \frac{n^2}{p\, k^4 \log n}\right) \tag{1}$$

bits of memory. When $m = \Omega(n^3/k^4)$ – that is, when $\sqrt{nm} \gg k^2 m/n$ – a simple edge-counting algorithm using one pass and $O(\log n)$ memory suffices to distinguish the planted distribution from uniform. Hence, our memory-sample tradeoff is tight up to logarithmic factors in the low-memory regime. In the statistically feasible regime – when $k = \Theta(\log n)$ and $m = \tilde{O}(n)$ – any constant-pass streaming algorithm must use $\tilde{\Omega}(n^2)$ bits of memory. Without delving into tedious details, we show the same memory hardness for any multi-pass streaming algorithm that distinguishes between a random $G(m, n, 1/2)$ bipartite graph and one with an added planted $(k \times k)$ biclique. While this result is significant in its own right and requires new techniques, our main contribution is a *general framework* providing sufficient conditions on the underlying distributions to yield such memory-sample tradeoffs for detecting planted structures. This framework further allows us to generalize our lower bounds to detecting planted $(k \times k)$ bicliques in random $G(m, n, q)$ bipartite graphs for any $0 < q < 1/2$, which we discuss in more detail in Section 1.2.

1.1 Our general framework

Changing notation slightly, consider the planted biclique problem on a bipartite graph with n left vertices and d right vertices. The streaming algorithm observes n adjacency-list vectors in $\{0,1\}^d$, such that at k uniformly chosen time-steps, these vectors contain all 1s on a fixed subset of coordinates $S \subseteq [d]$. In our general framework for detecting planted structures, we retain the property that a fraction of the rows follow a planted distribution, but we additionally constrain the subset S to lie within a predefined "partition". This modification allows us to model a broader class of planted distributions, and we formalize this general setup below (see Figure 1 for an illustration). Given a vector x, we represent its projection to coordinates in S by x_S.

Problem 1.2 (General planted structure detection)**.** *Consider some n, d, $0 < k \leq n$ and $0 < t \leq d$. Let $\{T_r\}_{r \in [d/t]}$ be some partition of $[d]$, where $\forall r, |T_r| = t$. Let $\mu_0, \{\mu_\theta\}$ be distributions on t-dimensional vectors, and P be some distribution over θ. The goal is to distinguish between the following joint distributions on n such d-dimensional vectors $x^1, \ldots, x^n \in \mathcal{X}^d$:*

1. *D_0: $\forall i \in [n]$ and $\forall r \in [d/t]$, $x^i_{T_r}$ is drawn from μ_0.*
2. *D_1: Pick r uniformly from $[d/t]$. Pick set R uniformly from subsets of $[n]$ of size k. Pick $\theta \sim P$.*

$$\forall i \in [n] \text{ and } \forall r' \neq r, x^i_{T_{r'}} \sim \mu_0$$

(i.e. except for the chosen partition T_r, draw coordinates in all partitions from μ_0, for all datapoints).

$$\forall i \notin R, x^i_{T_r} \sim \mu_0$$

(i.e. for datapoints not in chosen set R, coordinates in all partitions are drawn similar to D_0).

$$\forall i \in R, x^i_{T_r} \sim \mu_\theta$$

(i.e. for datapoints in chosen set R, the coordinates in chosen partition T_r are drawn from μ_θ).

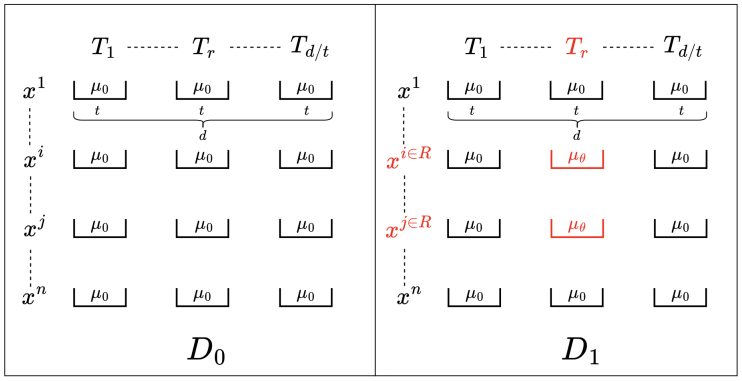

Figure 1: The distributions D_0 and D_1 from Problem 1.2. The partition $\{T_r\}_{r \in [d/t]}$ is shown to be over contiguous segments here only for convenience. In D_1, r is drawn uniformly from $[d/t]$, R is drawn uniformly from subsets of $[n]$ of size k, and θ is drawn from P. The planted structure is highlighted in red.

The above setup captures settings with sparsely planted structures on certain coordinates of the datapoints (through T_r), as well as scenarios where a subset of datapoints are outliers containing planted structure (through R). In addition to encompassing the planted biclique detection problem in random bipartite graphs $G(n, d, q)$ – where each edge across the partition appears independently with probability q – this framework also models canonical learning problems over Gaussian distributions. Let μ_0 be a product distribution over t-dimensional vectors with *i.i.d.* $\mathcal{N}(0, 1)$ entries, let P be the uniform distribution over k-sparse subsets $\theta \subseteq [t]$, and let μ_θ be the distribution obtained from μ_0 by shifting the mean to +1 on coordinates in θ. This yields the problem of detecting a mixture of a standard multivariate Gaussian and a sparse-mean Gaussian. While both the planted

biclique and sparse-mean Gaussian problems involve planted distributions that are *i.i.d.* over the selected coordinates, we use our framework to also model sparse PCA, where the planted distribution μ_θ introduces correlations among the selected coordinates – specifically, the projection of a datapoint onto θ is drawn from a Gaussian with a shifted covariance. Table 1 summarizes the specific parameters used for these applications.

Table 1: Different instantiations for Problem 1.2. We consider θ to be a uniformly random ℓ-sized subset of the chosen t-sized partition T_r.

Application	**Null Distribution** μ_0	**Planted Distribution** μ_θ	**Parameters** k **and** t**, where** $1 \le k \le n$ **and** $1 \le t \le d$
Planted biclique on random $G(n, d, q)$ graphs	$\mathrm{Ber}(q)^{\otimes t}$	Set coordinates in θ to be 1	$k = \ell$, $t = \tilde{\Theta}\left(\frac{\ell^2}{q}\right)$
Planted biclique with monotone adversaries	$\mathrm{Ber}(1/2)^{\otimes t}$	Set coordinates in θ to a fixed string in $\{0,1\}^\ell$	$k = \ell$, $t = \ell$
$(1-q)$: q mixture of a standard Gaussian and a sparse-mean Gaussian	$\mathcal{N}(0,1)^{\otimes t}$	Coordinates in θ are drawn from $\mathcal{N}(1,1)^{\otimes \ell}$	$k = qn$, $t = \tilde{\Theta}(d^{o(1)}\ell^2)$
ℓ-sparse PCA	$\mathcal{N}(0,1)^{\otimes t}$	Coordinates in θ are drawn from $\mathcal{N}(0, I_\ell + \alpha vv^T)$, for some small α and unit vector v	$k = n$, $t = \tilde{\Theta}(d^{0.01}\ell)$

Next we state our main theorem establishing memory-sample tradeoffs for the general planted structure detection problem.

Theorem 1.3 (Informal version of Theorem 4.3)**.** *Let* $\mu_1 = \mathbb{E}_{\theta\sim P}[\mu_\theta]$. *Suppose* $\mu_1(x) \le c\cdot\mu_0(x)\ \forall\ x \in \mathcal{X}^t$. *Then, any p-pass streaming algorithm that solves Problem 1.2 requires at least* $\Omega\left(\frac{nd}{p\cdot c\cdot k^2\cdot t}\right)$ *bits of memory.*

The above theorem provides a sufficient condition on the null and planted distributions, μ_0 and μ_θ respectively, for proving memory-hardness of detecting planted structures. Note that we must at least require the *distance* between μ_0 and the average planted distribution $\mathbb{E}_{\theta\sim P}[\mu_\theta]$ to be small; otherwise, a single sample would suffice to distinguish the two distributions. Our condition is both simple to state – as it depends only on the average planted distribution – and broadly applicable. However, ensuring it holds for the distributions used in our applications (listed in Table 1) requires careful truncation and modification. To prove the above theorem, we first establish a new, generalized distributed data processing inequality that is of independent interest (see Theorem 2.1 for a detailed statement). Theorem 1.3 then follows through multiple applications of direct-sum-type arguments – one over the partitions and another over the rows. Each step is nontrivial, as it crucially depends on the specific distributions used in the information complexity notions. We provide a detailed outline of our technical contributions in Section 2.

1.2 Applications to planted biclique detection and its variants

Firstly, we consider memory requirements for detecting planted bicliques. Let $G(m, n, q)$ denote the distribution over bipartite graphs with m left vertices and n right vertices, where each edge across

the partition is present independently with probability q. In the generalized planted biclique detection problem, the goal is to distinguish between a random bipartite graph drawn from $G(m, n, q)$ and one containing a planted biclique of size $(k \times k)$ on a uniformly chosen set of vertices. By instantiating Problem 1.2 with the null (μ_0) and planted distributions (μ_θ) as in Table 1, and setting $t = \tilde{\Theta}(k^2/q)$, we obtain the following multi-pass streaming lower bound.

Theorem 1.4 (Informal version of Theorem 5.2)**.** *For the planted k-biclique problem in random bipartite graphs $G(m, n, q)$, any p-pass streaming algorithm that observes the adjacency lists of left vertices in random order and achieves a constant distinguishing advantage requires at least $\tilde{\Omega}\left(\frac{nmq}{p \cdot k^4}\right)$ bits of memory.*

To apply Theorem 1.3, we require that the likelihood ratio $\mu_1(x)/\mu_0(x)$ be bounded for all x. This fails when $t \ll k^2/q$, as planting k ones substantially changes the number of ones under μ_0. Using careful truncation arguments (briefly outlined in Section 2.4), we show the condition holds for $t \gg k^2/q$. While $q = 1/2$ is the most common setting, it is noteworthy that our framework applies to general q. In particular, the case $q = \frac{\text{polylog}(n)}{n}$ is crucial for our new memory lower bound on approximating the density of subgraphs, discussed in the next subsection.

Fix $m = n$ and $q = 1/2$. In the regime where the algorithm has $\mathsf{poly}\log n$ space (a common notion of space-efficient computation, particularly with regards to planted clique [Mar21a]), the result says that it is not possible to detect cliques of size $O(n^{1/2-\delta})$, unless the algorithm makes $n^{\Omega(\delta)}$ passes over the data. Since we are usually interested in algorithms with constant or logarithmic number of passes over the data in streaming settings, the bound says that the problem cannot be solved with $n^{o(\delta)}$ space in those settings. The result is tight in the sense that for clique size $k = \Omega(\sqrt{n \log n})$, simply counting edges (which uses $O(\log n)$ space and one pass) suffices [Kuč95].

We next relate our result to prior work on the hardness of planted clique detection in the streaming model. For worst-case graphs, [HSSW12] and [BLS+18] prove a $\tilde{\Omega}(n^2/r^2)$ memory lower bound for one-pass algorithms that compute an r-approximation of the maximum clique size, and [BLS+18] also provide a matching upper bound (which extends to bicliques). Since the largest (bi)clique in $G(n, n, 1/2)$ has size $\Theta(\log n)$ with high probability, a $\tilde{\Theta}(k)$-approximation suffices to detect a planted k-clique, yielding a $\tilde{O}(n^2/k^2)$-space one-pass algorithm for the planted biclique problem – leaving room to tighten[1] our lower bound by a factor of $\tilde{O}(k^2)$ when $k < \sqrt{n}$. For the planted clique problem, [RWYZ21] also establish a $\Omega(n^2/(pk^4))$ memory lower bound for p-pass algorithms, but in a stronger model where edges arrive in an adversarial order. In contrast, our model is arguably more natural and easier, as it reveals all neighbors of each vertex together while vertices arrive in random order. The only prior work establishing non-trivial lower bounds in a related communication model – where each player receives the adjacency list of a vertex – is [CG19], which applies only to cliques of size at most $n^{1/4}$.

Planted biclique under monotone adversaries Starting with the work of [FK98], the *monotone adversary model* studies the extent to which algorithms for planted clique depend on the specific distributional assumptions of the problem. The monotone adversary model corresponds to starting with the standard input for planted clique, after which an adversary is allowed to remove any edges not belonging to the planted clique (if the graph has a planted clique). Since the adversary only removes such edges, it is in some sense helpful. [FK00] showed that while simpler algorithms

[1] Note that since our general framework yields memory–sample tradeoffs, Theorem 1.4 is tight for the distributional version of the planted biclique problem (Problem 1.1).

based on edge counting and the spectral method fail at the $k = \tilde{O}(\sqrt{n})$ threshold, a semi-definite programming based method still recovers cliques at the previous $k = \tilde{O}(\sqrt{n})$ threshold in the presence of such adversaries. Our framework can capture monotone adversaries (see Table 1 for the parameters), and we get the following result against streaming algorithms that detect whether there is a clique of size greater than k.

Theorem 1.5 (Informal version of Theorem 6.4)**.** *For the planted biclique problem in the presence of a monotone adversary, any successful p-pass streaming algorithm that detects the presence of planted cliques of size at least k, requires $\tilde{\Omega}\left(\frac{n^2}{p \cdot k^3}\right)$ bits of memory.*

The result shows that the threshold for solving the problem in constant passes with $O(\log n)$ memory moves from $k = \tilde{\Omega}(\sqrt{n})$ to $k = \tilde{\Omega}(n^{2/3})$ — showing that somewhat strong distributional assumptions are needed to solve the problem at the $k = \tilde{\Omega}(\sqrt{n})$ threshold with small memory. Note that the previous algorithm based on counting the number of edges no longer works in this model, though the $\tilde{O}\left(\frac{n^2}{k^2}\right)$ memory one-pass algorithm from [BLS+18] does work. It is possible that no $O(\log n)$-space, constant-pass algorithm can solve the problem in the presence of a monotone adversary for $k = \tilde{\Omega}(n)$, suggesting that even a monotone adversary may make the planted biclique problem as hard for streaming algorithms as in the worst-case setting.

1.3 Application to graph streaming under the vertex arrival model

In this section, we focus on general undirected graphs, not necessarily bipartite, and study the memory requirements for approximating certain graph properties in the *vertex arrival* streaming model. In this model, vertices arrive in a worst-case order, and each new vertex reveals its connectivity to all previously arrived vertices. This model is natural for graph streaming problems and has been fairly studied; the seminal work of [KVV90] on online bipartite matching in the vertex arrival model sparked extensive research in this area. More recently, [Kap21] established a separation between the edge and vertex arrival models for the online bipartite matching problem.

For other graph properties, while interesting upper bounds are known (e.g., [KMPV19] for triangle counting), lower bounds in the vertex-arrival streaming model are hard to come by. Among such problems, approximating the maximum clique or independent set size has been the most studied [BLS+18, CDK18, CDK19]. While [HSSW12] established a tight $\Omega(n^2/\alpha^2)$ memory lower bound for one-pass algorithms, under the edge-arrival model, that compute an α-approximation to the maximum clique size, it is conceivable that the same approximation might be achievable using lesser memory in the vertex-arrival model. In fact, [CDK19] showed that computing a *maximal* independent set is trivial in vertex-arrival streams but requires $\Omega(n^2)$ space in edge-arrival streams. In terms of lower bounds, [CDK19] proved that any α-approximation of the maximum clique size in one-pass vertex-arrival streams requires $\Omega(n^2/\alpha^7)$ space, while [BLS+18] established an incomparable $\Omega(n/\alpha^2)$ lower bound for one-pass adjacency-list streams. Theorem 1.5 implies the following stronger memory lower bound for *multi-pass* streaming algorithms that approximate the size of the largest *biclique* in an undirected graph.

Theorem 1.6 (Informal version of Corollary 6.6)**.** *Any p-pass streaming algorithm in the vertex-arrival model that approximates the maximum biclique size within a factor of $\alpha \geq 1$, must use $\tilde{\Omega}\left(\frac{n^2}{p\alpha^3}\right)$ memory.*

Approximating density for β-bounded subgraphs Next, we turn to the *densest subgraph* problem, a fundamental primitive in graph mining that has been extensively studied since the 1970s (see the excellent survey by [LMFB24] on the problem and its variants). Broadly, the goal is to find a subset of vertices S maximizing the ratio of the number of edges within S to $|S|$, referred to as the *density* of S. [BKV12] initiated the study of streaming algorithms for this problem, presenting an $O(\log n)$-pass algorithm that achieves a constant-factor approximation using $O(n)$ bits of memory. They also proved that any p-pass streaming algorithm that α-approximates the maximum density requires $\Omega(n/(p\alpha^2))$ bits of memory under worst-case edge arrival streams. We establish the following stronger memory lower bound for the harder problem of approximating the maximum density of subgraphs of size at most β, in the (potentially stronger) vertex-arrival streaming model.

Theorem 1.7 (Informal version of Corollary 5.9)**.** *For any $\alpha \geq 1$, any p-pass streaming algorithm in the vertex arrival model, which α-approximates the maximum density among all subgraphs of size at most β for $\beta = o(n/\alpha^2)$, requires at least $\tilde{\Omega}\left(\frac{n^2}{p\cdot\alpha^4\beta}\right)$ bits of memory.*

The multi-pass streaming lower bound of [BKV12] is based on a reduction from set disjointness, which critically relies on the worst-case edge-arrival order and does not extend to vertex-arrival streams. To prove the above theorem, we establish a reduction from the planted biclique detection problem (Theorem 1.4) with parameters $k = \alpha \log n$ and $q = \log n/\beta$. Since detecting planted bicliques is believed to be computationally hard even in "sparse" graphs, we cannot hope to extend our hardness result beyond an approximation factor of $\alpha = n/\beta$; since simple greedy algorithms [AITT00] are known to give such approximation to maximum density of β-bounded subgraphs.

1.4 Applications to canonical learning problems over Gaussians

We now discuss our results for learning problems over Gaussians. Gaussian distributions pose significantly more challenges in bounding the likelihood ratio μ_1/μ_0 between the planted and null distributions. In Section 2.5, we discuss how to suitably truncate the distributions to apply our framework in more detail.

Detecting sparse mean Gaussians We now consider the problem of testing whether the data — or some of the data — is coming from a Gaussian with a sparse mean. This is a fundamental problem with a long line of work [Ing96, BAR02, IS03, DJ04, JW07, CCT17, CCTV18]. It models various applications where the goal is to do hypothesis testing to determine if there is some sparse signal present in the data. In many applications such as anomaly detection the signal is also 'weak' and not all datapoints come from the planted distribution (see [DJ04] and the survey [DJ15]), and there has been significant work on detecting such signals which are both sparse and weak [DJ04, DJ08, HJ10, KS13]. This aspect can also be captured by our general setting in Problem 1.2 (through choice of the set 'R').

We now describe the sparse Gaussian testing setting in more detail. We first draw the planted mean vector $\theta \in \{0, \alpha\}^d$ uniformly at random from the set $\{0, \alpha\}^d$, but subject to it being ℓ sparse. Here $\alpha \in [0, 1]$ is the signal strength parameter. Let $q > 0$ be the probability of getting a planted sample. In the null distribution, we always get samples from $N(0, I)$. In the planted distribution, at every time step with probability q we get a sample from $N(\theta, I)$, and with probability $(1 - q)$ we get a sample from $N(0, I)$. Using our general framework, we show the following memory-sample

tradeoff for algorithms which take as input a lower bound on the sparsity in the planted case, and then work for all sparsity levels above this lower bound.

Theorem 1.8 (Informal version of Theorem 7.2)**.** *For the problem of detecting sparse mean Gaussians where the mean vector has sparsity at least ℓ, any successful p-pass, s-bit memory algorithm which uses n samples requires $s \cdot n \geq \Omega\left(\frac{d^{0.99}}{p \cdot (\alpha \ell q)^2}\right)$.*

We note that our general framework is versatile enough to capture dependence of the tradeoff on the signal strength α here, and the bound also holds for constant values of α where the planted vector has a super-constant norm. Several other remarks about this lower bound are in order, starting with upper bounds for this problem. By storing the sum of all the co-ordinates of all the vectors, it is possible to distinguish the two distributions with $n = O\left(\frac{d}{(\alpha \ell q)^2}\right)$ samples (since, roughly, the means in the planted versus null case differ by $\alpha \ell q n$, and the variance is $O(nd)$). Therefore the problem can be solved with a one-pass $O(\log d)$ memory algorithm, but using $O\left(\frac{d}{(\alpha \ell q)^2}\right)$ samples. Our bound shows that this sample-complexity is near-optimal and necessary for $O(\log d)$ memory constant pass algorithms. This required sample complexity for $O(\log n)$ memory algorithms is significantly worse than the optimal sample complexity without memory constraints. We can solve the problem by storing $\tilde{O}(d/\ell)$ randomly chosen co-ordinates of $\tilde{O}\left(\frac{1}{q\alpha^2}\right)$ datapoints, and by checking the empirical averages of the co-ordinates for every $\tilde{O}\left(\frac{1}{\alpha^2}\right)$ sized subset of the stored points. This requires $\tilde{O}\left(\frac{d}{\ell \alpha^2 q}\right)$ memory and $\tilde{O}\left(\frac{1}{\alpha^2 q}\right)$ samples. $\tilde{O}\left(\frac{1}{\alpha^2 q}\right)$ is the information-theoretic sample complexity of the problem, and hence our memory lower bound to achieve optimal sample complexity is optimal up to a factor of $\frac{d^{0.01}}{\ell \alpha^2}$. We also note that the lower bound shows that memory-limited algorithms need a sample complexity which depends on $1/q^2$, whereas information-theoretically only a $1/q$ dependence is needed — hence memory-limited algorithms could need many more samples to detect outliers or find weak signals in the data distribution. This is similar to gaps observed for the *needle problem* [AMOP08, CCM08, LZ23], where the goal is to detect if a data stream has one element which appears with a higher than uniform probability.

Even for the case of $q = 1$ where all samples are drawn from a Gaussian with a sparse mean, we are unaware of previous memory lower bounds for the detection problem, though memory lower bounds are known for the estimation version of the sparse Gaussian mean problem [ZDJW13, GMN14, BGM+16].

Sparse PCA detection problem Sparse PCA adds a sparsity constraint to the PCA problem and has found widespread applications in statistics, ML and data analysis [ZHT06, ZX18]. As discussed earlier, it is also a prototypical problem for studying understanding statistical-computational tradeoffs. From the perspective of memory constraints, streaming algorithms have been developed for sparse PCA [MBPS10, YX15, WL16, KS24] — building on developments in streaming PCA [MCJ13, JJK+16]. These algorithms all need at least $\Omega(d)$ memory to find the sparse principal component, but the trivial information-theoretic lower bound only says that $\tilde{\Omega}(k)$ memory is needed for the estimation problem if the principal component is k-sparse. We are unaware of any previous non-trivial memory lower bound for the problem.

We describe the detection version of the sparse PCA problem. We first draw the sparse principal component $\theta \in \{0, 1/\sqrt{\ell}\}^d$ uniformly at random from the set $\{0, 1/\sqrt{\ell}\}^d$, but subject to it being ℓ-

sparse. The goal is to distinguish whether the samples are coming from $N(0, I)$ or from $N(0, \Sigma)$, where $\Sigma = I + \alpha\theta\theta^\top$. This is the widely studied spiked covariance model, also known as the spiked Wishart model [ZHT06, JL09]. Here $\alpha > 0$ is the signal strength parameter, and we consider α which is a small enough constant. Note that in contrast to previous settings, here all the samples have the sparse, planted structure, as is standard in sparse PCA. We show the following tradeoff for this problem.

Theorem 1.9 (Informal version of Theorem 8.2)**.** *For the sparse PCA detection problem, any successful p-pass, s-bit memory algorithm which uses n samples requires $s \cdot n \geq \Omega\left(\frac{d^{0.99}}{p \cdot \ell}\right)$.*

For the small-memory regime where $s = O(\log n)$ and $p = 1$, our result shows that $\Omega(d^{0.99}/\ell)$ samples are necessary. In contrast, note that the problem is information-theoretically solvable with only $\tilde{O}(\ell)$ samples [MW15a]. Therefore, small, $O(\log n)$-memory algorithms need significantly more samples than the information-theoretic limit to solve the problem. In the $O(\log n)$-memory regime, it is possible to solve the problem with $\tilde{O}(d)$ samples by thresholding the sum of the squares of all the co-ordinates over all the samples. Therefore, there is a gap of ℓ (and some other less significant terms) between our lower bound and the best-known upper bound. However, we show our lower bound for a more structured version of the problem where a consecutive set of co-ordinates of θ are non-zero (in the technical overview in Section 2.5, we discuss this further). In the presence of this structure, it is possible to solve the problem with $\tilde{\Omega}(d/\ell)$ samples, by thresholding the squares of the sum of consecutive co-ordinates instead. Therefore, our bound is nearly tight for this setting that we consider.

To the best of our knowledge, our result represents the first memory-sample tradeoffs for sparse PCA in the standard spiked covariance model, either for the detection or the estimation version of the problem. Note that a reduction is known from the planted clique problem to the sparse PCA [BR13, Mar21b], however this reduction does not work in the streaming model. The closest related setting for which memory-sample tradeoffs are known is for detecting if a pair of co-ordinates in samples drawn from an unknown distribution are correlated [Sha14, DS18]. This is similar to the sparse PCA problem when $\ell = 2$. However, the bound of [Sha14, DS18] only holds when the correlation (which is analogous to our signal strength parameter α) is polynomially small in d (in which regime they prove a stronger bound than Thm 1.9), in contrast our bound holds for constant values of α, and importantly, generalizes beyond the case of correlations where $\ell = 2$.[2]

1.5 Other Related Work

We now discuss some other relevant literature. There has been significant work on understanding learning under information constraints such as limited memory or communication constraints [BBFM12, DJW13, Sha14, AS15, SD15, Raz18, DKS19, WBSS21]. The works closely related to our work are [BBS22, LWZ25, BGL+24], where the former two also study information cost for a similar setup to planted biclique (Task B in their paper) en route to showing memory lower bounds for certain *estimation* problems. However they measure information cost with respect to a non-uniform distribution which prevents our subsequent direct sum application, and hence their analysis is not helpful for us. Particularly, while Task B in these papers plants $0/1$ uniformly on a k-sized subset, at best their proof can be massaged to show that detecting cliques of size k requires $\Omega(n/k^3)$ memory,

[2]Note that in the sparse PCA problem the co-ordinates of the samples are not independent, and hence we cannot do a direct reduction from $\ell = 2$ to larger values of ℓ.

whereas we show a $\Omega(n^2/k^4)$ bound which requires significantly new techniques and measuring information *w.r.t.* the null distribution. Secondly, hybrid arguments used by these papers fail to work for our other applications such as sparse PCA where planted coordinates are correlated.

Work on the needle problem [AMOP08, CCM08, LZ23, BGL+24] also shares elements of our analysis. However, bounds for the needle problem do not yield bounds for our general planted structure detection problem since the needle problem is limited in the sense that a needle is chosen uniformly from the null distribution. Another relevant set of papers is the work of [FGR+17] on variants of statistical query (SQ) dimension for Problem 1.1 and the work of [GRT18] which shows memory-sample tradeoffs parameterized by the SQ dimension of the problem using extractor-based bounds. However, this connection is weak to give anything non-trivial.

In the graph streaming literature, there is an extensive literature on both upper bounds and lower bounds (see survey [McG14] on upper bounds and [Ass23] on lower bounds for an overview). Typical lower bounds here are for worst-case edge arrival graphs, whereas our results hold in the vertex arrival model as well as random order. Finally, we note that there is a large body of work on the problem of finding outliers in streaming data, such as [TTL11, MMA16, ALPA17], see [LWZ23] for a detailed survey. There is also work on memory lower bounds for streaming outlier detection [SGW18], but for worst-case data.

2 Technical Overview

In this section, we present an overview of our proofs, beginning with the general planted structure detection problem in Problem 1.2. Recall that in this problem, only one set in the partition contains a planted structure, and only a subset of rows are planted (see also Figure 1).

We start with a simpler setting involving only a single set in the partition, and where all rows are planted. For this case, we prove a new data processing inequality to establish information cost lower bounds with respect to the *null* distribution (Section 2.1). We then extend our analysis to the case where only one set in the partition is planted – matching the structure of Problem 1.2 – but all rows remain planted. Using direct-sum-type arguments, we establish an information complexity lower bound for this case (again with respect to the null distribution), building on the bounds provided by the data processing inequality (Section 2.2). Finally, we address the full problem where only a subset of rows are planted. Here, we apply the recent multi-pass information cost framework of [BGL+24] to derive memory lower bounds (Section 2.3). This step introduces an additional $\sim n$ factor in the bound and critically relies on the fact that the previous bounds measure information under the null distribution. We conclude by describing applications of our general framework to graph problems and statistical detection tasks in Section 2.4 and Section 2.5, respectively.

As an instrumental warm-up exercise, we first consider the case of single partition; under the **no** case, at each time-step, we get a sample drawn from some distribution μ_0 on t-dimensional vectors. Whereas in the **yes** case, a sample is drawn from the *planted* distribution – μ_θ – with probability γ (think of γ as k/n). As we want to develop a general framework for studying hardness of detecting planted structures, we want to make as few assumptions on μ_0 and μ_θ, where the parameter θ takes value in some set Ω.

[BGM+16] studies a similar distribution detection question (albeit) under the communication complexity model, where every player independently gets a sample either from μ_0 or from μ_1. [BGM+16, Theorem 1.1] establishes information complexity (IC) lower bounds (*w.r.t.* μ_0) when μ_1 is

point-wise bounded by $O(\mu_0)$. Even though such a restriction might seem stringent, distributions under many natural detection problems such as Gaussian mean estimation can be truncated to satisfy it. However, for the distributions we consider – for example, in the planted clique problem, μ_0 is the uniform distribution over t-dimensional vectors, whereas μ_θ has a planted 1s on a set indexed by θ — $\mu_\theta(x)/\mu_0(x)$ can be exponentially large for "typical" xs. In fact, we cannot hope to prove memory lower bounds when we know θ, as even a single time-step can detect the outlier without any prior knowledge. Hence, under the random process $\theta \sim P$, we at least want that $||\mathbb{E}_{\theta\sim P}\mu_\theta - \mu_0||_{TV} = o(1)$. One of our main technical contributions is the following generalized distributed data processing inequality, when the expected distribution $\mathbb{E}_{\theta\sim P}\mu_\theta$ is point-wise bounded by $O(\mu_0)$.

Theorem 2.1 (Informal version of Theorem 4.8 + Lemma 4.12)**.** *Let $\mu_0, \{\mu_\theta\}_{\theta\in\Omega}$ be a family of distributions over some sample space $\mathcal{X}$ such that $\mathbb{E}_{\theta\sim P}\mu_\theta \leq O(\mu_0)$. Consider the distributed detection setting, where if $V = 0$, then each party receives $x_i \sim \mu_0$, and if $V = 1$ then we first draw $\theta \sim P$, and then each party receives $x_i \sim \mu_\theta$. Then, for any multi-party communication protocol Π that learns V with large enough constant probability,*

$$I(X;\Pi(X)) = \Omega(1), \text{ when } \forall i, X_i \sim \mu_0.$$

One can view the above theorem as a generalization of [BGM+16, Theorem 3.1] to non-product distributions and might be of independent interest.[3] Before diving into the proof overview, let's talk about a bit about the implications of this theorem. If V is a uniform bit, then it is trivial to show that any communication protocol that detects V requires $\Omega(1)$ information $w.r.t.$ the mixture distribution. However, our key objective is to establish an information complexity lower bound with respect to μ_0, that is, the **no** distribution. This is crucial for two reasons. Firstly, using a direct sum argument over d/t partitions, we can prove an $\Omega(d/t)$ multi-party communication lower bound for detecting planted structures when every player gets a sample from the planted distribution (that is, $\gamma = 1$). Secondly, leveraging the recently introduced information cost notions of multi-pass streaming algorithms [BGL+24], we are able to lift IC bounds for communication protocols to memory lower bounds that grow quadratically with $1/\gamma$. Again, this is possible because we always use information cost measures $w.r.t.$ **no** distribution, which is a product distribution. We discuss these in more detail in Section 2.2 and Section 2.3 respectively.

2.1 Generalized distributed data processing inequality

We follow and build upon the proof of [BGM+16, Theorem 3.1]. Let $\mu_1 = \mathbb{E}_{\theta\sim P}\mu_\theta$. Then, the theorem from [BGM+16] would show an $\Omega(1)$ bound on IC $w.r.t.$ μ_0 for communication protocols that distinguish between cases where each player either receives *independent* samples from μ_0 or from μ_1. In fact, their results potentially offer a stronger bound involving the strong data processing inequality (SDPI) constant[4] of μ_0 and μ_1. In Theorem 2.1, we are aiming for an $\Omega(1)$ IC bound $w.r.t.$ μ_0 but when under the $V = 1$ case, samples are not independent.

When $V = 1$, first we draw θ from a prior distribution P, and then every player gets an independent sample drawn from μ_θ. Note that if the players knew θ, then a *single* player can

[3] While [BGM+16, Theorem 3.1] has been widely used to analyze distributed computing under information constraints, [DR19] noted that the independence condition in the theorem is often too strong.

[4] We refer the reader to [Rag16] for a survey on SDPI. In this paper, we will only be using data processing inequalities.

solve the detection problem, without sending $\Omega(1)$ bits of information under μ_0. For example, for the planted biclique problem on t-dimensional vectors, a single player can send the AND on k bits of the plant, which reveals $O(1/2^k)$ bits of information about the input under uniform distribution. This is the first step in the proof – to show that a single player cannot solve the distinguishing problem without revealing information under μ_0 distribution. In fact, there is no distinction between the two detection problems (the one we study from the one where each player gets a sample from μ_1 under the yes case), when only *one* player receives a sample from the planted distribution ($\mu_\theta, \theta \sim P$) while all others receive samples from the null distribution μ_0. For the *single player* setting, we can adopt the previous proof to show that to detect V, the player needs to send $\Omega(1)$ bits of information under μ_0.

For communication protocols under *product* input distributions, one can use the cut-and-paste property [BYJKS04] to connect the distinguishing capacity of the single player setting to when each player gets a sample according to μ_0 or μ_1. However, this property does not hold for non-product input distributions. Our approach capitalizes on the fact that once we condition on θ, the cut-and-paste property still holds. As we are using information-theoretic quantities that easily tensorize and work well with linearity of expectation, we are able to use both 1) that the cut-and-paste property holds for every θ, to lift hardness of single player setting to the when all receive samples from μ_0 or μ_θ, and 2) that in the single player setting, to solve the detection problem on average over $\theta \sim P$, one needs to reveal information $w.r.t.$ μ_0.

2.2 Direct sum over the partitions

In Problem 1.2, under the planted distribution, we embed samples from μ_θ on a random partition of d-dimensional vectors of size t, as well as on a random subset of "rows". While Theorem 2.1 is a crucial part of the our multi-pass memory lower bounds, under the general framework, it doesn't say anything about memory needed to distinguish. To prove Theorem 1.3, we apply *direct-sum* like techniques twice; once on the partitions and another on the randomness of rows. In this subsection, we discuss the former. As Theorem 2.1 gives us an IC bound $w.r.t$ μ_0, using a standard direct-sum argument, we get $\Omega(d/t)$ IC bound for any communication protocol that distinguishes the case when all player get samples from $\mu_0^{\otimes d/t}$, from the case when *all* players get samples from μ_θ embedded at a fixed partition (that is, $n = k$ case in Problem 1.2). See Lemma 4.5 for the formal statement of the result.

Consider the special case of the planted clique distributed detection problem, where under the planted distribution, each player (total k players) receives an n-dimensional vector with 1s planted on a pre-chosen set of coordinates S of size k. For $n >> k^2$, we can adjust the distributions over the vectors to satisfy the requirements of Theorem 2.1, and get an $\Omega(1)$ IC lower bound $w.r.t.$ uniform distribution (we discuss the truncation more in Section 2.4). In fact, using Lemma 4.5, we can get an $\tilde{\Omega}(n/k^2)$ IC lower bound[5], $w.r.t.$ the uniform distribution, for any communication protocol that solves this planted clique distributed detection problem. In Section 2.3, we will crucially use the fact that this is an IC bound $w.r.t.$ uniform distribution (μ_0 generally), which is stronger than a lower bound on the amount of communication needed to solve the detection problem.

The total communication lower bound of $\tilde{\Omega}(n/k^2)$ is interesting in its own right. Notably, it immediately implies a memory lower bound of $\tilde{\Omega}(n/pk^3)$ bits for any p-pass streaming algorithm solving the planted biclique version of Problem 1.2. This is because, even if we fix the rows

[5]This is tight upto $\log n$ factors if we increase the number of players to $\sim n/k^2$.

where the plants are made, distinguishing them from the uniform distribution would still require significant communication. In the next subsection, we leverage row-level randomness to establish a stronger memory lower bound of $\Omega(n^2/pk^4)$ bits, which is optimal (upto $\log n$ factors) for detecting planted bicliques of size $\mathsf{poly}\log n$, and provides non-trivial memory lower bounds for biclique sizes up to the critical threshold of $k = \sqrt{n}$.

This communication complexity lower bound also implies that no $n^{o(1)}$ round protocol in the Broadcast Congested Clique model ($BCAST(\log n)$) can detect planted bicliques in directed graphs, even when the biclique size is as large as $n^{1/3-\epsilon}$ for some constant $\epsilon > 0$. In comparison, the previous work of [CG19] established similar bounds for the planted clique problem in directed graphs for cliques of size at most $n^{1/4-\epsilon}$. Since our primary focus is on multi-pass streaming algorithms, we do not include further discussion on the implications for the $BCAST(\log n)$ model.

2.3 Lifting to memory lower bounds

Next, our goal is to lift the communication lower bounds established in Lemma 4.5, when each row gets a "plant" in the planted distribution, to memory lower bounds in the streaming setting (beyond the implications discussed in the last subsection), when each row gets a "plant" with probability γ. For sake of exposition, let us look at the planted bi-clique problem when $k = \mathsf{poly}\log n$. All the techniques discussed in this subsection readily generalize to Problem 1.2, when IC notions are measured $w.r.t.$ μ_0. Lemma 4.5 gives an $\Omega(n/k^2)$ bound on information complexity ($w.r.t.$ $\sim$ uniform distribution on every player's input) for any k-party communication protocol that solves the planted bi-clique problem when $n = k$.

In Problem 1.2, the planted distribution only samples from the non-uniform distribution on $k = \gamma n$ number of rows R, and this set R is chosen uniformly at random. Thus, we want to use the fact that any multi-pass streaming algorithm that doesn't know R needs to solve, the distributed detection problem for the special case, where all rows are planted, for multiple instances embedded in the stream, *simultaneously*. Such arguments are usually made using direct sum theorems in communication complexity, but it is challenging to use these techniques to prove optimal[6] memory bounds that we get. One challenge is that as we want hardness for a distributional problem, we are aiming for multi-pass lower bounds for stochastic streams and hence, we cannot strategically embed multiple instances of the communication problem in the stream, so as to force a *single* time-step to communicate a lot to the next time-step.

Recent work of [BGW20] introduced a new information cost notion for one-pass streaming algorithms, which is amenable to memory lower bounds for stochastic streams. We leverage the multi-pass information cost (MIC) notion, introduced in [BGL+24], to prove our result. Briefly, these information cost notions measure the information a time-step needs to retain about the input stream *on average*. As these information cost notions are only meaningful for product distributions over the time-steps, it is crucial that we prove the IC bound $w.r.t.$ a product distribution over the players' inputs (Lemma 4.5). Secondly, as the rows – where the communication problem is embedded in the stream – is random, to be able to use tensorization properties of the MIC notion, we need the distribution over every sample without a plant to have the same distribution as the no distribution. Hence, the proof heavily relies on the fact that we prove information complexity bounds $w.r.t.$ $\mu_0^{\otimes d/t}$, which is the distribution for every sample under the null distribution. Once we have the right IC bounds, to get a multi-pass information cost lower bound that depends

[6]Here, by optimal, we mean that we get non-trivial memory lower bounds even for $k = n^{1/2-\epsilon}$ for constant $\epsilon > 0$.

quadratically on $1/\gamma$, we use a similar argument to one made in [BGL+24] for lifting the hardness of MostlyEQ to the needle problem.

2.4 Applications to planted biclique and other graph streaming problems

Our general-purpose framework for proving memory lower bounds for detecting planted structures yields multi-pass memory lower bounds for the classical planted bi-clique problem in bipartite graphs, as well as a semi-random version of the planted bi-clique problem. These lower bounds in turn allow us to derive worst-case memory-bounded hardness of approximation for fundamental graph problems like the *Maximum Bi-Clique* and *Densest at-most-β Subgraph*, in a natural (potentially stronger) vertex-arrival streaming model. Obtain these bounds require a careful application of our general framework, and we elaborate on some of the technical aspects below.

Consider first the planted bi-clique problem. We first note that to get a lower bound for the planted bi-clique problem where k out of n coordinates are planted uniformly at random, it is sufficient to show lower bounds for an easier version where we partition the n coordinates into n/t subsets of size $t \geq k$, and all the planted k co-ordinates belong to one subset in the partition. To see this, note that an algorithm $\mathcal{A}$ that can solve the general non-partition version of the problem can be used to solve the partition version, by simply permuting the n coordinates of each input according to a consistent, uniformly random permutation, and feeding this input to $\mathcal{A}$. Now, to show a lower bound for this partition version of the problem, we first need to fix the size of the partition. Note that if $t = o(k^2)$, then simply counting the number of ones in each subset of the partition suffices to distinguish between the uniform and planted distributions, since with high probability only $t/2 \pm O(\sqrt{t})$ ones are observed in each subset for the uniform distribution. Therefore, we will choose $t = \Omega(k^2)$.

The next step, which is also the key technical step in all applications of our framework, is to appropriately truncate the distributions μ_0 and μ_θ to obtain new distributions $\tilde{\mu}_0$ and $\tilde{\mu}_\theta$ such that: (1) $\tilde{\mu}_0$ and $\tilde{\mu}_\theta$ are close to μ_0 and μ_θ respectively; (2) but they satisfy that $\mathbb{E}_\theta \tilde{\mu}_\theta := \tilde{\mu}_1$ is pointwise upper-bounded by $c \cdot \tilde{\mu}_0$ for some constant c. The reason this truncation is necessary is that the original distributions μ_0 and μ_θ *do not* satisfy that $\mathbb{E}_\theta \mu_\theta := \mu_1$ is pointwise upper-bounded by $c \cdot \mu_0$ for some constant c. This is because t-bit strings which have exactly k ones have roughly 2^k more probability under $\mathbb{E}_\theta \mu_\theta$ than under μ_0, since strings under μ_θ always have at least k ones. To address this, we restrict to typical strings which have $t/2 \pm O(\sqrt{t \log t})$ ones, and define $\tilde{\mu}_0$ and $\tilde{\mu}_\theta$ by restricting μ_0 and μ_θ to such strings. This truncation allows us to bound c by a constant, and effectively leverage the lower bound from the general framework.

Our next result shows a stronger memory lower bound for the semi-random version of the planted bi-clique problem, where an adversary is monotone—it can only remove ones from non-planted locations. Our main insight here is to relate this problem with a monotone adversary to a slightly modified version of the planted bi-clique problem itself. While in the standard planted bi-clique problem, we plant the $\overrightarrow{1}_k$ vector at some subset of k coordinates, we instead think of the version where we plant an arbitrary vector $v \in \{0,1\}^k$ at these coordinates. To related this "fixed-pattern" planted bi-clique problem to monotone adversaries, note that the generated samples for a fixed pattern vector v with k' ones are equivalent to samples from a planted bi-clique problem with a bi-clique of size k', but in the presence of a monotone adversary which forces a consistent set of $k - k'$ non-planted coordinates in the planted rows to be 0. The fixed-pattern planted bi-clique problem is at least as hard as the standard problem, and in fact we show a stronger lower bound for

it. This is because for this problem we can choose the size of the partitions to be as small as $t = k$, since the vector that we plant within the partition is also a uniformly random vector, and therefore the number of ones in the partition is still typical. This allows us to improve on the memory lower bound for this problem by a factor of k.

Finally, we outline how our memory lower bounds above for the planted problems allow us to derive hardness results for approximating both, the densest at-most β subgraph, as well as the maximum bi-clique in undirected graphs, in the vertex arrival model. Both these results are for undirected graphs, whereas the planted bi-clique lower bounds stated above are for bipartite graphs. In transferring the hardness to these undirected graph applications, we need to transition to a different streaming model, where vertices arrive in a worst-case order, and connectivity is only revealed to vertices that have previously occurred in the stream. For both the applications, the technique is similar: if the bipartite graph is planted, the translated undirected graph has a sizable edge density on some subgraph, and also a sizable bi-clique. However, if there is no plant, using standard concentration arguments, we can show these quantities to be small in the translated graph. Thus, if we had an accurate approximation, we can use it to figure out which case we are in. We note that for the densest subgraph application, we crucially rely on our framework allowing us to instantiate the hardness of the planted bi-clique problem with q (the Bernoulli parameter at the non-planted locations) being as small as $\frac{\log n}{\beta}$, as opposed to $q = 1/2$.

2.5 Applications to detecting ℓ-sparse Gaussians and sparse PCA

We now discuss our proof for the detecting sparse Gaussians and for the sparse PCA detection problem. For these applications, appropriately truncating the distributions is more challenging and subtle than for the planted biclique problem.

We first sketch the proof for the sparse Gaussian detection problem. Recall that in the d dimensional ℓ-sparse Gaussian detection problem, in the planted distribution, with probability $(1 - q)$ we get samples from $N(0, I)$ and with probability q we get samples from $N(\theta, I)$, where θ is ℓ-sparse. In the null distribution, we always get samples from $N(0, I)$. For simplicity, we first consider a simpler version of the problem where $\|\theta\|_2 = 1$. The proof here generally follows a similar outline to the planted biclique problem. As in the planted biclique case, we partition the coordinates into sets of size t. In this special case of the sparse Gaussian mean problem, we take $t = \ell$, and $\mu_1 = N((1/\sqrt{t})\overrightarrow{1}_t, I)$. Notice that there is no parameter θ to choose in the planted case in this partition version of the problem, and the co-ordinates in the chosen set in the partition are simply sampled from μ_1. Our goal is now to show that μ_1 is pointwise upper-bounded by $c \cdot \mu_0$ for some that c that is not too large. In this case for any $x \in \mathbb{R}^t$, $\mu_1(x)/\mu_0(x) = \exp((1/\sqrt{t})\sum_j x_j)$. Note that $\sum_j x_j$ can be unbounded, therefore, as in the planted bipartite case, we need to truncate the distributions μ_1 and μ_0. We can truncate the distributions to the set $\{x \in \mathbb{R}^t : \sum_j x_j \leq \sqrt{Ct}\}$ for some C. This is satisfied with high-probability, and allows us to bound $\mu_1(x)/\mu_0(x)$. Using our result for the general planted detection setup (informal version in Theorem 1.3) we get a $s \cdot n \geq \Omega\left(\frac{d^{0.99}}{p \cdot \ell q^2}\right)$ lower bound for p-pass, s-bit algorithms which use n samples.

The general Gaussian case with signal strength α (i.e. non-zero co-ordinates of θ are α) has a similar outline, but requires a much more careful analysis to get the dependence on the signal strength parameter α. First, we show via a reduction that to show a bound for the original problem where θ is exactly ℓ-sparse it suffices to show a lower bound for the case where each co-ordinate of the planted mean vector θ is non-zero with probability ℓ/t. Independence across co-ordinates of θ

facilitates the analysis, and for this distribution of θ we can show that for $\mu_1 = \mathbb{E}_\theta \mu_\theta$, $\mu_1(x)/\mu_0(x)$ is bounded if $\sum_{j=1}^{t} \exp(\alpha x_j)$ is bounded—analogous to the simple case of $\alpha = 1/\sqrt{t}$ sketched above—as long as t is sufficiently larger than ℓ^2. This suggests a truncation: we truncate the distributions μ_0 and μ_θ to x which satisfy an appropriate, α-dependent bound on $\sum_{j=1}^{t} \exp(\alpha x_j)$. Finally, we show that the truncated distributions are close to the original ones using certain concentration bounds.

We now sketch the proof for the sparse PCA detection problem. Recall that in the sparse PCA detection problem the goal is to distinguish whether samples are drawn from the standard Gaussian distribution $\mu_0 = N(0, I_d)$ or from $\mu_\theta = N(0, \Sigma)$ for $\Sigma = I_d + \alpha\theta\theta^T$ for a ℓ-sparse unit vector θ. In this case, $\mu_\theta(x)/\mu_0(x)$ depends on $\exp\left(\frac{\alpha}{2(\alpha+1)}(x^\intercal\theta)^2\right)$. The fact that there is a quadratic instead of a linear term in the exponent makes truncation significantly more difficult here than in previous settings, because $\mathbb{E}_{x\sim N(0,I)}[\exp(cx^2)]$ diverges for $c \geq 1/2$. To see the challenge this poses, consider the following quantity for some $\theta \in \Omega$ (where Ω is the domain of the parameters, such as all k-sparse unit vectors),

$$\mathbb{E}_{x\sim\mu_\theta}\left[\frac{\mu_1(x)}{\mu_0(x)}\right].$$

For all our previous applications such as planted biclique and sparse Gaussian mean detection, $\mathbb{E}_{x\sim\mu_\theta}\left[\frac{\mu_1(x)}{\mu_0(x)}\right]$ is a *constant*. Intuitively, concentration bounds then allow us to show that with high probability over $x \sim \mu_\theta$, $\frac{\mu_1(x)}{\mu_0(x)}$ is bounded. This allows us to find a truncation set T such that with high probability x lies in T for all μ_θ, and moreover $\frac{\mu_1(x)}{\mu_0(x)}$ is bounded for all $x \in T$, which allows us to use our general framework to derive lower bounds. In the case of sparse PCA detection, $\mathbb{E}_{x\sim\mu_\theta}\left[\frac{\mu_1(x)}{\mu_0(x)}\right]$ is *unbounded* if $\alpha \geq 1$, due to the Gaussian integral diverging. This is the reason why our bounds for sparse PCA detection only hold for small constants α. In addition, we have to consider a more structured version of the problem, where the co-ordinates of θ are divided into blocks of size ℓ, and we uniformly select one of the blocks and set all the co-ordinates in that block to $1/\sqrt{\ell}$, with the remaining co-ordinates being 0. With these assumptions, we can then derive a suitable truncation which is satisfied with high probability, and for which $\mu_1(x)/\mu_0(x)$ is bounded.

2.6 Organization of the paper

In Section 3, we define preliminaries and notations. Section 4 formally defines the general problem (Problem 1.2) of detecting planted structures, and proves the main lower bound (Theorem 1.3) for this problem. Next, in Section 5, we instantiate the planted bi-clique problem within this framework, and show the formal memory lower bound (Theorem 1.4) for it. We also outline the densest at-most β subgraph application (Theorem 1.7) here. In Section 6, we continue to study the planted bi-clique problem in the presence of a monotone adversary, and derive a memory lower bound (Theorem 1.5). Here, we also state the application about approximating the maximum bi-clique (Theorem 1.6). Section 7 proves our results on detecting sparse Gaussians (Theorem 1.8). Finally, Section 8 shows our bounds for the sparse PCA detection problem (Theorem 1.9).

3 Preliminaries

We use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$. Given an n-bit vector x, we use x_S to denote the projection of x on set S (ordered lexicographically), that is, $\{x_i\}_{i \in S}$. We use $\overrightarrow{1}$ to denote the all 1s vector when dimension is clear from the context. We use $|x|$ to denote the number of 1s in x. Given two distributions $\mu_0, \mu_1 : \mathcal{X} \to [0, 1]$, we say $\mu_0 \leq c \cdot \mu_1$, if for all $x \in \mathcal{X}$, $\mu_0(x) \leq c\mu_1(x)$. We use capital letters or bold letters, such as $X, Y, Z, \boldsymbol{\theta}$, etc., to denote random variables and x, y, z, θ etc., to denote the values these random variables take. Given a probability distribution $\mathcal{D} : X \to [0, 1]$, we use the notation $x \sim \mathcal{D}$ when value x is sampled according to distribution $\mathcal{D}$. Similarly, we use the notation $z \sim Z$ to denote the process that Z takes value z with probability $\Pr[Z = z]$. We use $\mathrm{Ber}(q)$ to denote the Bernoulli distribution which takes value 1 with probability q and 0 with probability $1 - q$. We use notations $\mathbb{E}[Z]$ to denote the expectation and of random variable Z, and $\mathbb{E}[Z|Y = y]$ to denote the expectation of the random variable Z conditioned on the event $Y = y$. We also use the notation $Z_{|Y=y}$ to denote the random variable Z conditioned on the event $Y = y$.

Basics of information theory. Given two distributions P, Q over $\mathcal{X}$, $D_{KL}(P||Q)$ represents the Kullback–Leibler (KL) divergence of P w.r.t. Q, that is,

$$D_{KL}(P||Q) = \int_{x \in \mathcal{X}} \log \frac{dP(x)}{dQ(x)} dP(x).$$

We will use log base 2 unless stated otherwise. Note that $D_{KL}(P||Q) \geq 0$ for all P and Q.

For random variables X and Y (not necessarily discrete) having joint distribution P_{XY}, and marginals P_X, P_Y respectively, the mutual information between X and Y, denoted $I(X; Y)$, is defined as

$$\mathrm{I}(X; Y) = D_{KL}(P_{XY}||P_X \otimes P_Y), \tag{2}$$

where $P_X \otimes P_Y$ is the product distribution of the marginals of X and Y. Note that $I(X; Y) = I(Y; X)$, and $I(X; Y) \geq 0$ always, by the non-negativity of KL divergence. Note also that mutual information between X and Y can be expressed in terms of the KL divergence as follows:

$$\mathrm{I}(X; Y) = \mathbb{E}_{y \sim Y} \left[D_{KL}(X_{|Y=y}||X) \right].$$

$\mathrm{I}(X; Y|Z)$ represents the mutual information between X and Y conditioned on the random variable Z; and is defined as

$$\mathrm{I}(X; Y|Z) = \mathbb{E}_Z \left[D_{KL}(P_{XY|Z}||P_{X|Z} \otimes P_{Y|Z}) \right], \tag{3}$$

where $P_{XY|Z}$ denotes the joint distribution of X, Y conditioned on Z, and $P_{X|Z}$ and $P_{Y|Z}$ denote the marginal distributions of X and Y conditioned on Z, respectively.

We will routinely use the chain rule for mutual information:

$$I(X; Y, Z) = I(X; Y) + I(X; Z|Y). \tag{4}$$

Using the chain rule, we can derive the following simple facts, which will also come in handy:

1. If $I(A,B;C|D) = 0$, then $I(A,B;C|B,D) = 0$. This follows since $I(A,B;C|D) = I(C;B|D) + I(C;A|B,D)$ by the chain rule; since $I(A,B;C|D) = 0$, by non-negativity of the chain rule, we have that $I(C;B|D) = I(C;A|B,D) = 0$. Finally, by another application of the chain rule, $I(A,B;C|B,D) = I(C;A|B,D) + I(C;B|B,D)$; the first summand is 0 by the preceding argument, and the latter summand is 0 since conditioning on B fully determines B.

2. If $I(A;B|C,D) = 0$, then $I(C;B|D,A) \leq I(C;B|D)$. To see this, apply the chain rule twice on $I(C,A;B|D)$, to get

$$\begin{aligned} I(C,A;B|D) &= I(C;B|D) + I(A;B|C,D) \\ &= I(A;B|D) + I(C;B|A,D). \end{aligned}$$

   Since $I(A;B|C,D) = 0$, we get that $I(C;B|A,D) = I(C;B|D) - I(A;B|D)$. Since $I(A;B|D) \geq 0$, the fact follows.

We will also use Hellinger distance and TV distance as other measures of distance between two distributions. For distributions P and Q over $\mathcal{X}$ having densities p and q respectively, these quantities are defined as follows,

$$h^2(P||Q) = 1 - \int_{x\in\mathcal{X}} \sqrt{p(x)\cdot q(x)}dx, \quad \text{and} \quad ||P-Q||_{TV} = \frac{1}{2}\int_{x\in\mathcal{X}} |p(x)-q(x)|dx.$$

Multi-pass streaming algorithms. Given a stream of n input elements, $x^1,\ldots,x^n$, we say M is a p-pass algorithm (for $p \geq 1$) when it goes over the entire stream p times in order. We use $m_{(\ell,i)}$, for $\ell \in [p], i \in [n]$, to denote the memory state of M in the ℓ-th pass after reading i input elements. Let $m_0 = m_{(1,0)}$ denote the starting memory state and for ease of notation, let $m_{(\ell+1,0)} = m_{(\ell,n)}$ for all $\ell \in [p]$. When the distribution on the input stream is specified, we will use $\mathsf{M}_{(\ell,i)}$, for $\ell \in [p], i \in \{0,1,\ldots,n\}$, to denote the random variable over the corresponding memory states. We will use the random variables $X^1,\ldots,X^n$ to denote the joint distribution over the input stream.

We use notation $[a,b]$ in the subscript to represent random variables indexed from a to b, for example, $\mathsf{M}_{([1,\ell],i)}$ represents i-th memory states for the first ℓ passes, that is, $\mathsf{M}_{(1,i)},\ldots,\mathsf{M}_{(\ell,i)}$. We use notations $< b$, $\leq b$ in the subscript to represent all the corresponding random variables with index less than b or at most b respectively. For example, $\mathsf{M}_{(\ell,\leq i)}$ represents random variables $\mathsf{M}_{(\ell,[0,i])}$.

We will require the following result of [BGL+24] which establishes independence between inputs and private randomness in two segments of the stream, once we condition on the public randomness and the memory states at two different time-steps for all passes. While [BGL+24] do not provide an explicit proof of this particular result, we give a proof in Appendix B for completeness.

Lemma 3.1 (Claim 3.4 in [BGL+24]). *Consider a stream $X^1,\ldots,X^n$ from a product distribution, and let M be a p-pass streaming protocol that uses public randomness P and private randomness $R^{\mathsf{M}} = \{R^{\mathsf{M}}_{l,i}\}_{l\in[p],i\in[n]}$, where the private randomness at every step is mutually independent, as well as independent of the public randomness. Then, for any $i,j \in [n]$, $i < j$, and any $l \in [p]$, it holds that:*

$$I(X^{[i,j-1]}, R_{([p],[i,j-1])}\ ;\ X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{<l,i-1}, \mathsf{M}_{<l,j-1}, P) = 0, \tag{5}$$

$$I(X^{[i,j-1]}, R_{([p],[i,j-1])}\ ;\ X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{\leq l,i-1}, \mathsf{M}_{<l,j-1}, P) = 0. \tag{6}$$

We will use the following notion of information cost for multi-pass streaming algorithms, introduced by [BGL+24]. For a given distribution μ over $X^1, \ldots, X^n$, the information cost of a p-pass streaming protocol M, which uses public randomness P, is given by:

$$MIC(\mathsf{M}, \mu) = \sum_{\ell=1}^{p} \sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\leq \ell, j-1)}, \mathsf{M}_{(\leq \ell-1, i)}, P\right) + \sum_{\ell=1}^{p} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\leq \ell-1, j-1)}, \mathsf{M}_{(\leq \ell-1, i)}, P\right). \quad (7)$$

Here, the random variables for the memory states depend both on the randomness of the input as well as private and public randomness used by the algorithm. When μ is clear from context, we will drop it from the notation.

We will also require the following lemma established in [BGL+24], which bounds the multi-pass information cost notion for any memory-bounded streaming algorithm.

Lemma 3.2 (Lemma 1.1, [BGL+24]). *Let $(X^1, X^2, \cdots, X^n)$ be drawn from a product distribution μ. Then, for any p-pass streaming algorithm M that uses public as well as private randomness, has memory size s and runs on input stream $X^1, \cdots, X^n$, it holds that:*

$$MIC(\mathsf{M}, \mu) \leq 2p \cdot s \cdot n.$$

We note that [BGL+24] proved the above result in the setting where M uses only private randomness; essentially the same proof works for the definition of *MIC* given in (7) when M can additionally use public randomness, and we give the proof in Appendix B for completeness.

4 General Multi-IC Lower Bound for Distinguishing Problems

In this section, we will prove communication and memory lower bounds for a general distinguishing problem, where the goal is to detect if a submatrix has been planted with an *outlier* distribution. Let $\mathcal{X}, \Omega$ be two sets such that $\mu_0, \{\mu_\theta\}_{\theta \in \Omega}$ are distributions on t-dimensional vectors over $\mathcal{X}$. Given a distribution P over the parameter space Ω, we denote the average distribution $\mathbb{E}_{\theta \sim P} \mu_\theta$ by μ_1. Let $d, n > 0$. We study the following distinguishing problem on $n \times d$ sized matrices, when each row of the matrix arrives in a stream.

Problem 4.1. *Let $0 < k \leq n$. Let $\mathcal{T} = \{T_r\}_{r \in [d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. The goal is to distinguish between the following joint distributions on d-dimensional vectors $x^1, \ldots, x^n \in \mathcal{X}^d$:*

1. *D_0: $\forall i \in [n]$ and $\forall r \in [d/t]$, $x^i_{T_r}$ is drawn from μ_0.*
2. *$D_1^{\mathcal{T}}$: Pick r uniformly from $[d/t]$. $\forall i \in [n]$ and $\forall r' \neq r$, $x^i_{T_{r'}}$ is drawn from μ_0.*
   R is drawn uniformly at random from all subsets of $[n]$ of size k. Pick $\theta \sim P$.
   $\forall i \notin R$, $x^i_{T_r}$ is drawn from μ_0. Whereas, $\forall i \in R$, $x^i_{T_r}$ is drawn from μ_θ.

We will refer to this distinguishing problem by $\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta \in \Omega}, P, \mathcal{T}, k, n)$.

Theorem 4.2. *Let $1 < k \le n$. Let $\mu_0, \mu_\theta, \theta \in \Omega$ be distributions on t-dimensional vectors in $\mathcal{X}^t$, and P be a distribution over parameter space Ω such that $\mathbb{E}_{\theta\sim P}\mu_\theta \le c \cdot \mu_0$. Let $\mathcal{T} = \{T_r\}_{r\in[d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. Then, any p-pass streaming algorithm (using public as well as private randomness) that solves the distinguishing problem $\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta\in\Omega}, P, \mathcal{T}, k, n)$ (as defined in Problem 4.1) with large enough constant advantage, requires at least $\Omega\left(\frac{nd}{p\cdot c\cdot k^2 t}\right)$ bits of memory.*

We will prove Theorem 4.2 using the following theorem on information cost of any multi-pass streaming algorithm that solves Problem 4.1.

Theorem 4.3. *Let $1 < k \le n$. Let $\mu_0, \mu_\theta, \theta \in \Omega$ be distributions on t-dimensional vectors in $\mathcal{X}^t$, and P be a distribution over parameter space Ω such that $\mathbb{E}_{\theta\sim P}\mu_\theta \le c \cdot \mu_0$. Let $\mathcal{T} = \{T_r\}_{r\in[d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. Let M be a p-pass streaming algorithm (using public as well as private randomness) that solves the distinguishing problem $\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta\in\Omega}, P, \mathcal{T}, k, n)$ with large enough constant probability. We add another pass to M such that in the $(p+1)$-th pass, M doesn't do any operations but stores $m_{(p,n)}$. Then,*

$$MIC(\mathsf{M}) \ge \Omega\left(\frac{n^2 d}{ck^2 t}\right).$$

Here, the multi-pass information cost is evaluated with respect to the distribution D_0.

Theorem 4.2 then easily follows; for any $(p+1)$-pass streaming algorithm M, that uses s bits of memory, $MIC(\mathsf{M})$ is upper bounded by $2(p+1)\cdot s \cdot n$ (Lemma 3.2).

To prove the above theorem, we will first show an information-complexity lower bound under the blackboard model, when the row set R (in Problem 4.1), that would contain the planted distribution, is known. To show the multi-pass information cost lower bound, we will then embed many such communication problems into the stream. The latter part is similar to the argument made in [BGL+24, Section 5.2]. First, we study the k-player communication protocol that solves Problem 4.1 when $n = k$. Informally, in the no case, each player gets a d-dimensional vector from $\mu_0^{\otimes(d/t)}$, whereas is the yes case, one of the d/t partitions is planted with μ_θ for every player.

$\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta\in\Omega}, P, \mathcal{T}, k, k)$ under k-player number-in-hand communication model

Next, we will prove an $\Omega\left(\frac{d}{ct}\right)$ communication lower bound for any k-party communication protocol that solves the distinguishing problem $\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta\in\Omega}, P, \mathcal{T}, k, k)$. We define the communication problem below for completeness.

Definition 4.4. *(k-party General Planted Problem) There are k parties in the communication problem, where the i-th party holds a d-dimensional vector $x^i \in \mathcal{X}^d$. Let $\mu_0, \{\mu_\theta\}_{\theta\in\Omega}$ be distributions on t-dimensional vectors in $\mathcal{X}^t$, and P be a distribution over parameter space Ω such that $\mathbb{E}_{\theta\sim P}\mu_\theta \le c \cdot \mu_0$. Let $\mathcal{T} = \{T_r\}_{r\in[d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. We promise that $(x^1, \ldots, x^k)$ are sampled from either of the following distributions:*

1. *(No) $\forall i \in [k]$ and $\forall r \in [d/t]$, $x^i_{T_r}$ is drawn from μ_0.*
2. *(Yes) Draw r uniformly from $[d/t]$. Draw $\theta \sim P$. $\forall i \in [k]$, $x^i_{T_r}$ is drawn from μ_θ and $\forall r' \ne r$, $x^i_{T_{r'}}$ is drawn from μ_0.*

The goal of the players is to distinguish which case they are in. Here, all parties communicate using a shared blackboard, and are allowed to use public as well as private randomness.

Given any C-bit communication protocol Π, we use $\Pi = (\Pi_0, \Pi_1, \ldots, \Pi_C)$ to also denote the transcript, that is, the concatenation of the public randomness with all the messages written on blackboard during the execution of Π. In the lemma below, we will measure the information complexity with respect to the No distribution.

Lemma 4.5. *For any communication protocol Π that solves the k-party General Planted Problem, with probability at least 0.9, we have that,*

$$\mathrm{I}\left(\mathbf{\Pi}; X^1, X^2, \ldots, X^k\right) \geq \Omega\left(\frac{d}{c \cdot t}\right).$$

Here, $\forall i \in [k]$, X^i is distributed according to the No case, and $\mathbf{\Pi}$ is the distribution over transcripts, which depends on the input distribution and randomness used by the protocol.

Using the generalized distributed data processing inequality proven in Section 4.2, we can show an $\Omega(1/c)$ bound on the information complexity of any communication protocol that distinguishes between the two cases when $d = t$, that is, all players get a t-dimensional vector drawn either from μ_0 or from μ_θ (where $\theta \sim P$). We can then prove Lemma 4.5 using a direct-sum argument. We first state the result for when $t = d$.

Lemma 4.6. *(Corollary of Theorem 4.8 and Lemma 4.12) Let $t > 0$, $k > 1$ and $\mu_0, \{\mu_\theta\}_{\theta \in \Omega}$ be distributions on t-dimensional vectors in $\mathcal{X}^t$. Let P be a distribution over parameter space Ω such that $\mathbb{E}_{\theta \sim P} \mu_\theta \leq c \cdot \mu_0$. Let Π be a k-party communication protocol that distinguishes between the following two cases, with probability at least 0.9:*

1. *(No) All players get a vector independently drawn from μ_0.*
2. *(Yes) θ is first drawn from P. All players then get a vector independently drawn from μ_θ.*

Then,

$$\mathrm{I}\left(\mathbf{\Pi}; Y^1, Y^2, \ldots, Y^k\right) \geq \Omega\left(1/c\right).$$

Here, $\forall i \in [k]$, Y^i — the input to the i-th player — is distributed according to the No case, and $\mathbf{\Pi}$ is the distribution over transcripts, which depends on the input distribution and randomness used by the protocol.

Proof of Lemma 4.5. The proof follows from a standard direct sum argument. Let Π be a protocol that solves the k-party General Planted Problem, with success probability 0.9. Using Π, we will construct a protocol Π' that distinguishes between the cases when all players get a t-dimensional vector drawn either from μ_0 or from μ_θ (where $\theta \sim P$), with probability 0.9. We will also show that the information complexity of Π' *w.r.t.* the No distribution is at most t/d times the information complexity of Π *w.r.t.* the No distribution. Formally, let $X^1, X^2, \ldots, X^k \in \mathcal{X}^d$ be independently drawn from the No distribution for the k-party General Planted Problem. Let $Y^1, \ldots, Y^k \in \mathcal{X}^t$ be independently drawn from μ_0. Then, we will prove that

$$\mathrm{I}\left(\mathbf{\Pi}'; Y^1, Y^2, \ldots, Y^k\right) \leq \frac{t}{d} \cdot \mathrm{I}\left(\mathbf{\Pi}; X^1, X^2, \ldots, X^k\right).$$

Here, $\mathbf{\Pi}'$ and $\mathbf{\Pi}$ are distributions over transcripts when the inputs are drawn from Y and X respectively. Hence, Lemma 4.5 follows from Lemma 4.6.

Protocol Π' Let $y^1, y^2, \ldots, y^k \in \mathcal{X}^t$ be the input to the k-players. Π' first samples j uniformly at random from $[d/t]$ using public randomness. $\forall i \in [k]$, the i-th player prepares a d-dimensional vector $\tilde{x}^i$ as follows: set $\tilde{x}^i_{T_j} = y^i$ and for $j' \neq j$, draw $\tilde{x}^i_{T_{j'}}$ from μ_0 using private randomness. Recall that $\mathcal{T} = \{T_r\}_{r\in[d/t]}$ is a partition of $[d]$ into t sized sets. All players run Π on inputs $\tilde{x}^1, \ldots, \tilde{x}^k$ and answer whatever Π answers. We will represent the corresponding random variables by capital letters.

Let us first calculate the success probability of Π', which is the average of the probabilities that Π' outputs "No" when the input to each player is *i.i.d.* μ_0 (No distribution) and Π' outputs "Yes" when the input to each player is *i.i.d.* μ_θ (where θ is in turn drawn from P, the Yes distribution).

$$\begin{aligned}
&\frac{1}{2}\cdot\left(\Pr_{\forall i, y^i\sim\mu_0}\left[\Pi'(y^1,y^2,\ldots,y^k)=\text{"No"}\right]+\Pr_{\theta\sim P;\forall i, y^i\sim\mu_\theta}\left[\Pi'(y^1,y^2,\ldots,y^k)=\text{"Yes"}\right]\right)\\
&=\frac{1}{2}\cdot\left(\Pr_{\forall i,y^i\sim\mu_0;\ j\in_R[d/t];\ \forall i,\ \tilde{x}^i_{T_j}=y^i,\ \tilde{x}^i_{T_{j'}}\sim\mu_0\forall j'\neq j}\left[\Pi(\tilde{x}^1,\tilde{x}^2,\ldots,\tilde{x}^k)=\text{"No"}\right]\right.\\
&\qquad\left.+\Pr_{\theta\sim P;\forall i,y^i\sim\mu_\theta;\ j\in_R[d/t];\ \forall i,\ \tilde{x}^i_{T_j}=y^i,\ \tilde{x}^i_{T_{j'}}\sim\mu_0\forall j'\neq j}\left[\Pi(\tilde{x}^1,\tilde{x}^2,\ldots,\tilde{x}^k)=\text{"Yes"}\right]\right)\\
&=\frac{1}{2}\cdot\left(\Pr_{\forall i,j,\ x^i_{T_j}\sim\mu_0}\left[\Pi(x^1,x^2,\ldots,x^k)=\text{"No"}\right]+\Pr_{\theta\sim P;j\in_R[d/t];\forall i,\ x^i_{T_j}\sim\mu_\theta,\ x^i_{T_{j'}}\sim\mu_0\forall j'\neq j}\left[\Pi(x^1,x^2,\ldots,x^k)=\text{"Yes"}\right]\right).
\end{aligned}$$

The last line is exactly equal to the probability of success for protocol Π to distinguish between No and Yes distributions of the k-party General Planted Problem. Hence, Π' succeeds with probability at least 0.9. Next, we calculate the information complexity of Π' when $Y^1, \ldots, Y^k$ are *i.i.d.* μ_0. Note that by definition of the protocol, the transcript $\mathbf{\Pi}' = (J, \mathbf{\Pi}'_{-J})$, where J is the public randomness used by the protocol Π' to sample uniformly from $[d/t]$, and $\mathbf{\Pi}'_{-J}$ is the subsequent transcript generated when the players simulate Π on the prepared inputs $\tilde{x}_1, \ldots, \tilde{x}_k$. Furthermore, the public randomness J is independent of the input $Y^1, \ldots, Y^k$. Then, we have that

$$\begin{aligned}
I(\mathbf{\Pi}'; Y^1,\ldots,Y^k) &= I(J, \mathbf{\Pi}'_{-J}; Y^1,\ldots,Y^k)\\
&= I(J; Y^1,\ldots,Y^k) + I(\mathbf{\Pi}'_{-J}; Y^1,\ldots,Y^k \mid J) && \text{(chain rule)}\\
&= I(\mathbf{\Pi}'_{-J}; Y^1,\ldots,Y^k \mid J) && (J \text{ independent of } Y^1,\ldots,Y^k)\\
&= \frac{t}{d}\cdot\sum_{j=1}^{d/t} \mathrm{I}\left(\mathbf{\Pi}'_{-j}; \tilde{X}^1_{T_j},\ldots,\tilde{X}^k_{T_j}\right)\\
&= \frac{t}{d}\cdot\sum_{j=1}^{d/t} \mathrm{I}\left(\mathbf{\Pi}; X^1_{T_j},\ldots,X^k_{T_j}\right). && (8)
\end{aligned}$$

The last equality follows from the fact that the joint distribution on $(\mathbf{\Pi}'_{-J}, \tilde{X})$ is the same as the joint distribution on $(\mathbf{\Pi}, X)$ when $Y^1, \ldots, Y^k$ are *i.i.d.* μ_0.

Now, for any $j \in [d/t]$, observe that by the chain rule,

$$\mathrm{I}(\mathbf{\Pi}, \{X^1_{T_{j'}},\ldots,X^k_{T_{j'}}\}_{j'<j}; X^1_{T_j},\ldots,X^k_{T_j}) = \mathrm{I}(\mathbf{\Pi}; X^1_{T_j},\ldots,X^k_{T_j}) + \mathrm{I}(\{X^1_{T_{j'}},\ldots,X^k_{T_{j'}}\}_{j'<j}; X^1_{T_j},\ldots,X^k_{T_j} \mid \mathbf{\Pi})$$

$$= \mathrm{I}(\{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j}; X^1_{T_j}, \ldots, X^k_{T_j}) + \mathrm{I}(\Pi; X^1_{T_j}, \ldots, X^k_{T_j} \mid \{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j}).$$

Since for all $j \in [d/t]$, $X^1_{T_j}, \ldots, X^k_{T_j}$ are independent of $\{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j}$, we get that

$$\mathrm{I}(\Pi; X^1_{T_j}, \ldots, X^k_{T_j}) + \mathrm{I}(\{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j}; X^1_{T_j}, \ldots, X^k_{T_j} \mid \Pi) = \mathrm{I}(\Pi; X^1_{T_j}, \ldots, X^k_{T_j} \mid \{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j})$$

$$\implies \quad \mathrm{I}(\Pi; X^1_{T_j}, \ldots, X^k_{T_j}) \leq \mathrm{I}(\Pi; X^1_{T_j}, \ldots, X^k_{T_j} \mid \{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j}).$$

(non-negativity of mutual information)

Substituting in Equation (8) above, we then get that

$$\begin{aligned}
\sum_{j=1}^{d/t} \mathrm{I}\left(\Pi; X^1_{T_j}, \ldots, X^k_{T_j}\right) &\leq \sum_{j=1}^{d/t} \mathrm{I}\left(\Pi; X^1_{T_j}, \ldots, X^k_{T_j} \mid \{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j'<j}\right) \\
&= \mathrm{I}\left(\Pi; \{X^1_{T_{j'}}, \ldots, X^k_{T_{j'}}\}_{j' \in [d/t]}\right) && \text{(chain Rule)} \\
&= \mathrm{I}\left(\Pi; X^1, X^2, \ldots, X^k\right).
\end{aligned}$$

Plugging this back into Equation (8) completes the proof. □

Now, we are ready to prove the information cost lower bound for multi-pass streaming algorithms that solve Problem 4.1.

4.1 Proof of Theorem 4.3

Let M be a $(p+1)$-pass algorithm that solves the distinguishing problem $\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta \in \Omega}, P, \mathcal{T}, k, n)$ with large enough constant probability, say $1 - \delta$ (recall that, we added another pass that doesn't do any operations to M). This implies that

$$\frac{1}{2}\left(\Pr_{(x^1, x^2, \ldots, x^n) \sim D_0}\left[\mathsf{M}(x^1, x^2, \ldots, x^n) = 0\right] + \Pr_{(x^1, x^2, \ldots, x^n) \sim D_1^{\mathcal{T}}}\left[\mathsf{M}(x^1, x^2, \ldots, x^n) = 1\right]\right) \geq 1 - \delta$$

Recall that $\mathcal{T} = \{T_r\}_{r \in [d/t]}$ is a partition of $[d]$ into t sized sets. Under D_0, $\forall i \in [n], j \in [d/t]$, $x^i_{T_j}$ is drawn from μ_0. Under $D_1^{\mathcal{T}}$, first j is chosen uniformly at random from $[d/t]$, $\theta \sim P$ and R is chosen uniformly from k-sized subsets of $[n]$ (we will use the notation $R \sim \binom{[n]}{k}$ to denote this random process), such that $\forall i \in R$, $x^i_{T_j}$ is drawn from μ_θ, and everything else is drawn as in D_0. Thus, we can rewrite the success probability of M as

$$\frac{1}{2}\left(\Pr_{\forall i,j,\, x^i_{T_j} \sim \mu_0}\left[\mathsf{M}(x^1, x^2, \ldots, x^n) = 0\right] + \right.$$

$$\left. \Pr_{\theta \sim P;\, j \in_R [d/t];\, R \sim \binom{[n]}{k};\, \forall i \in R,\, x^i_{T_j} \sim \mu_\theta;\; x^i_{T_{j'}} \sim \mu_0 \;\forall i, j' \;(i \notin R \vee j' \neq j)}\left[\mathsf{M}(x^1, x^2, \ldots, x^n) = 1\right]\right) \geq 1 - \delta. \tag{9}$$

Let q_R be the success probability of distinguishing between D_0 and $D_1^{\mathcal{T}}$, when the rows where μ_θ is "planted", are fixed to be R, that is,

$$q_R = \frac{1}{2}\left(\Pr_{\forall i,j,\, x^i_{T_j} \sim \mu_0} \left[\mathsf{M}(x^1, x^2, \ldots, x^n) = 0\right] + \right.$$
$$\left. \Pr_{\theta\sim P;\ j\in_R [d/t];\ \forall i\in R,\, x^i_{T_j}\sim\mu_\theta;\ x^i_{T_{j'}}\sim\mu_0\ \forall i,j'\ (i\notin R \vee j'\neq j)} \left[\mathsf{M}(x^1, x^2, \ldots, x^n) = 1\right] \right). \tag{10}$$

By Equation (9), we have that $\mathbb{E}_{R\sim\binom{[n]}{k}}[q_R] \geq 1-\delta$. Let $\delta < 0.01$, and we call a set R good if $q_R \geq 0.9$. Then, with probability of at least 0.5 (over $R \sim \binom{[n]}{k}$), $q_R \geq 1 - 2\delta \geq 0.9$ and R is good. Next, we will show that every good set R, using a reduction to communication protocols for k-party General Planted Problem and Lemma 4.5, contributes $\Omega(d/ct)$ to the multi-pass information cost of M *w.r.t.* D_0. Recall that,

$$MIC(\mathsf{M}) = \sum_{\ell=1}^{p+1}\sum_{i=1}^{n}\sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\leq \ell, j-1)}, \mathsf{M}_{(\leq \ell-1, i)}, P\right)$$
$$+ \sum_{\ell=1}^{p+1}\sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\leq \ell-1, j-1)}, \mathsf{M}_{(\leq \ell-1, i)}, P\right).$$

Fix a good $R = \{i_1, i_2, \ldots, i_k\}$ in sorted order. We will denote R's contribution to $MIC(\mathsf{M})$ by MIC^R, which is defined as

$$MIC^R = \sum_{\ell=1}^{p+1}\sum_{a=1}^{k}\sum_{b=1}^{a-1} \mathrm{I}\left(\mathsf{M}_{(\ell, i_a-1)}\ ;\ X^{i_b} \mid \mathsf{M}_{(\leq \ell, i_b-1)}, \mathsf{M}_{(<\ell, i_a-1)}, P\right) +$$
$$+ \sum_{\ell=1}^{p+1}\sum_{a=1}^{k}\sum_{b=a+1}^{k} \mathrm{I}\left(\mathsf{M}_{(\ell, i_a-1)}\ ;\ X^{i_b} \mid \mathsf{M}_{(<\ell, i_b-1)}, \mathsf{M}_{(<\ell, i_a-1)}, P\right).$$

Using M, we will construct a communication protocol $\Pi = \Pi(R)$ for the k-party General Planted Problem, with success probability at least 0.9, such that the information complexity of Π is less than MIC^R. Lemma 4.5 would then imply that $MIC^R \geq \Omega\left(\frac{d}{c\cdot t}\right)$. We restate this as the following formal claim.

Claim 4.7. *For every good R (where q_R as defined above is at least 0.9), $MIC^R \geq \Omega\left(\frac{d}{c\cdot t}\right)$.*

This claim is proved by converting the streaming algorithm into a communication protocol in the standard way so as to invoke Lemma 4.5, and using calculations similar those used in the proof of Claim 5.4 in [BGL+24]. We defer the details of this proof to Appendix B.

We now show how to use Claim 4.7 to get an $\Omega\left(\frac{n^2 d}{ck^2 t}\right)$ bound on $MIC(\mathsf{M})$. The next argument is almost identical to [BGL+24, Section 5.2], with the difference in how R is sampled.

Notice that since R is good with probability at least 0.5, the claim implies that $\mathbb{E}_{R\sim\binom{[n]}{k}} MIC^R \geq \Omega\left(\frac{d}{c\cdot t}\right)$. We will show that $MIC(\mathsf{M}) \geq \Omega\left(\left(\frac{n}{k}\right)^2 \cdot \mathbb{E}_{R\sim\binom{[n]}{k}} MIC^R\right)$, which would suffice for Theorem 4.3.

We begin by writing

$$\mathbb{E}_{R\sim\binom{[n]}{k}} MIC^R = \mathbb{E}_{R\sim\binom{[n]}{k}} \sum_{\ell=1}^{p+1}\sum_{a=1}^{k}\sum_{b=1}^{a-1} \mathrm{I}\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_b} \mid \mathsf{M}_{(\le\ell,i_b-1)}, \mathsf{M}_{(<\ell,i_a-1)}, P\right) + \\ + \mathbb{E}_{R\sim\binom{[n]}{k}} \sum_{\ell=1}^{p+1}\sum_{a=1}^{k}\sum_{b=a+1}^{k} \mathrm{I}\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_b} \mid \mathsf{M}_{(<\ell,i_b-1)}, \mathsf{M}_{(<\ell,i_a-1)}, P\right).$$

We will compare the first term in the expectation, with the first term of $MIC(\mathsf{M})$, that is,

$$\sum_{\ell=1}^{p+1}\sum_{i=1}^{n}\sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le\ell,j-1)}, \mathsf{M}_{(\le\ell-1,i)}, P\right).$$

For a random R, each term $\mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le\ell,j-1)}, \mathsf{M}_{(\le\ell-1,i)}, P\right)$ for an (i,j) pair with $j \le i$, appears in MIC_R with probability at most $\binom{n-2}{k-2}/\binom{n}{k} = \frac{k(k-1)}{n(n-1)}$, since this happens only if both $i+1$ and j are in R. Similarly, we will compare the second term in expectation with the second term of $MIC(\mathsf{M})$, that is,

$$\sum_{\ell=1}^{p+1}\sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le\ell-1,j-1)}, \mathsf{M}_{(\le\ell-1,i)}, P\right).$$

For a random R, each term $\mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le\ell-1,j-1)}, \mathsf{M}_{(\le\ell-1,i)}\right)$ for an (i,j) pair with $j > i+1$, appears in MIC_R with probability at most $\binom{n-2}{k-2}/\binom{n}{k} = \frac{k(k-1)}{n(n-1)}$, since again, this happens only if both $i+1$ and j are in R. When $j = i+1$, no such term appears in MIC^R, as $b \neq a$. This implies that

$$\mathbb{E}_{R\sim\binom{[n]}{k}} MIC^R \le \frac{k(k-1)}{n(n-1)} \cdot MIC(\mathsf{M}).$$

4.2 Generalized distributed data processing inequalities

Theorem 4.8. *Consider a family of distributions $\{\mu_\theta\} : \mathcal{X} \to [0,1]$ parameterized by a random variable $\boldsymbol{\theta}$, which takes values in some domain Ω and has distribution P. Let $\mu_1 = \mathbb{E}_{\theta\sim P}[\mu_\theta]$. Consider the distributed detection setting where if $V = 0$ then each party receives $X_i \sim \mu_0$ (for some distribution $\mu_0 : \mathcal{X} \to [0,1]$), and if $V = 1$ then we first draw $\theta \sim P$, and then each party receives $X_i \sim \mu_\theta$. If $\mu_1 \le c\mu_0$, then for some constant $K > 0$, for any multi-party communication protocol Π,*

$$\mathbb{E}_{\theta\sim P}\left[h^2(\boldsymbol{\Pi}_{|V=0} \parallel \boldsymbol{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \le K(c+1)I(X;\boldsymbol{\Pi} \mid V=0). \tag{11}$$

Here, $\boldsymbol{\Pi}_{|V=0}$ and $\boldsymbol{\Pi}_{|V=1,\boldsymbol{\theta}=\theta}$ represent the random variables for the transcript of protocol Π, when inputs to the parties are drawn from μ_0 and μ_θ, respectively.

Proof. Our proof builds on the proof of Theorem 3.1 in [BGM+16]. Let Π be an m-party communication protocol ($m \ge 2$). We first note that since X_i's are independent conditioned on $V = 0$, that is, $\mathrm{I}(X_i; X_{<i} \mid V = 0) = 0$, the RHS of (11) tensorizes and we get,

$$\mathrm{I}(X;\boldsymbol{\Pi} \mid V=0) = \sum_{i=1}^{m} \mathrm{I}(X_i;\boldsymbol{\Pi} \mid V=0, X_{<i}) \qquad \text{(Chain Rule)}$$

$$\geq \sum_{i=1}^{m} \mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0). \tag{12}$$

Fix $i \in [m]$. To bound $\mathrm{I}(X_i; \Pi \mid V = 0)$, we consider the following single-machine setting. Fix θ. Let W be a random variable which is uniformly distributed in $\{0, 1\}$. Let data X' be generated as follows: $X'_i \sim \mu_W^\theta$ (where $\mu_0^\theta = \mu_0$ and $\mu_1^\theta = \mu_\theta$) and for any $j \neq i$, $X'_j \sim \mu_0$. We apply the protocol Π on the input X', and consider the resulting transcript Π'. We will also use Π' to denote the one argument randomized function, that takes in x'_i, samples $x'_j \sim \mu_0 \forall j \neq i$, and outputs $\Pi(x)$. Note that the function Π' doesn't depend on θ. Then $W \to X'_i \to \Pi'$ forms a Markov chain, and by the data processing inequality,

$$\mathrm{I}(W; \mathbf{\Pi}') \leq \mathrm{I}(X'_i; \mathbf{\Pi}').$$

Here, $\mathbf{\Pi}'$ denotes the random variable for output of Π'. (When the distribution for input x_i to Π', say $x_i \sim Y$, is not clear from the context, we will use $\Pi'(Y)$ to denote the random variable for output of Π'). Using Lemma 10 in [BGM+16], we can lower bound $\mathrm{I}(W; \mathbf{\Pi}')$ using the squared Hellinger distance,

$$h^2(\mathbf{\Pi}'_{|W=0} \parallel \mathbf{\Pi}'_{|W=1}) \leq \mathrm{I}(W; \mathbf{\Pi}') \quad \implies \quad h^2(\mathbf{\Pi}'_{|W=0} \parallel \mathbf{\Pi}'_{|W=1}) \leq \mathrm{I}(X'_i; \mathbf{\Pi}'). \tag{13}$$

Equation (13) relates the mutual information and squared Hellinger distance for the single machine case; next we want to relate the single machine setting to the distributed setting. To do this, we first establish some notation. For any fixed vector $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in \{0, 1\}^m$, let $\mu_{\mathbf{b}}^\theta$ denote the product distribution on m inputs, where input to each machine i is drawn independently from $\mu_{\mathbf{b}_i}^\theta$. We use notation $\mathbf{\Pi}_{\mathbf{b}}^\theta$ to denote the random variable for transcript of protocol Π when inputs $(x_1, \ldots, x_m) \sim \mu_{\mathbf{b}}^\theta$.

With this notation, we note that the random variable $\mathbf{\Pi}'_{|W=0}$ has distribution as $\mathbf{\Pi}_{\mathbf{0}}^\theta$. And $\mathbf{\Pi}'_{|W=1}$ has the same distribution as $\mathbf{\Pi}_{\mathbf{e}_i}^\theta$. Here, $\mathbf{e}_i$ is the standard basis vector with 1 at the ith coordinate. Then we can rewrite (13) as,

$$h^2(\mathbf{\Pi}_{\mathbf{0}}^\theta \parallel \mathbf{\Pi}_{\mathbf{e}_i}^\theta) \leq \mathrm{I}(X'_i; \mathbf{\Pi}') = \mathrm{I}(X_i'^\theta; \Pi'(X_i'^\theta)). \tag{14}$$

By taking expectation over $\theta \sim P$, we get

$$\mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{\mathbf{0}}^\theta \parallel \mathbf{\Pi}_{\mathbf{e}_i}^\theta)\right] \leq \mathbb{E}_{\theta \sim P}\left[\mathrm{I}(X_i'^\theta; \Pi'(X_i'^\theta))\right]. \tag{15}$$

Next, we will show that $\mathbb{E}_{\theta \sim P}\left[\mathrm{I}(X_i'^\theta; \Pi'(X_i'^\theta))\right] \leq \frac{c+1}{2} \cdot \mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0)$. Let Y be a random variable that takes values according to μ_0. First, note that

$$\mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0) = \mathrm{I}(Y; \Pi'(Y)),$$

as conditioned on $V = 0$, $\forall i \in [m], X_i \sim \mu_0$; thus the joint distribution of (X_i, Π) is identical to $(Y, \Pi'(Y))$. Hence, it suffices to prove that $\mathbb{E}_{\theta \sim P}\left[\mathrm{I}(X_i'^\theta; \Pi'(X_i'^\theta))\right] \leq \frac{c+1}{2} \cdot \mathrm{I}(Y; \Pi'(Y))$.

We will first prove a generalization of Lemma 11 in [BGM+16], and then come back to proving the above inequality.

Lemma 4.9. *Consider a family of distributions $\{\mu'_s\}$ parameterized by $s \sim S$, and let $\mu' = \mathbb{E}_{s \sim S}\left[\mu'_s\right]$. For some distribution μ, let $\mu \geq c\mu'$. Let $f(z)$ be a random function that depends only on z, and whose range is discrete. If $Z \sim \mu$ and $s \sim S, Z' \sim \mu'_s$, then we have that,*

$$\mathrm{I}(Z; f(Z)) \geq c \cdot \mathrm{I}(Z'; f(Z') \mid S).$$

Proof. Since f is a random function which depends only on z and $\mu \geq c \cdot \mu'$, we have

$$\mathrm{I}(Z; f(Z)) = \mathbb{E}_{z\sim Z}\left[D_{KL}(f(z) \parallel f(Z))\right] = \mathbb{E}_{z\sim\mu}\left[D_{KL}(f(z) \parallel f(Z))\right] \geq c \cdot \mathbb{E}_{z\sim\mu'}\left[D_{KL}(f(z) \parallel f(Z))\right]. \tag{16}$$

Note that,

$$\begin{aligned}
&\mathbb{E}_{z\sim\mu'}\left[D_{KL}(f(z) \parallel f(Z))\right] = \mathbb{E}_{s\sim S}\mathbb{E}_{z\sim\mu'_s}\left[D_{KL}(f(z) \parallel f(Z))\right] \\
&= \mathbb{E}_{s\sim S}\int_z \left[\sum_\pi \Pr[f(z)=\pi]\log\left(\frac{\Pr[f(z)=\pi]}{\int_z \Pr[f(z)=\pi]d\mu(z)}\right)\right] d\mu'_s(z) \\
&= \mathbb{E}_{s\sim S}\int_z \left[\sum_\pi \Pr[f(z)=\pi]\log\left(\frac{\Pr[f(z)=\pi]}{\int_z \Pr[f(z)=\pi]d\mu'_s(z)}\right)\right] d\mu'_s(z) \\
&\quad + \mathbb{E}_{s\sim S}\int_z \left[\sum_\pi \Pr[f(z)=\pi]\log\left(\frac{\int_z \Pr[f(z)=\pi]d\mu'_s(z)}{\int_z \Pr[f(z)=\pi]d\mu(z)}\right)\right] d\mu'_s(z) \\
&= \mathbb{E}_{s\sim S}\mathbb{E}_{z\sim Z'_{|S=s}}\left[D_{KL}(f(z) \parallel f(Z'_{|S=s}))\right] + \mathbb{E}_{s\sim S}\sum_\pi \left[\log\left(\frac{\int_z \Pr[f(z)=\pi]d\mu'_s(z)}{\int_z \Pr[f(z)=\pi]d\mu(z)}\right)\int_z \Pr[f(z)=\pi]d\mu'_s(z)\right] \\
&= \mathrm{I}(Z'; f(Z') \mid S) + \mathbb{E}_{s\sim S} D_{KL}(f(Z'_{|S=s}) \parallel f(Z)).
\end{aligned}$$

Since KL-divergence is always non-negative, plugging into Equation (16), we get that

$$\mathrm{I}(Z; f(Z)) \geq c \cdot \mathrm{I}(Z'; f(Z') \mid S). \qquad \square$$

We now apply Lemma 4.9 with θ as the parameterization. We take $Z = Y$; $\mu = \mu_0$. We take Z' conditioned on parameter θ to be $X_i'^{\theta}$; thus, $\mu'_s = \frac{\mu_0+\mu_\theta}{2}$ and $\mu' = \mathbb{E}_{\theta\sim P}\frac{\mu_0+\mu_\theta}{2} = \frac{\mu_0+\mu_1}{2}$. Since $\mu_1 \leq c\mu_0$, $\mu_0 \geq \frac{2}{c+1}\left(\frac{\mu_0+\mu_1}{2}\right)$. As Π' is a randomized function only of x_i, Lemma 4.9 says that,

$$\mathbb{E}_{\theta\sim P}\mathrm{I}(X_i'^{\theta}; \Pi'(X_i'^{\theta})) \leq \frac{c+1}{2} \cdot \mathrm{I}(Y; \Pi'(Y)).$$

Plugging this into (15), and using the fact that $\mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0) = \mathrm{I}(Y; \Pi'(Y))$, we get,

$$\mathbb{E}_{\theta\sim P}\left[h^2(\mathbf{\Pi}_{\mathbf{0}}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{e}_i}^{\theta})\right] \leq \frac{c+1}{2} \cdot \mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0). \tag{17}$$

Next, we lower bound the LHS of (17). In the following claim, we first show that the distributions $\mathbf{\Pi}_{\mathbf{b}}^{\theta}$ (over the transcripts under protocol Π) satisfies the cut-paste property developed in [BYJKS04] and used in [BGM+16], because after fixing θ the inputs to each machine are independent. The proof relies on basic properties of transcripts established in [BGM+16] and is deferred to Appendix B.

Claim 4.10 (Cut-paste property of the protocol)**.** *For any θ and transcript π, and any $\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4$ with $\{b_i^1, b_i^2\} = \{b_i^3, b_i^4\}$ (in a multi-set sense) for every $i \in [m]$,*

$$\Pr\left[\mathbf{\Pi}_{\mathbf{b}^1}^{\theta} = \pi\right] \cdot \Pr\left[\mathbf{\Pi}_{\mathbf{b}^2}^{\theta} = \pi\right] = \Pr\left[\mathbf{\Pi}_{\mathbf{b}^3}^{\theta} = \pi\right] \cdot \Pr\left[\mathbf{\Pi}_{\mathbf{b}^4}^{\theta} = \pi\right],$$

and therefore,

$$h^2\left(\mathbf{\Pi}_{\mathbf{b}^1}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{b}^2}^{\theta}\right) = h^2\left(\mathbf{\Pi}_{\mathbf{b}^3}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{b}^4}^{\theta}\right).$$

We now use a result for transcript distributions that satisfy the cut-paste property.

Claim 4.11 (Theorem E.1 in [BGM+16], corollary of Theorem 7 in [Jay09])**.** *Suppose a family of distribution $\{P_{\mathbf{b}} : \mathbf{b} \in \{0,1\}^m\}$ satisfies the cut-paste property: for any $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}, \mathbf{d}$ with $\{a_i, b_i\} = \{c_i, d_i\}$ (in a multi-set sense) for every $i \in [m]$, $h^2(P_{\mathbf{a}}, P_{\mathbf{b}}) = h^2(P_{\mathbf{c}}, P_{\mathbf{d}})$. Then we have*

$$\sum_{i=1}^{m} h^2(P_{\mathbf{0}}, P_{\mathbf{e}_i}) \geq \Omega(1) \cdot h^2(P_{\mathbf{0}}, P_{\mathbf{1}}),$$

where $\mathbf{0}$ and $\mathbf{1}$ are all 0's and all 1's vectors respectively, and $\mathbf{e}_i$ is the unit vector that only takes 1 in the i th entry.

Using Claim 4.11 and Claim 4.10,

$$h^2(\mathbf{\Pi}_{\mathbf{0}}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{1}}^{\theta}) \leq O(1) \sum_{i=1}^{m} h^2(\mathbf{\Pi}_{\mathbf{0}}^{\theta} \parallel \Pi_{\mathbf{e}_i}^{\theta}),$$

$$\implies \mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{\mathbf{0}}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{1}}^{\theta})\right] \leq O(1) \sum_{i=1}^{m} \mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{\mathbf{0}}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{e}_i}^{\theta})\right].$$

Using (17) to simplify the RHS above,

$$\begin{aligned}
\mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{\mathbf{0}}^{\theta} \parallel \mathbf{\Pi}_{\mathbf{1}}^{\theta})\right] &\leq O(1) \sum_{i=1}^{m} \frac{c+1}{2} \cdot \mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0), \\
&\leq O(1) \frac{c+1}{2} \cdot \sum_{i=1}^{m} \mathrm{I}(X_i; \mathbf{\Pi} \mid V = 0) \\
&\leq O(1) \frac{c+1}{2} \mathrm{I}(X; \mathbf{\Pi} \mid V = 0),
\end{aligned}$$

where in the last step we use the tensorization from (12). Note that the distribution $\mathbf{\Pi}_{\mathbf{0}}^{\theta}$ is identical to $\mathbf{\Pi}_{|V=0}$, for all θ. And distribution $\Pi_{\mathbf{1}}^{\theta}$ is identical to $\mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta}$. Therefore for some constant $K > 0$ we get,

$$\mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \leq K(c+1)\mathrm{I}(X; \mathbf{\Pi} \mid V = 0),$$

proving the theorem. □

Finally, we prove that, for any protocol Π that solves the distributed detection problem with probability at least 0.9, the expected hellinger distance as in Theorem 4.8 is $\Omega(1)$. The proof of this result is a calculation and is deferred to Appendix B.

Lemma 4.12. *Consider a family of distributions $\{\mu_\theta\} : \mathcal{X} \to [0,1]$ parameterized by a random variable $\boldsymbol{\theta}$, which takes values in some domain Ω and has distribution P. Consider the distributed detection setting where if $V = 0$ then each party receives $X_i \sim \mu_0$ (for some distribution $\mu_0 : \mathcal{X} \to [0,1]$), and if $V = 1$ then we first draw $\theta \sim P$, and then each party receives $X_i \sim \mu_\theta$. Suppose there is an m-party communication protocol Π that detects whether $V = 0$ or $V = 1$ with probability at least 0.9. Then*

$$\mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \geq \Omega(1).$$

5 Multi-pass Streaming Lower Bound for Bi-Clique

In this section, we will prove multi-pass memory lower bounds for detecting planted bi-cliques in random bipartite graphs. Formally, we study the following distinguishing problem.

Problem 5.1 (Planted Bi-Clique). *Let $1 < k \leq \min(m, n)$, and $0 < q \leq 1/2$. The goal is to distinguish between the following joint distributions on n-bit vectors $x^1, \ldots, x^m$:*

1. *D_{uniform}: $\forall i \in [m], \forall j \in [n]$, x^i_j is drawn as $Ber(q)$.*
2. *D_{planted}: $S \subseteq [n]$ is drawn uniformly at random from all subsets of $[n]$ of size k. R is drawn uniformly at random from all subsets of $[m]$ of size k.*

   $\forall i \notin R, \forall j \in [n]$, x^i_j is drawn as $Ber(q)$.

   $\forall i \in R, \forall j \in S$, $x^i_j = 1$, and $\forall j \notin S$, x^i_j is drawn as $Ber(q)$.

Our main hardness result for Problem 5.1 is the following:

Theorem 5.2 (Memory Lower Bound for Planted Bi-clique). *Let $0 < q \leq 1/2$ and $0 < k < O\left(\sqrt{\frac{q \cdot n}{\log(nm)}}\right)$. Any p-pass streaming algorithm (using public as well as private randomness), that distinguishes between D_{uniform} and D_{planted} (as in Problem 5.1) when $x^1, x^2, \ldots, x^m$ arrive in a stream requires at least $\Omega\left(\frac{nmq}{pk^4 \log(nm)}\right)$ bits of memory.*

Remark 5.3. *It is straightforward to modify the above theorem to obtain a memory lower bound of $\Omega\left(\frac{nmq}{pk'^2k^2 \log(nm)}\right)$ for any p-pass streaming algorithm detecting cliques of size $(k' \times k)$ in $G(m, n, q)$. Taking $k' = \frac{k}{n}m$ yields the bound stated in Equation (1). The only subtlety is that Problem 1.1 is a distributional version, whereas the target statement concerns exact detection of a clique of size $(k' \times k)$. This can be resolved by noting that any algorithm for the distributional version works for some $k' \approx \frac{k}{n}m$.*

Our objective will be to frame Problem 5.1 as an instantiation of Problem 4.1, and thereafter leverage the lower bound for the general problem. For this, we will require partitioning $[n]$ into n/t subsets of size $t \geq \Omega((k^2 \log(nm))/q)$. We define the distributions $\mu_0, \{\mu_\theta\}_{\theta \in \Omega}$ over such t-sized subsets in terms of the following specialized distributions P^0_{trunc} and $\left\{P^{1,S}_{\text{trunc}}\right\}_S$

Let $C > 0$ be a large enough constant. We define P^0_{trunc} to be the uniform distribution over the set

$$T := \left\{x \in \{0,1\}^t : |x| \in \left[tq - C\sqrt{tq \log(nm)}, tq + C\sqrt{tq \log(nm)}\right]\right\}. \tag{18}$$

In words, this set comprises of all t-bit vectors x that have $|x|$ in the *typical range* of a $Bin(t, q)$ random variable. Additionally, for $S \subseteq [t]$, $|S| = k$, we define $P^{1,S}_{\text{trunc}}$ to be the uniform distribution over the set

$$T_S := \left\{x \in \{0,1\}^t : x_S = \overrightarrow{1}, |x| \in \left[tq - C\sqrt{tq \log(nm)}, tq + C\sqrt{tq \log(nm)}\right]\right\}. \tag{19}$$

In words, this set comprises of all t-bit vectors x that have S set to 1, and have $|x|$ in the same typical range as the support of P^0_{trunc}.

For the distributions P^0_{trunc} and $P^{1,S}_{\text{trunc}}$ thus defined, we can derive the technical condition necessary in Theorem 4.2 about $\mathbb{E}_S\left[P^{1,S}_{\text{trunc}}\right]$ being pointwise upper-bounded by P^0_{trunc}.

Claim 5.4. *Let $t \geq \frac{Ck^2 \log(nm)}{q}$ for some large enough constant C. Suppose S is drawn uniformly at random from all subsets of $[t]$ of size k. Let μ_0 and μ_1 be the probability mass functions of P^0_{trunc} and $\mathbb{E}_S[P^{1,S}_{\text{trunc}}]$ respectively. Then,*

$$\mu_1 \leq O(1) \cdot \mu_0.$$

The proof of Claim 5.4 is a calculation, and is deferred to Appendix C. It crucially uses two properties: that S is chosen uniformly at random over subsets of $[t]$, together with the fact that the sparsity of vectors in the supports of both P^0_{trunc} and $P^{1,S}_{\text{trunc}}$ are constrained to be in the typical range of a $Bin(t, q)$ random variable.

Now, we define the following distinguishing problem defined over a given fixed partition of $[n]$, when all sub-vectors in each vector in the stream are "typical", and also, the planted set of coordinates (in the planted distribution) belongs wholly to a single t-sized partition. As we will prove formally, any algorithm that solves Problem 5.1 also solves the following distinguishing problem.

Problem 5.5 (Partition Planted Bi-Clique)**.** *Let $0 < k \leq \min(m, n)$ and $\frac{Ck^2 \log(nm)}{q} \leq t \leq n$, where C is a large enough constant. Let $n' = t \cdot \left\lfloor \frac{n}{t} \right\rfloor$. Let $\mathcal{T} = \{T_r\}_{r \in [n'/t]}$ be a partition of $[n']$, where $\forall r, |T_r| = t$. The goal is to distinguish between the following joint distributions on n'-bit vectors $x^1, \ldots, x^m$:*

1. *D_0: $\forall i \in [m]$ and $\forall r \in [n'/t]$, $x^i_{T_r}$ is drawn from P^0_{trunc}.*
2. *$D_1^{\mathcal{T}}$: Draw r uniformly from $[n'/t]$. $\forall i \in [m]$ and $\forall r' \neq r$, $x^i_{T_{r'}}$ is drawn from P^0_{trunc}.*
   Draw a uniformly random subset $S \subseteq T_r$ of size k.
   Draw a uniformly random subset $R \subseteq [m]$ of size k.
   $\forall i \notin R$, $x^i_{T_r}$ is drawn from P^0_{trunc}, whereas, $\forall i \in R$, $x^i_{T_r}$ is drawn from $P^{1,S}_{\text{trunc}}$.

Informally, under distribution D_0, at each time-step, x^i is drawn from the uniform distribution on n'-bit vectors conditioned on the number of ones in each partition being typical. On the other hand, under distribution $D_1^{\mathcal{T}}$, for all but k time-steps, x^i is drawn as in distribution D_0; otherwise, some partition in x^i is drawn from the planted distribution while still conditioning on the number of ones being typical. Note again that we assume $k << \sqrt{t}$.

Problem 5.5 fits into the framework of Problem 4.1, and we can therefore show the following hardness result for it.

Lemma 5.6 (Memory Lower Bound for Partition Planted Bi-Clique)**.** *Let $0 < k \leq \min(m, n)$ and $\frac{Ck^2 \log(nm)}{q} \leq t \leq n$, where C is a large enough constant. Let $n' = t \cdot \left\lfloor \frac{n}{t} \right\rfloor$. Let $\mathcal{T} = \{T_r\}_{r \in [n'/t]}$ be a partition of $[n']$, where $\forall r, |T_r| = t$. Then, any p-pass streaming algorithm (using public as well as private randomness), that distinguishes between D_0 and $D_1^{\mathcal{T}}$ (as defined in Problem 5.5) requires at least $\Omega\left(\frac{mn'}{pk^2t}\right)$ bits of memory.*

Proof. Observe that Problem 5.5 is a specific instantiation of Problem 4.1 with $n = m$ and $d = n'$. Let μ_0, μ_1 denote the probability mass functions of P^0_{trunc} and $\mathbb{E}_S[P^{1,S}_{\text{trunc}}]$ respectively. Claim 5.4 shows that $\mu_1 \leq O(1) \cdot \mu_0$, which satisfies the assumption of Theorem 4.2. The result follows. □

With Lemma 5.6 established, we now sketch how Theorem 5.2 is derived. The proof is a reduction: given a low-memory streaming algorithm $\mathcal{A}$ for Problem 5.1, we obtain a low-memory streaming algorithm for Problem 5.5, with the partition size t set to $t = \left\lceil \frac{Ck^2 \log(nm)}{q} \right\rceil$. For simplicity, assume that t divides n. The high-level strategy is as follows: Given an input from Problem 5.5, $\mathcal{A}'$ first uses public randomness to permute the coordinates of all the inputs consistently according to a uniformly random permutation, and feeds it to $\mathcal{A}$. This has the effect of turning the fixed partition into a uniformly random partition of $[n]$. Under the null, every group in the partition for each input is a draw from P^0_{trunc}. We now realize that the inputs under the null distribution of Problem 5.1 follow the same distribution, except that every group in the partition for each input is a draw from $\{0,1\}^t$ where every bit is drawn as $\text{Ber}(q)$. Since P^0_{trunc} is the uniform distribution over the subset of $\{0,1\}^t$ that is typical, these distributions are close. Conversely, under the planted distribution, we have that precisely k inputs each have a group in the partition which is a draw from $P^{1,S}_{\text{trunc}}$, where S is a random subset of size k within the group. All the other groups are draws from P^0_{trunc}. Again, we realize that the inputs under the planted distribution of Problem 5.1 follow the same distribution, except that the planted groups are drawn from $\{0,1\}^t$ where every bit is drawn as $\text{Ber}(q)$, and then a uniformly random subset of size k is forced to 1. The typical support of this distribution is precisely the support T_S of $P^{1,S}_{\text{trunc}}$, and hence we can again show that the planted distributions of both the problems are close in TV distance. So, $\mathcal{A}'$ can solve Problem 5.5 by simply returning the output of $\mathcal{A}$. The formal details are given in Appendix C.

5.1 Application: Densest at-most β Subgraph

Theorem 5.2 allows us to derive a hardness of approximation result for the "Densest at-most β Subgraph Problem" (see Section 3.12 in [LMFB24]) in the *Vertex Arrival* streaming model. We define the model and problem here.

Definition 5.7 (Vertex Arrival Streaming Model)**.** *In the vertex arrival streaming model, the algorithm is presented with vertices from an undirected graph, and their neighbors amongst previously revealed vertices in an arbitrary, worst-case order. That is, the algorithm sees a stream $\{(v^i, E_{\le i})\}_{i \le n}$, where $E_{\le i}$ only contains edges that v^i shares with vertices $v^1, \ldots, v^i$.*

Problem 5.8 (Densest at-most β Subgraph)**.** *Consider an undirected graph $G = (V, E)$ on n vertices (self-edges allowed), and let $1 \le \beta \le n$. For any subset $H \subseteq V$, let $G(H) = (H, E(H))$ be the induced subgraph (i.e., $G(H)$ has vertex set H and all edges $(u, v) \in E$ that satisfy $u, v \in H$). The edge density of $G(H)$ is defined as $\frac{|E(H)|}{|H|}$. The goal is to approximate the largest edge density among all subgraphs of size at most β in G, i.e., $\max_{H \subseteq V, 1 \le |H| \le \beta} \left\{ \frac{|E(H)|}{|H|} \right\}$. For $\alpha \ge 1$, an α-approximation to a quantity y is any number x such that $(1/\alpha)y \le x \le y$.*

Corollary 5.9 (Memory Lower Bound for Densest at-most β Subgraph)**.** *Consider any $\alpha \ge 1$ and $200\alpha \log n \le \beta \le o\left(\frac{n}{\alpha^2 \log^2 n}\right)$. Any p-pass streaming algorithm that approximates the size of the largest edge density among all subgraphs of size at most β in an undirected graph that is presented in the Vertex Arrival Model to a factor α requires at least $\tilde{\Omega}\left(\frac{n^2}{p\beta\alpha^4}\right)$ bits of memory.*

Proof. We will reduce from the planted bi-clique problem with $m = n$, and an appropriate choice of k and q. Let $\mathcal{A}$ be a p-pass streaming algorithm that uses $\tilde{o}\left(\frac{n^2}{p\beta\alpha^4}\right)$ bits of memory and always

approximates the size of the densest at-most β subgraph in a graph presented in the Vertex Arrival Model to a factor α. Using $\mathcal{A}$, we will construct a p-pass streaming algorithm $\mathcal{A}'$ that processes $x^1, \ldots, x^n$ arriving in a stream, which solves Problem 5.1 for $k = 600\alpha \log n$ and $q = \frac{\log n}{\beta}$, while using only $\tilde{o}\left(\frac{n^2}{p\beta\alpha^4}\right) = o\left(\frac{n^2 q}{pk^4 \log n}\right)$ bits of memory. This would contradict Theorem 5.2, and give us the claimed result.

The algorithm $\mathcal{A}'$ operates as follows. Given an input stream $x^1, x^2, \ldots, x^n$, $\mathcal{A}'$ interprets each x^i in the stream as a vertex v^i in an undirected graph G, and presents it to $\mathcal{A}$ in the Vertex Arrival Model. For each v^i, it will read off connectivity to $v^1, \ldots, v^i$ from $x^i_{[1:i]}$. That is, for $j \le i$, there is an undirected edge between v^j and v^i iff $x^i_j = 1$. Note that $\mathcal{A}'$ can simulate this input space-efficiently for $\mathcal{A}$ (it only needs to keep track of a counter).

Now, suppose $x^1, x^2, \ldots, x^n$ are drawn from D_{uniform}. Then, observe that the graph G that $\mathcal{A}'$ presents to $\mathcal{A}$ is a random graph, where every (v^i, v^j) is connected by an edge with probability q. We claim that the maximum edge density in this graph is at most $O(\log n)$ with high probability. To see this, observe that for every fixed $H \subseteq V$ of size at most β, we have that $\mathbb{E}\left[|E(H)|\right] = |H|^2 q$, since $|E(H)|$ is precisely the sum of $|H|^2$ independent $Ber(q)$ random variables. By a Chernoff bound, we have that

$$\Pr\left[|E(H)| \ge (1+\delta)|H|^2 q\right] \le \exp\left(-\frac{|H|^2 q \delta^2}{2+\delta}\right).$$

Plugging in $\delta = \frac{100 \log n}{|H|q}$, we get that

$$\Pr\left[|E(H)| \ge |H|^2 q + 100|H| \log n\right] \le \exp\left(-\Omega\left(|H|^2 q \cdot \frac{\log n}{|H|q}\right)\right) = \exp\left(-\Omega\left(|H| \log n\right)\right) \le n^{-2|H|}.$$

Therefore, with probability at least $1 - n^{-2|H|}$, $|E(H)|$ is at most

$$|H|^2 q + 100|H| \log n = |H| \cdot |H|q + 100|H| \log n \le 101|H| \log n,$$

where we plugged in $q = \frac{\log n}{\beta}$ and used $|H| \le \beta$ in the last inequality. By a union bound, the probability that $|E(H)| \le 101|H| \log n$ for *every* $H \subseteq [n], |H| \le \beta$ is at most

$$\sum_{i=1}^{\beta} \binom{n}{i} \cdot n^{-2i} \le \sum_{i=1}^{\beta} n^{-i} \le O\left(\frac{1}{n}\right).$$

This means that the edge density of every $H \subseteq [n], |H| \le \beta$ is at most $\frac{|E(H)|}{|H|} \le 101 \log n$ with probability $O(1/n)$, which means that the output of $\mathcal{A}$ when $\mathcal{A}'$ presents this undirected random graph G to it will be at most $101 \log n$.

On the other hand, suppose $x^1, x^2, \ldots, x^n$ are drawn from D_{planted}. Recall that S, R are uniformly random subsets of $[n]$ drawn without replacement of size k. By Hoeffding's bound (e.g., Proposition 1.2 in [BM15]), the size of $\{i \in S : i \ge n/2\}$ is at least $\frac{k}{3}$ with probability at least $1 - e^{-\Omega(k)}$. Similarly, the size of $\{j \in R : j \le n/2\}$ is at least $\frac{k}{3}$ with probability at least $1 - e^{-\Omega(k)}$. Together with a union bound, we get that the size of both these sets is at least $\frac{k}{3}$ with probability at least $1 - e^{-\Omega(k)}$. But then note that conditioned on this event, in the undirected graph G that $\mathcal{A}'$ presents to $\mathcal{A}$, at least $\frac{k}{3}$ vertices in $v^{n/2}, \ldots, v^n$ are all connected to at least $\frac{k}{3}$ vertices in $v^1, \ldots, v^{n/2}$. Note also

that $k/3 \le \beta$, and hence the maximum edge density amongst at most β-sized subgraphs of G is at least $\frac{k^2/9}{k/3} = k/3$, meaning that output of $\mathcal{A}$ will be at least $k/3\alpha = 200 \log n$.

Therefore, $\mathcal{A}'$ can distinguish between D_{uniform} and D_{planted} with constant advantage by checking if the output of $\mathcal{A}$ is at most $101 \log n$ or at least $200 \log n$. It does so using only $\tilde{o}\left(\frac{n^2}{p\beta\alpha^4}\right) = o\left(\frac{n^2 q}{pk^4 \log n}\right)$ bits of memory, which gives us the desired contradiction.

□

6 Multi-pass Streaming Lower Bounds in the Semi-random Model

In this section, we will prove multi-pass memory lower bounds for detecting planted bi-cliques in random bipartite graphs under the presence of a monotone adversary. While the general outline will follow that of the previous section, we will require making subtle and crucial updates, which will allow us to prove a *stronger* lower bound for this model.

Formally, we will study the following distinguishing problem.

Problem 6.1 (Semi-random Planted Bi-Clique)**.** *Let $0 < k_1, k_2 \le n$. Consider the following joint distributions on n-bit vectors $x^1, \ldots, x^n$:*

1. *D_{uniform}: $\forall i \in [n]$, x^i is drawn from the uniform distribution over $\{0,1\}^n$.*
2. *D_{planted}: $S \subseteq [n]$ is drawn uniformly at random from all subsets of $[n]$ of size k_2. R is drawn uniformly at random from all subsets of $[n]$ of size k_1.*

   $\forall i \notin R$, x^i is drawn from the uniform distribution over $\{0,1\}^n$.
   $\forall i \in R$, $\forall j \in S$ $x^i_j = 1$, and $\forall j \notin S$ x^i_j is a uniform $\{0,1\}$ bit.

Let A be a matrix with rows as $x^1, \ldots, x^n$, we consider it as an adjacency matrix of a bipartite graph with n left vertices and n right vertices. A (computationally unbounded) monotone adversary is allowed to examine the rows of A, and if the matrix was drawn from D_{planted} the adversary is allowed to delete any edges which did not belong to the planted bi-clique. More formally, for $i \notin R$, the adversary can set $x^i_j = 0$ for any $j \in [n]$; for $i \in R$, the adversary can set $x^i_j = 0$ for any $j \notin S$. Given (possibly modified) vectors $x^1, \ldots, x^n$, the goal is to distinguish if the vectors were originally drawn from D_{uniform} or D_{planted}.

To show hardness for Problem 6.1, we define a distinguishing problem which is similar to the Planted Bi-clique problem, but instead of planting the $\overrightarrow{1}$ pattern, allows planting an arbitrary pattern on some of the rows of the data.

Problem 6.2 (Pattern Planted Bi-Clique)**.** *Let $0 < k \le n$. The goal is to distinguish between the following joint distributions on n-bit vectors $x^1, \ldots, x^n$:*

1. *D_{uniform}: $\forall i \in [n]$, x^i is drawn from uniform distribution over $\{0,1\}^n$.*
2. *D_{planted}: $S \subseteq [n]$ is drawn uniformly at random from all subsets of $[n]$ of size k. Let $S = \{j_1, j_2, \ldots, j_k\}$. A vector v is drawn uniformly at random from $\{0,1\}^k$. R is drawn uniformly at random from all subsets of $[n]$ of size k.*

   $\forall i \notin R$, x^i is drawn from uniform distribution over $\{0,1\}^n$.
   $\forall i \in R$, $\forall m \in [k]$ $x^i_{j_m} = v_m$, and $\forall j \notin S$ x^i_j is a uniform $\{0,1\}$ bit.

We can show the following memory lower bound for Problem 6.2:

Theorem 6.3. *[Memory Lower Bound for Pattern Planted Bi-Clique] Let $0 < k \leq n$. Any p-pass streaming algorithm that solves Problem 6.2, when $x^1, x^2, \ldots, x^n$ arrive in a stream, requires at least $\Omega\left(\frac{n^2}{pk^3}\right)$ bits of memory.*

The proof of Theorem 6.3 (given in Appendix D) uses arguments similar to those in the proof of Theorem 5.2, by first decomposing the problem into a partitioned version, and showing hardness for the partitioned version. Crucially, because the planted pattern is a random pattern, this allows us to use smaller-sized partitions, and also simplifies the calculations involved in upper-bounding μ_1 by μ_0.

Using the hardness of Problem 6.2, we can derive the following memory lower bound for detecting planted bi-cliques in the monotone adversary/semi-random model.

Theorem 6.4 (Memory Lower Bound for Semi-random Planted Bi-Clique)**.** *Consider any $0 < k \leq n$. For any p-pass streaming algorithm that processes $x^1, x^2, \ldots, x^n$ arriving in a stream and only ever uses $o\left(\frac{n^2}{pk^3}\right)$ bits of memory, there exists some integer $k' \in \left[\frac{k}{3}, \frac{2k}{3}\right]$ and instance of Problem 6.1 with $k_1 = k, k_2 = k'$, for which the algorithm does not have advantage better than 0.9.*

We note that the theorem above holds for any algorithm that knows k, but does not know the *precise* value of $k' \in \left[\frac{k}{3}, \frac{2k}{3}\right]$.

Proof. Let $\mathcal{A}$ be any p-pass streaming algorithm that processes $x^1, \ldots, x^n$ arriving in a stream, uses only $o\left(\frac{n^2}{pk^3}\right)$ bits of memory, and satisfies that:

(1) If $x^1, \ldots, x^n \sim D_{\text{uniform}}$, then $\mathcal{A}$ outputs D_{uniform} with probability at least 0.9.

(2) For *every* $k' \in \left[\frac{k}{2} - 100\sqrt{k \log n}, \frac{k}{2} + 100\sqrt{k \log n}\right]$: if $x^1, \ldots, x^n \sim D_{\text{planted}}$ (as in Problem 6.2) *conditioned* on $|v| = k'$, then $\mathcal{A}$ outputs D_{planted} with probability at least 0.9.

But notice that when v is drawn uniformly at random from $\{0,1\}^k$, the probability that $|v| \in \left[\frac{k}{2} - 100\sqrt{k \log n}, \frac{k}{2} + 100\sqrt{k \log n}\right]$ is at least $1 - n^{-10}$. Together with (2) above, we conclude that: if $x^1, \ldots, x^n \sim D_{\text{planted}}$ (as in Problem 6.2), then $\mathcal{A}$ outputs D_{planted} with probability at least 0.89. But this contradicts the lower bound from Theorem 6.3. Thus, it must be the case that there exists $k' \in \left[\frac{k}{2} - 100\sqrt{k \log n}, \frac{k}{2} + 100\sqrt{k \log n}\right]$ such that $\mathcal{A}$ *does not* distinguish between D_{uniform} and D_{planted} conditioned on $|v| = k'$, with advantage 0.9.

For such a k', consider a problem instance of Problem 6.1 with $k_1 = k$, $k_2 = k'$, and a monotone adversary, who upon seeing $x^1, \ldots, x^n \sim D_{\text{planted}}$ (as in Problem 6.1), corrupts the non-planted columns (i.e., $[n] \setminus S$) in the planted rows R as follows: the adversary chooses a uniformly random subset $I \subseteq [n] \setminus S$ of size $k - k'$, and for every $i \in R$, the adversary sets $x^i_I = \mathbf{0}$. Then, observe that the final distribution of $x_1, \ldots, x_n$ after the corruption is exactly the distribution D_{planted} (from Problem 6.2), *conditioned* on $|v| = k'$. From our reasoning in the above paragraph, for this particular monotone adversary, $\mathcal{A}$ cannot distinguish between D_{uniform} and D_{planted} (as in Problem 6.1) with advantage 0.9. The theorem follows. □

6.1 Application: Maximum Bi-Clique

Theorem 6.4 implies a multi-pass streaming lower bound for approximating the size of the largest bi-clique in a graph in the Vertex Arrival Model (Definition 5.7).

Problem 6.5 (Maximum Bi-Clique)**.** *Consider an undirected graph G on n vertices (self-edges allowed). Let $1 < k' \leq n$. A k'-biclique in G corresponds to subsets $S, R \subseteq [n]$, $|S| = |R| = k'$, such that for every $u \in S, v \in R$, u and v are connected by an edge in G. The goal is to approximate the size of the largest biclique in G, i.e., $k = \max\{k' : \exists k'\text{-biclique in } G\}$. For $\alpha \geq 1$, an α-approximation to k is any number x such that $(1/\alpha)k \leq x \leq k$.*

Corollary 6.6 (Memory Lower Bound for Maximum Bi-Clique)**.** *Consider any $1 < \alpha \leq n$. Any p-pass streaming algorithm that approximates the size of the largest bi-clique in an undirected graph that is presented in the Vertex Arrival Model to a factor α requires at least $\tilde{\Omega}\left(\frac{n^2}{p\alpha^3}\right)$ bits of memory.*

Proof. Let $\mathcal{A}$ be a p-pass streaming algorithm that uses only $\tilde{o}\left(\frac{n^2}{p\alpha^3}\right)$ bits of memory and always approximates the size of the largest biclique in a graph presented in the worst-case, vertex-arrival model to a factor α. We will set $k = 40\alpha \log n$. Using $\mathcal{A}$, we will construct a p-pass streaming algorithm $\mathcal{A}'$ that processes $x^1, \ldots, x^n$ arriving in a stream, which solves every instance of Problem 6.1 for which $k_1 = k, k_2 \in \left[\frac{k}{3}, \frac{2k}{3}\right]$, while using only $\tilde{o}\left(\frac{n^2}{p\alpha^3}\right) = o\left(\frac{n^2}{pk^3}\right)$ bits of memory. This would contradict Theorem 6.4, and give us the claimed result.

The algorithm $\mathcal{A}'$ operates as follows. Given an input stream $x^1, x^2, \ldots, x^n$, $\mathcal{A}'$ interprets each x^i in the stream as a vertex v^i in an undirected graph G, and presents it to $\mathcal{A}$ in the vertex-arrival model. For each v^i, it will read off connectivity to $v^1, \ldots, v^i$ from $x^i_{[1:i]}$. That is, for $j \leq i$, there is an undirected edge between v^j and v^i iff $x^i_j = 1$. Note that $\mathcal{A}'$ can simulate this input space-efficiently for $\mathcal{A}$ (it only needs to keep track of a counter).

Now, suppose $x^1, x^2, \ldots, x^n$ are drawn from D_{uniform}. Then, observe that the graph G that $\mathcal{A}'$ presents to $\mathcal{A}$ is a random graph, where every (v^i, v^j) is connected by an edge with probability $1/2$. The size of the maximum biclique in such a graph is at most $3 \log n$ with probability $1 - o(1)$ [Tre17], and hence, the output of $\mathcal{A}$ will be at most $3 \log n$.

On the other hand, suppose $x^1, x^2, \ldots, x^n$ are drawn from D_{planted} in the semi-random model for $k_1 = k$, and any $k_2 \in \left[\frac{k}{3}, \frac{2k}{3}\right]$. Recall that S, R are uniformly random subsets of $[n]$ drawn without replacement of size k_2, k_1 respectively. By Hoeffding's bound (e.g., Proposition 1.2 in [BM15]), the size of $\{i \in S : i \geq n/2\}$ is at least $\frac{k_2}{3} \geq \frac{k}{10}$ with probability at least $1 - e^{-\Omega(k_2)} = 1 - e^{-\Omega(k)}$. Similarly, the size of $\{j \in R : j \leq n/2\}$ is at least $\frac{k_1}{3} = \frac{k}{3}$ with probability at least $1 - e^{-\Omega(k_1)} = 1 - e^{-\Omega(k)}$. Together with a union bound, we get that the size of both these sets is at least $\frac{k}{10}$ with probability at least $1 - e^{-\Omega(k)}$. But then note that conditioned on this event, in the undirected graph G that $\mathcal{A}'$ presents to $\mathcal{A}$, at least $\frac{k}{10}$ vertices in $v^{n/2}, \ldots, v^n$ are all connected to at least $\frac{k}{10}$ vertices in $v^1, \ldots, v^{n/2}$, meaning that the size of the largest biclique in G is at least $\frac{k}{10}$. Hence, the output of $\mathcal{A}$ will be at least $\frac{k}{10\alpha} = 4 \log n$.

Therefore, $\mathcal{A}'$ can distinguish between D_{uniform} and D_{planted} with constant advantage by checking if the output of $\mathcal{A}$ is at most $3 \log n$ or at least $4 \log n$. It does so using only $\tilde{o}\left(\frac{n^2}{p\alpha^3}\right) = o\left(\frac{n^2}{pk^3}\right)$ bits of memory, which gives us the desired contradiction.

□

7 Memory-Sample Tradeoffs for Distinguishing Sparse Gaussians

In this section, we prove our result for the sparse Gaussian distinguishing problem. We begin by stating the formal definition of the problem.

Problem 7.1. *Let $0 < \ell \leq d$ and $\alpha \in (0, 1]$. The goal is to distinguish between the following joint distributions on d-dimensional vectors $x^1, \ldots, x^n$:*

1. *D_{null}: $\forall i \in [n]$, x^i is drawn from the standard Gaussian distribution $N(0, I_d)$.*
2. *D_{planted}: A vector $v \in \mathbb{R}^d$ is drawn as follows: first, choose $S \subseteq [d], |S| = \ell$ uniformly at random. For every $i \in S$, set $v_i = \alpha$, and for every $i \in [d] \setminus S$, set $v_i = 0$. Then $\forall\, i \in [n]$,*

$$x^i \sim \begin{cases} N(v, I_d), & \text{w.p. } q, \\ N(0, I_d), & \text{w.p. } 1 - q. \end{cases} \tag{20}$$

A primary qualitative difference in the problem above compared to the distinguishing problems stated previously is that earlier, the planted distribution had exactly a *fixed* number k of vectors from amongst $x^1, \ldots, x^n$ that were drawn from the planted distribution; in contrast, in the problem above, each x^i independently has a probability q of being drawn from the planted distribution.

We show the following memory-sample tradeoff for Problem 7.1.

Theorem 7.2. *Let $\epsilon \in (0, 0.01)$ be a constant, d be sufficiently large, $\ell \leq d$, $n \leq d^{10}$ and $\alpha \in \left(\frac{1}{\ell\sqrt{\log d}}, 1\right]$. Any s-bit, p-pass algorithm (using public as well as private randomness) that solves Problem 7.1 for every $\ell' \in [2\ell/3, 4\ell/3]$ satisfies that $s \cdot n \geq \tilde{\Omega}\left(\frac{d^{1-\epsilon}}{p(\alpha\ell)^2 q^2}\right)$.*

Again, the theorem above holds for any algorithm that knows ℓ, but does not know the precise value of $\ell' \in [2\ell/3, 4\ell/3]$.

Proof. As in the other proofs, we consider a partition version of the problem, where the planted coordinates in the vector v are confined to being within a partition. Furthermore, while Problem 7.1 has the property that nq vectors, *in expectation*, have a plant corresponding to a vector v of *fixed sparsity* ℓ, this property is flipped in the partition version; namely, it will be the case that a *fixed number* $k = nq$ of the vectors have a plant corresponding to a vector v of *expected sparsity* ℓ.

Concretely, consider the following distribution D over vectors in $\mathbb{R}^t$, for $t \geq (\alpha\ell)^2 d^\epsilon \log^2(200nd)$. Independently, for every co-ordinate v_j, $j \in [t]$,

$$v_j = \begin{cases} \alpha, & \text{w.p. } \ell/t, \\ 0, & \text{w.p. } 1 - \ell/t. \end{cases} \tag{21}$$

We can see that originally in Problem 7.1, there were *exactly* ℓ coordinates where v was non-zero, whereas $v \sim D$ above has ℓ coordinates that are non-zero *in expectation*, and all these coordinates are contained within the same partition (of size t). Consider now the following problem, for which we will show hardness.

Problem 7.3. *Let $t \geq \max\{(\alpha\ell)^2 d^\epsilon \log^2(200nd), 2\ell\}$ and suppose that t divides d. Let $\mathcal{T} = \{T_r\}_{r \in [d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. The goal is to distinguish between the following joint distributions on d-bit vectors $x^1, \ldots, x^n$:*

1. D_0 *(no instance):* $\forall i \in [n]$ *and* $\forall r \in [d/t]$, $x^i_{T_r}$ *is drawn from* $N(0, I_t)$.

2. $D_1^{\mathcal{T}}$ *(yes instance): Draw* r *uniformly from* $[d/t]$. $\forall i \in [n]$ *and* $\forall r' \neq r$, $x^i_{T_{r'}}$ *is drawn from* $N(0, I_t)$.
   Draw $v \sim D$. *Draw a uniformly random subset* $R \subseteq [n]$ *of size* k.
   $\forall i \notin R$, $x^i_{T_r}$ *is drawn from* $N(0, I_t)$, *whereas,* $\forall i \in R$, $x^i_{T_r}$ *is drawn from* $N(v, I_t)$.

In the following lemma, we show that proving hardness for Problem 7.3 is enough to prove the theorem. The proof of this lemma is a sequence of reductions that uses arguments similar to those used earlier in the paper, and is deferred to Appendix E.

Lemma 7.4. *Let* $\epsilon \in (0, 0.01)$ *be a constant,* d *be sufficiently large,* $\ell \leq d$, $n \leq d^{10}$ *and* $\alpha \in \left(\frac{1}{\ell\sqrt{\log d}}, 1\right]$. *Let* $\mathcal{A}$ *be a* p*-pass streaming algorithm that uses* s *bits of memory and* $n/400$ *samples, and solves Problem 7.1 with probability 0.99 for every value of* $\ell' \in [2\ell/3, 4\ell/3]$. *Then, there exists a* p*-pass streaming algorithm* $\mathcal{A}'$ *that uses* $s + \tilde{O}(1)$ *bits of memory and* n *samples, and solves Problem 7.3 for* $k = nq$ *with probability 0.97*

We note that the assumption of t dividing d in Problem 7.3 is for convenience; we can handle the technicality of t not dividing d similarly to how we did in the proof of Theorem 5.2; this is fleshed out in more detail in the proof of the lemma.

Using Lemma 7.4, it suffices to show hardness for Problem 7.3. For this, however, we will need to define a truncation, on both the x^is and v. There are some steps where we will not be able to get a good bound for all vectors v. So, we define a set $V_{good} = \{v : \|v\|_0 \leq 100\ell\}$ and let D_{good} be the distribution D that is restricted to vectors in the set V_{good}. We will also need to truncate the distribution over x^is to get our bound. We define the set

$$T = \left\{ x \in \mathbb{R}^t : \sum_{j=1}^{t} e^{\alpha x_j} \leq t e^{\alpha^2/2} + (C_1\alpha)\sqrt{t} d^{\epsilon/2} \log(200nd) \right\} \tag{22}$$

for a constant C_1 to be later determined. Let P^0_{trunc} be the restriction of the (t-dimensional) Gaussian distributions $N(0, I_t)$ to this set T. For a vector $v \in \mathbb{R}^t$, we let $P^{1,v}_{\text{trunc}}$ denote the restriction of the Gaussian distribution $N(v, I_t)$ to the set T.

We now further define a *truncated* version of Problem 7.3.

Problem 7.5. *Let* $t \geq \max\{(\alpha\ell)^2 d^\epsilon \log^2(200nd), 2\ell\}$ *and suppose that* t *divides* d. *Let* $\mathcal{T} = \{T_r\}_{r \in [d/t]}$ *be a partition of* $[d]$, *where* $\forall r, |T_r| = t$. *The goal is to distinguish between the following joint distributions on* d*-bit vectors* $x^1, \ldots, x^n$*:*

1. D_0*:* $\forall i \in [n]$ *and* $\forall r \in [d/t]$, $x^i_{T_r}$ *is drawn from* P^0_{trunc}.

2. $D_1^{\mathcal{T}}$*: Draw* r *uniformly from* $[d/t]$. $\forall i \in [n]$ *and* $\forall r' \neq r$, $x^i_{T_{r'}}$ *is drawn from* P^0_{trunc}.
   Draw $v \sim D_{good}$. *Draw a uniformly random subset* $R \subseteq [n]$ *of size* k.
   $\forall i \notin R$, $x^i_{T_r}$ *is drawn from* P^0_{trunc}, *whereas,* $\forall i \in R$, $x^i_{T_r}$ *is drawn from* $P^{1,v}_{\text{trunc}}$.

The following lemma, proved in Appendix E, shows that the truncated distributions are close to the original ones.

Lemma 7.6. *Let $v \in V_{good}$ be arbitrary. The distributions $N(0, I_t)$ and $N(v, I_t)$ are close (in TV distance) to their respective truncations P^0_{trunc} and $P^{1,v}_{\text{trunc}}$:*

$$\left\|P^0_{\text{trunc}} - N(0, I_t)\right\|_{TV} \leq 0.01/(nd/t),$$
$$\left\|P^{1,v}_{\text{trunc}} - N(v, I_t)\right\|_{TV} \leq 0.01/(nd/t).$$

Also, $\Pr_{v\sim D}[v \in V_{good}] \geq 0.99$.

By Lemma 7.6 and the triangle inequality applied to the partitions of t coordinates, the TV distance between D_0 defined in Problem 7.3 and in Problem 7.5 is at most 0.01. Similarly, the TV distance between $D_1^{\mathcal{T}}$ in these problems, taking additionally into account that $\Pr_{v\sim D}[v \notin V_{good}] \leq 0.01$, is at most 0.02; note that in both cases, the TV distance for a fixed setting of r, R, v is at most 0.02, and the random processes by which r, R, v are selected in both problems are identical.

Since the respective distributions in Problem 7.3 and Problem 7.5 are close, it follows that if some algorithm can solve Problem 7.3 with advantage 0.99, then the algorithm can also solve Problem 7.5 with advantage 0.97. Therefore, we will now show a lower bound for Problem 7.5, which as we can observe, conveniently fits the template of Problem 4.1. With a view to invoke Theorem 4.2, the next claim, whose proof is a calculation and is also given in Appendix E, bounds the ratio $\mathbb{E}_{v\sim D_{good}}[P^{1,v}_{\text{trunc}}]/P^0_{\text{trunc}}$.

Claim 7.7. *Let μ_0 and μ_1^v be the probability density functions of P^0_{trunc} and $P^{1,v}_{\text{trunc}}$ respectively. Then, there exists a positive constant C such that*

$$\mathbb{E}_{v\sim D_{good}}[\mu_1^v] \leq C\mu_0.$$

We have thus shown that Problem 7.5 fits the generic description of Problem 4.1, and also satisfies the requirement of Theorem 4.2. The statement of the theorem then implies that any s-bit, p-pass streaming algorithm which solves the problem satisfies $s = \Omega\left(\frac{nd}{pk^2t}\right)$. Taking $k = nq$ and $t = (\alpha\ell)^2 d^\epsilon \log^2(200nd)$ gives us that $s \cdot n \geq \tilde{\Omega}\left(\frac{d^{1-\epsilon}}{p(\alpha\ell)^2q^2}\right)$. Invoking Lemma 7.6 and Lemma 7.4 completes the proof of the theorem.

□

We now discuss how our memory-sample tradeoff established in Theorem 7.2 relates to those of algorithms that solve Problem 7.1. First, consider a statistical test that computes the sum of all the coordinate values across the n samples it receives (i.e., the sum $\sum_{i=1}^n \sum_{j=1}^d x_j^i$). When samples are drawn from D_{null}, the sum is a Gaussian with mean 0 and variance nd. On the other hand, when samples are drawn from D_{planted}, the sum is a Gaussian with mean $nq\ell\alpha$ and variance nd. Now suppose that the test declares D_{null} if and only if the sum is at most $nq\ell\alpha/2$. By a Gaussian tail bound, the test's failure probability is at most δ if $n = O(d\log(1/\delta)/(q\ell\alpha)^2)$. For a constant success probability, this test would require $O(d/(\alpha\ell q)^2)$ samples. Note also that the test can be computed with $O(\log d)$ bits of precision.[7] Therefore, our bound given in Theorem 7.2 is nearly optimal for algorithms in the $O(\log d)$ memory regime.

[7] The accumulation error for the sum computation is at most $nd \cdot 2^{-\rho}$, where ρ is the precision of bits for each coordinate's floating points. Since we require the accumulation error to not exceed each distribution's standard deviation of $\sqrt{nd}$, we can take $\rho = O(\log n + \log d)$ bits. Furthermore, Chebyshev's inequality implies that the computed sum is at most $2nd$ with high probability, so we can also take $O(\log n + \log d)$ bits for the integral component of the sum. Since we assume $n \leq d^{10}$, we therefore need $O(\log d)$ bits altogether.

We remark that Problem 7.1 can also be solved with improved sample complexity via the following procedure. The algorithm fixes a set R of randomly chosen coordinates and for each sample x^j it receives, it stores all the entries at those coordinates (that is, the value x^j_R). For each subset $S_1 \subseteq [n]$ of the received samples with $|S_1| = s_1$ and each subset of the coordinates $S_2 \subseteq R$ with $|S_2| = s_2$, the algorithm computes the statistic $Y_{S_1,S_2} = \sum_{j\in S_1} \sum_{i\in S_2} x^j_i$. The algorithm declares D_{null} if no statistic Y_{S_1,S_2} exceeds some fixed threshold; otherwise, it declares D_{planted}. In the following claim, we show that there is a regime in which this test is able to distinguish between the distributions D_{null} and D_{planted} using $\tilde{O}(1/(q\alpha^2))$ samples and $\tilde{O}\left(\frac{d}{\ell\alpha^2 q}\right)$ memory.

Claim 7.8. *Fix a constant $\delta \in (0,1)$ and let $C_{\delta,\alpha} = \left(\frac{8+4\log(4/\delta)}{\alpha^2}\right)$. For all n, d sufficiently large that satisfy $nq \geq 2C_{\delta,\alpha}\log(nd)$, the following holds. If $|R| = 2C_{\delta,\alpha}(d/\ell)\log(nd/\delta)\log(nd)$, $\ell \geq s_1 = s_2 = C_{\delta,\alpha}\log(nd)$ and $\tau = \sqrt{2s_1s_2\log\left(2\binom{n}{s_1}\binom{|R|}{s_2}/\delta\right)}$, then*

$$\max\left\{\Pr_{D_{\text{null}}}\left[\max_{\substack{S_1\subseteq[n],|S_1|=s_1\\ S_2\subseteq R,|S_2|=s_2}} \sum_{j\in S_1}\sum_{i\in S_2} x^j_i \geq \tau\right], \Pr_{D_{\text{planted}}}\left[\max_{\substack{S_1\subseteq[n],|S_1|=s_1\\ S_2\subseteq R,|S_2|=s_2}} \sum_{j\in S_1}\sum_{i\in S_2} x^j_i \leq \tau\right],\right\} \leq \delta.$$

8 Memory-Sample Tradeoffs for Sparse PCA Detection

In this section, we prove our result for the sparse PCA detection problem. We begin by stating the formal definition of the problem.

Problem 8.1. *Let $\ell \leq d$ be some integers and let $\alpha > 0$ be some parameter. The goal is to distinguish between the following joint distributions on d-dimensional vectors $x^1, \ldots, x^n$:*

1. *D_{null}: $\forall i \in [n]$, x^i is drawn from $N(0, I_d)$.*
2. *D_{planted}: Draw a uniformly random subset of ℓ indices, $S \subseteq [d]$ and let $v = \frac{1}{\sqrt{\ell}}1_S$.*

   $\forall i \in [n]$, x^i is drawn from $N(0, \Sigma_S)$, where $\Sigma_S = I_d + \alpha vv^\top$.

We show the following memory-sample tradeoff for Problem 8.1.

Theorem 8.2. *Let $\epsilon \in (0, 0.01)$ be a constant, $\alpha \in \left(0, \frac{\epsilon}{22}\right)$ be a constant, d be sufficiently large, $\ell \leq d$, and $n \leq d^{10}$. Then, any s-bit, p-pass algorithm (using public as well as private randomness) that solves Problem 8.1 satisfies $s \cdot n \geq \tilde{\Omega}\left(\frac{d^{1-\epsilon}}{p\ell}\right)$.*

Proof. Our approach is to show a lower bound for the following simpler distinguishing problem where the planted distribution is a more structured mixture of ℓ-sizes subsets of indices. A lower bound for this problem implies a lower bound for the more general Problem 8.1.

Problem 8.3. *Let $\ell \leq d$ be some integers and let $\alpha > 0$ be some parameter. The goal is to distinguish between the following joint distributions on d-dimensional vectors $x^1, \ldots, x^n$:*

1. *D_{null}: $\forall i \in [n]$, x^i is drawn from $N(0, I_d)$.*
2. *D_{planted}: Draw S uniformly from $\{[1,\ell], [\ell+1, 2\ell], \ldots, [d-\ell+1, d]\}$ and let $v = \frac{1}{\sqrt{\ell}}1_S$.*

   $\forall i \in [n]$, x^i is drawn from $N(0, \Sigma_S)$, where $\Sigma_S = I_d + \alpha vv^\top$.

As in the other proofs, we consider a partition version of the problem, where the planted coordinates in the vector v are confined to being within a partition. Consider now the following problem for which we will show hardness.

Problem 8.4. *Let $t \geq \ell d^\epsilon \log(400nd)$ and suppose that ℓ divides t and that t divides d. Let $\mathcal{T} = \{T_r\}_{r\in[d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. The goal is to distinguish between the following joint distributions on d-bit vectors $x^1, \ldots, x^n$:*

1. *D_0 (no instance): $\forall i \in [n]$ and $\forall r \in [d/t]$, $x^i_{T_r}$ is drawn from $N(0, I_t)$.*
2. *$D_1^{\mathcal{T}}$ (yes instance): Draw r uniformly from $[d/t]$. $\forall i \in [n]$ and $\forall r' \neq r$, $x^i_{T_{r'}}$ is drawn from $N(0, I_t)$.*
   Draw S uniformly from $\mathcal{S} = \{[1, \ell], [\ell + 1, 2\ell], \ldots, [t - \ell + 1, t]\}$ and let $v = \frac{1}{\sqrt{\ell}} 1_S$.
   $\forall i \in [n]$, $x^i_{T_r}$ is drawn from $N(0, \Sigma_S)$, where $\Sigma_S = I_t + \alpha vv^\intercal$.

We will need to truncate the distribution over x^is to get our bound. We define the set

$$T = \left\{ x \in \mathbb{R}^t : \sum_{R\in\mathcal{S}} \exp\left(\frac{\alpha}{2(\alpha+1)} \cdot \frac{1}{\ell} (x^\intercal 1_R)^2 \right) \leq (t/\ell)(1-\alpha)^{-1/2} + \delta \right\}, \tag{23}$$

where $\delta = Cd^{\epsilon/2}\sqrt{(t/\ell)\log(400nd)}$. Let P^0_{trunc} be the restriction of the Gaussian distribution $N(0, I_t)$ to this set T. For each set $S \in \mathcal{S}$, we let $P^{1,S}_{\text{trunc}}$ denote the restriction of the Gaussian distribution $N(0, \Sigma_S)$ to the set T, where $\Sigma_S = I_t + \alpha vv^\intercal$ and $v = \frac{1}{\sqrt{\ell}} 1_S$.

We now further define a *truncated* version of Problem 8.4.

Problem 8.5. *Let $t \geq \ell d^\epsilon \log(400nd)$ and suppose that ℓ divides t and that t divides d. Let $\mathcal{T} = \{T_r\}_{r\in[d/t]}$ be a partition of $[d]$, where $\forall r, |T_r| = t$. The goal is to distinguish between the following joint distributions on d-bit vectors $x^1, \ldots, x^n$:*

1. *D_0: $\forall i \in [n]$ and $\forall r \in [d/t]$, $x^i_{T_r}$ is drawn from P^0_{trunc}.*
2. *$D_1^{\mathcal{T}}$: Draw r uniformly from $[d/t]$. $\forall i \in [n]$ and $\forall r' \neq r$, $x^i_{T_{r'}}$ is drawn from P^0_{trunc}.*
   Draw S uniformly from $\mathcal{S} = \{[1, \ell], [\ell + 1, 2\ell], \ldots, [t - \ell + 1, t]\}$.
   $\forall i \in [n]$, $x^i_{T_r}$ is drawn from $P^{1,S}_{\text{trunc}}$.

The following lemma, proved in Appendix F, shows that the truncated distributions are close to the original ones.

Lemma 8.6. *For any set $S \in \mathcal{S}$, the distributions $N(0, I_t)$ and $N(0, \Sigma_S)$ are close (in TV distance) to their respective truncations P^0_{trunc} and $P^{1,S}_{\text{trunc}}$:*

$$\left\| P^0_{\text{trunc}} - N(0, I_t) \right\|_{TV} \leq 0.01/(nd/t),$$
$$\left\| P^{1,S}_{\text{trunc}} - N(0, \Sigma_S) \right\|_{TV} \leq 0.01/(nd/t).$$

By Lemma 8.6 and the triangle inequality applied to the partitions of t coordinates, the TV distance between D_0 defined in Problem 8.4 and in Problem 8.5 is at most 0.01. Similarly, the TV distance between $D_1^{\mathcal{T}}$ in these problems is at most 0.01.

Since the respective distributions in Problem 8.4 and Problem 8.5 are close, it follows that if some algorithm can solve Problem 8.4 with advantage 0.99, then the algorithm can also solve Problem 8.5 with advantage 0.98. Therefore, we will now show a lower bound for Problem 8.5, which as we can observe, conveniently fits the template of Problem 4.1. With a view to invoke Theorem 4.2, the next claim, whose proof is a calculation and is also given in Appendix F, bounds the ratio $\mathbb{E}_S[P^{1,S}_{\text{trunc}}]/P^0_{\text{trunc}}$.

Claim 8.7. *Let μ_0 and μ_1^S be the probability density functions of P^0_{trunc} and $P^{1,S}_{\text{trunc}}$ respectively. Then, there exists a positive constant C such that*

$$\mathbb{E}_{S\sim\mathcal{S}}[\mu_1^S] \le C\mu_0.$$

We have thus shown that Problem 8.5 fits the generic description of Problem 4.1, and also satisfies the requirement of Theorem 4.2. The statement of Theorem 4.2 then implies that any s-bit, p-pass streaming algorithm which solves the problem satisfies $s = \Omega\left(\frac{nd}{pk^2t}\right)$. Taking $k = n$ and $t = \ell d^\epsilon \log(400nd)$ gives us that $s \cdot n \ge \tilde{\Omega}\left(\frac{d^{1-\epsilon}}{p\ell}\right)$. Invoking Lemma 8.6 completes the proof of the theorem. □

We now discuss how our memory-sample tradeoff established in Theorem 8.2 relates to those of algorithms that solve Problem 8.3. Consider the statistical test that squares the sum of coordinates within each block and computes the cumulative sum across samples (i.e, the sum $\sum_{j=1}^n \sum_{R\in\mathcal{S}}(\sum_{i\in R} x_i^j)^2$). If the sum exceeds the threshold $\tau = nd + n\alpha\ell/2$, then the test declares D_{planted}; otherwise it declares D_{null}. In the following claim, whose proof is a calculation given in Appendix F, we show that the test is able to distinguish between D_{null} and D_{planted} with a constant failure probability using $O(d/\ell)$ samples.

Claim 8.8. *Fix a constant $\delta \in (0,1)$ and suppose that $n \ge \log\left(\frac{2}{\delta}\right)\left[\frac{4C_1^2(1+\alpha)^2}{c\alpha^2}\cdot\frac{d}{\ell}\right]$. Then,*

$$\max\left\{\Pr_{D_{\text{null}}}\left[\sum_{j=1}^n\sum_{R\in\mathcal{S}}\left(\sum_{i\in R}x_i^j\right)^2 \ge \tau\right], \Pr_{D_{\text{planted}}}\left[\sum_{j=1}^n\sum_{R\in\mathcal{S}}\left(\sum_{i\in R}x_i^j\right)^2 \le \tau\right]\right\} \le \delta$$

It is straightforward to verify that the statistical test can be computed with $O(\log d)$ bits of precision (and the justification mirrors that given for the Gaussian mean distinguishing in Section 7). Therefore, our bound given in Theorem 8.2 is nearly optimal for algorithms in the $O(\log d)$ memory regime that solve Problem 8.3.

Acknowledgments

VS was supported by NSF CAREER Award CCF-2239265, an Amazon Research Award, a Google Research Scholar Award and a Okawa Foundation Award. The work was done in part while VS was visiting the Simons Institute for the Theory of Computing. CP was supported by Gregory

Valiant's and Moses Charikar's Simons Investigator Awards, and a Google PhD Fellowship. JH is supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness and the Simons Foundation investigators award 689988. CP and JH would like to thank Gregory Valiant and Annie Marsden for many insightful discussions on the planted clique problem.

A Proofs from Section 3

We first restate and prove Lemma 3.1.

Lemma 3.1 (Claim 3.4 in [BGL+24]). *Consider a stream $X^1, \dots, X^n$ from a product distribution, and let M be a p-pass streaming protocol that uses public randomness P and private randomness $R^{\mathsf{M}} = \{R^{\mathsf{M}}_{l,i}\}_{l\in[p],i\in[n]}$, where the private randomness at every step is mutually independent, as well as independent of the public randomness. Then, for any $i, j \in [n]$, $i < j$, and any $l \in [p]$, it holds that:*

$$I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{<l,i-1}, \mathsf{M}_{<l,j-1}, P) = 0, \tag{5}$$

$$I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{<l,j-1}, P) = 0. \tag{6}$$

Proof. We will prove this lemma by induction. For the base case, consider (5) for $l = 1$. In this case, the memory states conditioned on are simply the initial memory state M_0. So, we have that

$$I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_0, P) = 0,$$

precisely because the inputs, the public randomness, the private randomness at every time step, and the initial memory state, are all independent of each other. Now, assume as the induction hypothesis that (5) holds for some l. We will first show that, if $l \le p$, then (6) holds. For this, observe that

$$\begin{aligned}
&I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{<l,j-1}, P) \\
&\le I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])}, \mathsf{M}_{l,i-1} \mid \mathsf{M}_{<l,i-1}, \mathsf{M}_{<l,j-1}, P) \\
&\qquad \text{(chain rule and non-negativity of mutual information)} \\
&= I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{<l,i-1}, \mathsf{M}_{<l,j-1}, P) \\
&\quad + I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; \mathsf{M}_{l,i-1} \mid \mathsf{M}_{<l,i-1}, \mathsf{M}_{<l,j-1}, X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])}, P) \\
&\qquad \text{(chain rule)} \\
&= I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; \mathsf{M}_{l,i-1} \mid \mathsf{M}_{<l,i-1}, \mathsf{M}_{<l,j-1}, X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])}, P) \\
&\qquad \text{(induction hypothesis)} \\
&= 0. \qquad (\mathsf{M}_{l,i-1} \text{ is determined by } \mathsf{M}_{<l,j-1}, X^{[j,n]}, R_{([p],[j,n])}, X^{[1,i-1]}, R_{([p],[1,i-1])}, P)
\end{aligned}$$

Next, we will show that if $l \le p - 1$, then (5) holds for $l + 1$. Namely, observe that

$$\begin{aligned}
&I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{\le l,j-1}, P) \\
&\le I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])}, \mathsf{M}_{l,j-1} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{<l,j-1}, P) \\
&\qquad \text{(chain rule and non-negativity of mutual information)} \\
&= I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{<l,j-1}, P) \\
&\quad + I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; \mathsf{M}_{l,j-1} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{<l,j-1}, X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])}, P) \\
&\qquad \text{(chain rule)}
\end{aligned}$$

$$= I(X^{[i,j-1]}, R_{([p],[i,j-1])} \; ; \; \mathsf{M}_{l,j-1} \mid \mathsf{M}_{\le l,i-1}, \mathsf{M}_{<l,j-1}, X^{[1,i-1]}, R_{([p],[1,i-1])}, X^{[j,n]}, R_{([p],[j,n])}, P)$$

(since we showed above that (6) holds for l)

$$= 0.$$ ($\mathsf{M}_{l,j-1}$ is determined by $\mathsf{M}_{\le l,i-1}, X^{[i,j-1]}, R_{([p],[i,j-1])}, P$)

This completes the proof by induction. □

We now restate and prove Lemma 3.2.

Lemma 3.2 (Lemma 1.1, [BGL+24])**.** *Let $(X^1, X^2, \cdots, X^n)$ be drawn from a product distribution μ. Then, for any p-pass streaming algorithm M that uses public as well as private randomness, has memory size s and runs on input stream $X^1, \cdots, X^n$, it holds that:*

$$MIC(\mathsf{M}, \mu) \le 2p \cdot s \cdot n.$$

Proof. The proof mimics the proof of Lemma 1.1 in [BGL+24], albeit with the addition of public randomness P. We will prove that, for every pass $\ell \in [p]$, it holds that

$$\sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le \ell, j-1)}, \mathsf{M}_{(\le \ell-1,i)}, P\right) \le s \cdot n \tag{24}$$

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le \ell-1, j-1)}, \mathsf{M}_{(\le \ell-1,i)}, P\right) \le s \cdot n, \tag{25}$$

which implies the lemma.

We start by establishing (24).

$$\sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le \ell, j-1)}, \mathsf{M}_{(\le \ell-1,i)}, P\right)$$

$$\le \sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j, X^{[1,j-1]}, \mathsf{M}_{\le \ell, \le j-2} \mid \mathsf{M}_{(\le \ell, j-1)}, \mathsf{M}_{(\le \ell-1,i)}, P\right)$$

(chain rule and non-negativity of mutual information)

$$= \sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{[1,j-1]}, \mathsf{M}_{\le \ell, \le j-2} \mid \mathsf{M}_{(\le \ell, j-1)}, \mathsf{M}_{(\le \ell-1,i)}, P\right)$$

$$+ \sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le \ell, \le j-1)}, \mathsf{M}_{(\le \ell-1,i)}, X^{[1,j-1]}, P\right) \quad \text{(chain rule)}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j \mid \mathsf{M}_{(\le \ell, \le j-1)}, \mathsf{M}_{(\le \ell-1,i)}, X^{[1,j-1]}, P\right) \quad (\star)$$

$$\le \sum_{i=1}^{n} \sum_{j=1}^{i} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^j, \mathsf{M}_{(\le \ell, j)} \mid \mathsf{M}_{(\le \ell, \le j-1)}, \mathsf{M}_{(\le \ell-1,i)}, X^{[1,j-1]}, P\right)$$

(chain rule and non-negativity of mutual information)

$$= \sum_{i=1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{[1,i]}, \mathsf{M}_{(\le \ell, \le i)} \mid \mathsf{M}_{(\le \ell-1,i)}, P\right) \quad \text{(chain rule)}$$

$$\leq s \cdot n, \qquad \text{(Claim A.1)}$$

which establishes (24). In the above, ($\star$) follows due to the fact that every summand in the first double-summation in the previous step is 0, namely,

$$\mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{[1,j-1]}, \mathsf{M}_{\leq \ell, \leq j-2} \mid \mathsf{M}_{(\leq \ell, j-1)}, \mathsf{M}_{(\leq \ell-1,i)}, P\right) = 0. \tag{26}$$

To see this, recall that (6) in Lemma 3.1 implies that

$$I(X^{[j,i]}, R_{([p],[j,i])} \; ; \; X^{[1,j-1]}, R_{([p],[1,j-1])}, X^{[i+1,n]}, R_{([p],[i+1,n])} \mid \mathsf{M}_{\leq \ell, j-1}, \mathsf{M}_{\leq \ell-1,i}, P) = 0.$$

Now, observe that $\mathsf{M}_{(\ell,i)}$ is a deterministic function of $X^{[j,i]}, R_{[p],[j,i]}$, conditioned on $\mathsf{M}_{(\leq \ell, j-1)}, P$ Similarly, $X^{[1,j-1]}, \mathsf{M}_{\leq \ell, \leq j-2}$ is a deterministic function of $X^{[1,j-1]}, R_{([p],[1,j-1])}, X^{[i+1,n]}, R_{([p],[i+1,n])}$, conditioned on $\mathsf{M}_{(\leq \ell-1,i)}, P$. This implies (26).

Similarly, for (25), we have that

$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{j} \mid \mathsf{M}_{(\leq \ell-1,j-1)}, \mathsf{M}_{(\leq \ell-1,i)}, P\right) \\
&\leq \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{j}, X^{[i+1,j-1]}, \mathsf{M}_{(\leq \ell-1,[i+1,j-2])} \mid \mathsf{M}_{(\leq \ell-1,j-1)}, \mathsf{M}_{(\leq \ell-1,i)}, P\right) \\
&\qquad\qquad \text{(chain rule and non-negativity of mutual information)} \\
&= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{[i+1,j-1]}, \mathsf{M}_{(\leq \ell-1,[i+1,j-2])} \mid \mathsf{M}_{(\leq \ell-1,j-1)}, \mathsf{M}_{(\leq \ell-1,i)}, P\right) \\
&\qquad + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{j} \mid \mathsf{M}_{(\leq \ell-1,[i+1,j-1])}, \mathsf{M}_{(\leq \ell-1,i)}, X^{[i+1,j-1]}, P\right) \qquad \text{(chain rule)} \\
&= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{j} \mid \mathsf{M}_{(\leq \ell-1,[i+1,j-1])}, \mathsf{M}_{(\leq \ell-1,i)}, X^{[i+1,j-1]}, P\right) \qquad (\star) \\
&\leq \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{j}, \mathsf{M}_{(\leq \ell-1,j)} \mid \mathsf{M}_{(\leq \ell-1,[i+1,j-1])}, \mathsf{M}_{(\leq \ell-1,i)}, X^{[i+1,j-1]}, P\right) \\
&\qquad\qquad \text{(chain rule and non-negativity of mutual information)} \\
&= \sum_{i=1}^{n} \mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{[i+1,n]}, \mathsf{M}_{(\leq \ell-1,[i+1,n])} \mid \mathsf{M}_{(\leq \ell-1,i)}, P\right) \qquad \text{(chain rule)} \\
&\leq s \cdot n, \qquad \text{(Claim A.1)}
\end{aligned}$$

which establishes (25). Again, ($\star$) above follows because every summand in the first double-summation in the previous step is 0, namely,

$$\mathrm{I}\left(\mathsf{M}_{(\ell,i)}; X^{[i+1,j-1]}, \mathsf{M}_{(\leq \ell-1,[i+1,j-2])} \mid \mathsf{M}_{(\leq \ell-1,j-1)}, \mathsf{M}_{(\leq \ell-1,i)}, P\right) = 0. \tag{27}$$

To see this, recall that (5) in Lemma 3.1 implies that

$$I(X^{[i+1,j-1]}, R_{([p],[i+1,j-1])} \; ; \; X^{[1,i]}, R_{([p],[1,i])}, X^{[j,n]}, R_{([p],[j,n])} \mid \mathsf{M}_{\leq \ell-1,i}, \mathsf{M}_{\leq \ell-1,j-1}, P) = 0,$$

Now, observe that $X^{[i+1,j-1]}, \mathsf{M}_{(\leq \ell-1,[i+1,j-2])}$ is a deterministic function of $X^{[i+1,j-1]}, R_{([p],[i+1,j-1])}$, conditioned on $\mathsf{M}_{\leq \ell-1,i}, P$. Similarly, $\mathsf{M}_{(\ell,i)}$ is a deterministic function of $X^{[1,i]}, R_{([p],[1,i])}, X^{[j,n]}, R_{([p],[j,n])}$, conditioned on $\mathsf{M}_{(\leq \ell-1,j-1)}, P$. This implies (27). □

Claim A.1. *If A is a discrete random variable with probability mass function p_A, and B is an arbitrary random variable, then $I(A;B) \leq H(A)$, where $H(A) = -\mathbb{E}_A[\log(p(A))]$ is the entropy of A. In particular, if A has finite support of size N, then $I(A;B) \leq \log(|N|)$.*

Proof. By definition,

$$\begin{aligned}
I(A;B) &= \mathbb{E}_B[D_{KL}(p_{A|B} \,||\, p_A)] = \mathbb{E}_B\left[\sum_a p_{A|B}(a)\log\left(\frac{p_{A|B}(a)}{p_A(a)}\right)\right] \\
&= \mathbb{E}_B\left[\sum_a p_{A|B}(a)\log(p_{A|B}(a))\right] - \mathbb{E}_B\left[\sum_a p_{A|B}(a)\log(p_A(a))\right] \\
&= \mathbb{E}_B\left[\sum_a p_{A|B}(a)\log(p_{A|B}(a))\right] - \sum_a \log(p_A(a))\mathbb{E}_B\left[p_{A|B}(a)\right] \\
&= \mathbb{E}_B\left[\sum_a p_{A|B}(a)\log(p_{A|B}(a))\right] - \sum_a \log(p_A(a))p_A(a) \\
&= \mathbb{E}_B\left[\sum_a p_{A|B}(a)\log(p_{A|B}(a))\right] + H(A).
\end{aligned}$$

It remains to argue that the first summand above is non-positive. Note that $\sum_a p_{A|B}(a)\log(p_{A|B}(a))$ is the expectation of the concave function $\log(p_{A|B}(\cdot))$ where the argument is drawn from the conditional distribution $A|B$. Applying Jensen's inequality, we get that

$$\sum_a p_{A|B}(a)\log(p_{A|B}(a)) \leq \log\left(\sum_a p_{A|B}(a)^2\right) \leq \log(1) = 0.$$

This concludes the proof. □

B Proofs from Section 4

We restate and prove Claim 4.7

Claim 4.7. *For every good R (where q_R as defined above is at least 0.9), $MIC^R \geq \Omega\left(\frac{d}{c \cdot t}\right)$.*

Proof of Claim 4.7. We begin by reiterating that M is a $(p+1)$-pass algorithm that solves the distinguishing problem $\mathsf{DP}(\mu_0, \{\mu_\theta\}_{\theta\in\Omega}, P, \mathcal{T}, k, n)$ with large enough constant probability, say $1-\delta$ (and recalling that, we added another pass that doesn't do any operations to M).

We will convert the streaming algorithm into a communication protocol and use calculations similar those used in the proof of Claim 5.4 in [BGL+24].

Communication Protocol Π for k-party General Planted Problem using M. Given input $z^1, z^2, \ldots z^k \in \mathcal{X}^d$ to the k-parties respectively, all the parties together prepare an input to the multi-pass streaming algorithm M with z embedded at rows in R. For all $a \in [k-1]$, the a-th player, using private randomness, samples $\{x^i\}_{i_a < i < i_{a+1}}$ independently according to D_0 (under D_0, each row is $i.i.d.$) and sets $x^{i_a} = z^a$. The last player sets $x^{i_k} = z^k$, and samples $\{x^i\}_{i_k < i \le n}$ as well as $\{x^i\}_{1 \le i < i_1}$.

All players then simulate M one pass at a time, using their part of the input stream, public randomness P, and any additional private randomness that M requires. Knowing $\{x^i\}_{1 \le i < i_1}$, the k-th player publishes memory state $m_{(1,i_1-1)}$, then knowing $\{x^i\}_{i_1 \le i < i_2}$, 1st player adds $m_{(1,i_2-1)}$ to the blackboard and so on. Finally, the last player adds the output of M, given $m_{(p,i_k-1)}$ and knowing $\{x^i\}_{i_k \le i \le n}$. As the $(p+1)$th pass doesn't do any operations, $\forall i, m_{(p+1,i)} = m_{(p,n)}$. Thus, the transcript under Π is the public randomness P, together with a sequence of memory states $m_{(1,i_1-1)}, m_{(1,i_2-1)}, \ldots, m_{(1,i_k-1)}, m_{(2,i_1-1)}, \ldots, m_{(2,i_k-1)}, \ldots, m_{(p,i_k-1)}, m_{(p+1,i_1-1)}$.

When $Z^1, Z^2, \ldots Z^k$ are distributed according to the No distribution for the k-party General Planted Problem, then $X^1, X^2, \ldots, X^n$ are distributed according to D_0, and when $Z^1, Z^2, \ldots Z^k$ are distributed according to the Yes distribution for the k-party General Planted Problem, then $X^1, X^2, \ldots, X^n$ are distributed according to $D_1^{\mathcal{T}}$ with the fixed R. As R is good, the success probability of Π is at least 0.9. By Lemma 4.5,

$$\mathrm{I}\left(\mathbf{\Pi}; Z^1, \ldots, Z^k\right) \ge \Omega\left(\frac{d}{c \cdot t}\right).$$

When $Z^1, \ldots, Z^k$ are distributed according to the No distribution for k-party General Planted Problem, $X^1, X^2, \ldots, X^n$ are distributed according to D_0, and we can rewrite the information complexity of Π as

$$\mathrm{I}\left(P, \mathsf{M}_{(1,i_1-1)}, \mathsf{M}_{(1,i_2-1)}, \ldots, \mathsf{M}_{(1,i_k-1)}, \mathsf{M}_{(2,i_1-1)}, \ldots, \mathsf{M}_{(2,i_k-1)}, \ldots, \mathsf{M}_{(p,i_k-1)}, \mathsf{M}_{(p+1,i_1-1)} \; ; \; X^{i_1}, X^{i_2} \ldots, X^{i_k}\right).$$

Using Chain Rule, we can now rewrite the above mutual information as

$$\begin{aligned}
&\underbrace{I\left(P; X^{i_1}, X^{i_2} \ldots, X^{i_k}\right)}_{=0} + \sum_{\ell=1}^{p} \sum_{a=1}^{k} \mathrm{I}\left(\mathsf{M}_{(\ell,i_a-1)} \; ; \; X^{i_1}, X^{i_2} \ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, \mathsf{M}_{(\ell,\{i_1-1,\ldots,i_{a-1}-1\})}, P\right) + \\
&\qquad\qquad \mathrm{I}\left(\mathsf{M}_{(p+1;i_1-1)} \; ; \; X^{i_1}, X^{i_2} \ldots, X^{i_k} \mid \mathsf{M}_{(\le p,\{i_1-1,\ldots,i_k-1\})}, P\right) \\
&\le \sum_{\ell=1}^{p+1} \sum_{a=1}^{k} \mathrm{I}\left(\mathsf{M}_{(\ell,i_a-1)} \; ; \; X^{i_1}, X^{i_2} \ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, \mathsf{M}_{(\ell,\{i_1-1,\ldots,i_{a-1}-1\})}, P\right). \qquad (28)
\end{aligned}$$

In the inner summation above, let us first consider any summand corresponding to a value of a satisfying $1 < a \le k$. For such a summand, define:

$$\mathsf{M}_{neighbors} = (\mathsf{M}_{<l,i_a-1}, \mathsf{M}_{\le l,i_{a-1}-1}, P), \qquad (29)$$

$$\mathsf{M}_{non-neighbors} = (\mathsf{M}_{\le l,i_1-1}, \ldots, \mathsf{M}_{\le l,i_{a-2}-1}, \mathsf{M}_{<l,i_{a+1}-1}, \ldots, \mathsf{M}_{<l,i_k-1}). \qquad (30)$$

Observe that

$$(\mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}) = (\mathsf{M}_{l,\{i_1-1,\ldots,i_{a-1}-1\}}, \mathsf{M}_{<l,\{i_1-1,\ldots,i_k-1\}}, P).$$

Now, let $X^{\neq i_{a-1}} = (X^{i_1}, \ldots, X^{i_{a-2}}, X^{i_a}, \ldots, X^{i_k})$. Notice then that we can write the summand in the inner summation, using the chain rule, as

$$\begin{aligned} &I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_1}, X^{i_2}\ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, \mathsf{M}_{(\ell,\{i_1-1,\ldots,i_{a-1}-1\})}, P\right) \\ &= I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_{a-1}} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}\right) + I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{\neq i_{a-1}} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}, X^{i_{a-1}}\right) \end{aligned} \tag{31}$$

We now focus on the second term in (31). From (6) in Lemma 3.1, where we consider $i = i_{a-1}, j = i_a$, we know that,

$$I(X^{[i_{a-1},i_a-1]}, R_{([p],[i_{a-1},i_a-1])}\ ;\ X^{[1,i_{a-1}-1]}, R_{([p],[1,i_{a-1}-1])}, X^{[i_a,n]}, R_{([p],[i_a,n])} \mid \underbrace{\mathsf{M}_{\leq l,i_{a-1}-1}, \mathsf{M}_{<l,i_a-1}, P}_{\mathsf{M}_{neighbors}}) = 0$$

$$\implies I(X^{[i_{a-1},i_a-1]}, R_{([p],[i_{a-1},i_a-1])}\ ;\ X^{[1,i_{a-1}-1]}, R_{([p],[1,i_{a-1}-1])}, X^{[i_a,n]}, R_{([p],[i_a,n])} \mid X^{i_{a-1}}, \mathsf{M}_{neighbors}) = 0.$$
$$\text{(since } I(A,B;C|D) = 0 \implies I(A,B;C|B,D) = 0\text{)}$$

Now observe that $\mathsf{M}_{(l,i_a-1)}$ is a deterministic function of $X^{[i_{a-1},i_a-1]}, R_{([p],[i_{a-1},i_a-1])}$—the first argument in the mutual information above—conditioned on $X^{i_{a-1}}, \mathsf{M}_{neighbors}$. Similarly, observe that $(X^{\neq i_{a-1}}, \mathsf{M}_{non-neighbors})$ are deterministic functions of $X^{[1,i_{a-1}-1]}, R_{([p],[1,i_{a-1}-1])}, X^{[i_a,n]}, R_{([p],[i_a,n])}$—the second argument in the mutual information above—conditioned on $X^{i_{a-1}}, \mathsf{M}_{neighbors}$. Accounting this in, we get that

$$I(\mathsf{M}_{(l,i_a-1)}\ ;\ X^{\neq i_{a-1}}, \mathsf{M}_{non-neighbors} \mid X^{i_{a-1}}, \mathsf{M}_{neighbors}) = 0 \tag{32}$$

$$\implies \quad I(\mathsf{M}_{(l,i_a-1)}\ ;\ \mathsf{M}_{non-neighbors} \mid X^{i_{a-1}}, \mathsf{M}_{neighbors}) = 0 \tag{33}$$

$$\text{as well as} \quad I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{\neq i_{a-1}} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}, X^{i_{a-1}}\right) = 0. \tag{34}$$

where the last two implications follow by chain rule. Substituting (34) in (31), we get that

$$\begin{aligned} &I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_1}, X^{i_2}\ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, \mathsf{M}_{(\ell,\{i_1-1,\ldots,i_{a-1}-1\})}, P\right) \\ &= I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_{a-1}} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}\right) \\ &\leq I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_{a-1}} \mid \mathsf{M}_{neighbors}\right) = I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_{a-1}} \mid \mathsf{M}_{<l,i_a-1}, \mathsf{M}_{\leq l,i_{a-1}-1}, P\right). \end{aligned} \tag{35}$$

In the inequality above, we used (33), together with the fact that, if $I(A;B|C,D) = 0$, then $I(C;B|D,A) \leq I(C;B|D)$ (for us, $A = \mathsf{M}_{non-neighbors}, B = \mathsf{M}_{(l,i_a-1)}, C = X^{i_{a-1}}, D = \mathsf{M}_{neighbors}$). In the last equality, we simply recalled the definition (29) of $\mathsf{M}_{neighbors}$.

We now consider the summand in the inner summation in (28) corresponding to $a = 1$. For this summand, define

$$\mathsf{M}_{neighbors} = (\mathsf{M}_{<l,i_1-1}, \mathsf{M}_{<l,i_k-1}, P), \tag{36}$$

$$\mathsf{M}_{non-neighbors} = (\mathsf{M}_{<l,\{i_2-1,\ldots,i_{k-1}-1\}}). \tag{37}$$

We have that

$$(\mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}) = (\mathsf{M}_{<l,\{i_1-1,\ldots,i_k-1\}}, P).$$

Now, let $X^{\neq i_k} = (X^{i_1}, \ldots, X^{i_{k-1}})$. Notice then that we can write the summand in the inner summation, using the chain rule, as

$$I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_1}, X^{i_2}\ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, P\right)$$

$$= I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_k} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}\right) + I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{\neq i_k} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}, X^{i_k}\right) \tag{38}$$

We now focus on the second term in (38). From (5) in Lemma 3.1 where we consider $i = i_1, j = i_k$, we know that,

$$I(X^{[i_1,i_k-1]}, R_{([p],[i_1,i_k-1])}\ ;\ X^{[1,i_1-1]}, R_{([p],[1,i_1-1])}, X^{[i_k,n]}, R_{([p],[i_k,n])} \mid \underbrace{\mathsf{M}_{<l,i_1-1}, \mathsf{M}_{<l,i_k-1}, P}_{\mathsf{M}_{neighbors}}) = 0$$

$$\implies I(X^{[i_1,i_k-1]}, R_{([p],[i_1,i_k-1])}\ ;\ X^{[1,i_1-1]}, R_{([p],[1,i_1-1])}, X^{[i_k,n]}, R_{([p],[i_k,n])} \mid X^{i_k}, \mathsf{M}_{neighbors}) = 0.$$
(since $I(A,B;C|D) = 0 \implies I(A,B;C|B,D) = 0$)

Now observe that $\mathsf{M}_{(l,i_1-1)}$ is a deterministic function of $X^{[1,i_1-1]}, R_{([p],[1,i_1-1])}, X^{[i_k,n]}, R_{([p],[i_k,n])}$—the second argument in the mutual information above—conditioned on $X^{i_k}, \mathsf{M}_{neighbors}$. Similarly, observe that $(X^{\neq i_k}, \mathsf{M}_{non-neighbors})$ are deterministic functions of $X^{[i_1,i_k-1]}, R_{([p],[i_1,i_k-1])}$—the first argument in the mutual information above—conditioned on $X^{i_k}, \mathsf{M}_{neighbors}$. Accounting for this, we get that

$$I(\mathsf{M}_{(l,i_1-1)}\ ;\ X^{\neq i_k}, \mathsf{M}_{non-neighbors} \mid X^{i_k}, \mathsf{M}_{neighbors}) = 0 \tag{39}$$

$$\implies \quad I(\mathsf{M}_{(l,i_1-1)}\ ;\ \mathsf{M}_{non-neighbors} \mid X^{i_k}, \mathsf{M}_{neighbors}) = 0 \tag{40}$$

$$\text{as well as} \quad I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{\neq i_k} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}, X^{i_k}\right) = 0. \tag{41}$$

where the last two implications follow by chain rule. Substituting (41) in (38), we get that

$$I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_1}, X^{i_2}\ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, P\right) = I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_k} \mid \mathsf{M}_{neighbors}, \mathsf{M}_{non-neighbors}\right)$$
$$\leq I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_k} \mid \mathsf{M}_{neighbors}\right) = I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_k} \mid \mathsf{M}_{<l,i_1-1}, \mathsf{M}_{<l,i_k-1}, P\right). \tag{42}$$

In the inequality above, we used (40), together with the fact that, if $I(A;B|C,D) = 0$, then $I(C;B|D,A) \leq I(C;B|D)$ (for us, $A = \mathsf{M}_{non-neighbors}, B = \mathsf{M}_{(l,i_1-1)}, C = X^{i_k}, D = \mathsf{M}_{neighbors}$). In the last equality, we simply recalled the definition (36) of $\mathsf{M}_{neighbors}$.

To conclude, observe that (35) and (42) together imply that (28) is upper bounded as

$$\sum_{\ell=1}^{p+1}\sum_{a=1}^{k} \mathrm{I}\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_1}, X^{i_2}\ldots, X^{i_k} \mid \mathsf{M}_{(<\ell,\{i_1-1,\ldots,i_k-1\})}, \mathsf{M}_{(\ell,\{i_1-1,\ldots,i_{a-1}-1\})}, P\right)$$
$$\leq \sum_{\ell=1}^{p+1} I\left(\mathsf{M}_{(\ell,i_1-1)}\ ;\ X^{i_k} \mid \mathsf{M}_{<l,i_1-1}, \mathsf{M}_{<l,i_k-1}, P\right) + \sum_{\ell=1}^{p+1}\sum_{a=2}^{k} I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_{a-1}} \mid \mathsf{M}_{<l,i_a-1}, \mathsf{M}_{\leq l,i_{a-1}-1}, P\right)$$
$$\leq \sum_{\ell=1}^{p+1}\sum_{a=1}^{k}\sum_{b=a+1}^{k} I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_b} \mid \mathsf{M}_{<l,i_a-1}, \mathsf{M}_{<l,i_b-1}, P\right) + \sum_{\ell=1}^{p+1}\sum_{a=1}^{k}\sum_{b=1}^{a-1} I\left(\mathsf{M}_{(\ell,i_a-1)}\ ;\ X^{i_b} \mid \mathsf{M}_{<l,i_a-1}, \mathsf{M}_{\leq l,i_b-1}, P\right)$$
(non-negativity of mutual information)

$$= MIC^R.$$

This completes the proof.

□

We restate and prove Claim 4.10.

Claim 4.10 (Cut-paste property of the protocol)**.** *For any θ and transcript π, and any $\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3, \mathbf{b}^4$ with $\{b_i^1, b_i^2\} = \{b_i^3, b_i^4\}$ (in a multi-set sense) for every $i \in [m]$,*

$$\Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}^1} = \pi\right] \cdot \Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}^2} = \pi\right] = \Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}^3} = \pi\right] \cdot \Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}^4} = \pi\right],$$

and therefore,

$$h^2\left(\mathbf{\Pi}^\theta_{\mathbf{b}^1} \parallel \mathbf{\Pi}^\theta_{\mathbf{b}^2}\right) = h^2\left(\mathbf{\Pi}^\theta_{\mathbf{b}^3} \parallel \mathbf{\Pi}^\theta_{\mathbf{b}^4}\right).$$

Proof of Claim 4.10. As in [BGM+16] we use certain basic properties of transcripts established in [BYJKS04]. We first note that fixing input $x_1, x_2, \ldots, x_m$, the probability of any transcript can be factored as,

$$\Pr[\Pi(x) = \pi] = p_{1,\pi}(x_1) \cdots p_{m,\pi}(x_m), \tag{43}$$

where $p_{i,\pi}(x_i)$ is some function which depends only on π and x_i. Recall that $\mathbf{\Pi}^\theta_{\mathbf{b}}$ is the distribution of $\Pi(x_1, \ldots, x_m)$ when $(x_1, \ldots, x_m) \sim \mu^\theta_{\mathbf{b}}$, which is a product distribution. Therefore, if $\tilde{X} \sim \mu^\theta_{\mathbf{b}}$ and since $\mu^\theta_{\mathbf{b}}$ is a product measure (for fixed θ), we can marginalize over $\tilde{X}$ and obtain the marginal distribution over the transcripts Π for all $\mathbf{b}$;

$$\Pr[\Pi(\tilde{X}) = \pi] = q_{1,\pi,\theta}(\mathbf{b}_1) \cdots q_{m,\pi,\theta}(\mathbf{b}_m), \tag{44}$$

where $q_{i,\pi,\theta}(\mathbf{b}_i) = \int_{x_i} p_{i,\pi}(x_i) d\mu^\theta_{\mathbf{b}_i}$ is the marginal distribution of $p_{i,\pi}(x_i)$ over $x_i \sim \mu^\theta_{\mathbf{b}_i}$. Therefore, for all $\mathbf{b}$;

$$\Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}} = \pi\right] = q_{1,\pi,\theta}(\mathbf{b}_1) \cdots q_{m,\pi,\theta}(\mathbf{b}_m),$$

and the claim follows because of this decomposition. Since the squared Hellinger distance $h^2\left(\mathbf{\Pi}^\theta_{\mathbf{b}^1} \parallel \mathbf{\Pi}^\theta_{\mathbf{b}^2}\right)$ only depends on the two distributions through the product of the probabilities, that is, $\Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}^1} = \pi\right] \cdot \Pr\left[\mathbf{\Pi}^\theta_{\mathbf{b}^2} = \pi\right]$ for all transcripts π, the result follows. □

We now restate and prove Lemma 4.12

Lemma 4.12. *Consider a family of distributions $\{\mu_\theta\} : \mathcal{X} \to [0,1]$ parameterized by a random variable $\boldsymbol{\theta}$, which takes values in some domain Ω and has distribution P. Consider the distributed detection setting where if $V = 0$ then each party receives $X_i \sim \mu_0$ (for some distribution $\mu_0 : \mathcal{X} \to [0,1]$), and if $V = 1$ then we first draw $\theta \sim P$, and then each party receives $X_i \sim \mu_\theta$. Suppose there is an m-party communication protocol Π that detects whether $V = 0$ or $V = 1$ with probability at least 0.9. Then*

$$\mathbb{E}_{\theta \sim P}\left[h^2(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \geq \Omega(1).$$

Proof of Lemma 4.12. Fix θ. If the protocol succeeds with probability α conditioned on $\boldsymbol{\theta} = \theta$, then we have,

$$\begin{aligned}
&(1/2) \Pr_{\forall i, x^i \sim \mu_0}[\Pi(x_1, x_2, \ldots, x_m) = \text{"No"}] + (1/2) \Pr_{\forall i, x^i \sim \mu_\theta}[\Pi(x_1, x_2, \ldots, x_k) = \text{"Yes"}] = \alpha, \\
\implies &\Pr_{\forall i, x^i \sim \mu_0}[\Pi(x_1, x_2, \ldots, x_m) = \text{"No"}] - \Pr_{\forall i, x^i \sim \mu_\theta}[\Pi(x_1, x_2, \ldots, x_k) = \text{"No"}] \geq 2\alpha - 1, \\
\implies &\|\mathbf{\Pi}_{|V=0} - \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta}\|_{TV} \geq 2\alpha - 1.
\end{aligned}$$

Since the protocol has overall success probability at least 0.9 on average over the randomness in the choice of θ, we have

$$\mathbb{E}_{\theta\sim P}\|\mathbf{\Pi}_{|V=0} - \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta}\|_{TV} \geq 2(0.9) - 1 = 0.8. \tag{45}$$

We will next use the following folklore result to relate the TV distance to the Hellinger distance.

Fact B.1. *For any two distributions P and Q we have,*

$$h^2(P,Q) \leq \|P - Q\|_{TV} \leq \sqrt{2}h(P,Q).$$

Using Fact B.1,

$$\mathbb{E}_{\theta\sim P}\left[h(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \geq \frac{1}{\sqrt{2}}\mathbb{E}_{\theta\sim P}\|\mathbf{\Pi}_{|V=0} - \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta}\|_{TV}. \tag{46}$$

By Jensen's inequality,

$$\mathbb{E}_{\theta\sim P}\left[h^2(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \geq \left(\mathbb{E}_{\theta\sim P}\left[h(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right]\right)^2.$$

Using (46),

$$\mathbb{E}_{\theta\sim P}\left[h^2(\mathbf{\Pi}_{|V=0} \parallel \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta})\right] \geq (1/2)\left(\mathbb{E}_{\theta\sim P}\|\mathbf{\Pi}_{|V=0} - \mathbf{\Pi}_{|V=1,\boldsymbol{\theta}=\theta}\|_{TV}\right)^2.$$

Combining this with (45) completes the proof. □

C Proofs from Section 5

We first restate and prove Claim 5.4

Claim 5.4. *Let $t \geq \frac{Ck^2\log(nm)}{q}$ for some large enough constant C. Suppose S is drawn uniformly at random from all subsets of $[t]$ of size k. Let μ_0 and μ_1 be the probability mass functions of P^0_{trunc} and $\mathbb{E}_S[P^{1,S}_{\text{trunc}}]$ respectively. Then,*

$$\mu_1 \leq O(1)\cdot\mu_0.$$

Proof of Claim 5.4. Note that a draw X from the distribution $\mathbb{E}_S[P^{1,S}_{\text{trunc}}]$ corresponds to first drawing a k-sized $S \subseteq [t]$ uniformly at random, and then drawing $X \sim P^{1,S}_{\text{trunc}}$. Fix any x: we will upper bound the mass that μ_1 assigns to x in terms of the mass that μ_0 assigns to x. This will establish the claim.

Note first that $\mu_0(x) = 1/|T|$, where T was defined in (18). Now let $I_x = \{i \in [t] : x_i = 1\}$, and observe that

$$\mu_1(x) = \sum_{S\subseteq I_x, |S|=k} \mu_1(S)\cdot\mu_1(x|S).$$

But conditioned on S, the distribution $\mu_1(\cdot|S)$ is uniform on the set T_S defined in (19). Noting that every T_S has the same size, this immediately gives us that

$$\mu_1(x) = \frac{1}{|T_S|} \sum_{S \subseteq I_x, |S|=k} \mu_1(S) = \frac{\binom{|I_x|}{k}}{|T_S|\binom{t}{k}}.$$

Thus, we obtain

$$\begin{aligned}
\frac{\mu_1(x)}{\mu_0(x)} = \frac{\binom{|I_x|}{k}|T|}{\binom{t}{k}|T_S|} &= \frac{\binom{|I_x|}{k}\sum_{y=tq-C\sqrt{tq\log(nm)}}^{tq+C\sqrt{tq\log(nm)}}\binom{t}{y}}{\binom{t}{k}\sum_{y=tq-C\sqrt{tq\log(nm)}}^{tq+C\sqrt{tq\log(nm)}}\binom{t-k}{y-k}} \le \frac{\binom{|I_x|}{k}}{\binom{t}{k}}\max_y \frac{\binom{t}{y}}{\binom{t-k}{y-k}} \le \frac{\binom{tq+C\sqrt{tq\log(nm)}}{k}}{\binom{t}{k}}\max_y \frac{\binom{t}{y}}{\binom{t-k}{y-k}} \\
&= \frac{(tq+C\sqrt{tq\log(nm)})!(t-k)!}{t!(tq+C\sqrt{tq\log(nm)}-k)!}\max_y \frac{\binom{t}{y}}{\binom{t-k}{y-k}},
\end{aligned}$$

where we used that $\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \le \max_i \frac{a_i}{b_i}$ for positive a_i, b_i. Observe that

$$\begin{aligned}
\frac{(tq+C\sqrt{tq\log(nm)})!(t-k)!}{t!(tq+C\sqrt{tq\log(nm)}-k)!} &= \frac{(tq+C\sqrt{tq\log(nm)})(tq+C\sqrt{tq\log(nm)}-1)\dots(tq+C\sqrt{tq\log(nm)}-(k-1))}{t(t-1)\dots(t-(k-1))} \\
&= \left(\frac{tq+C\sqrt{tq\log(nm)}}{t}\right)\dots\left(\frac{(t-(k-1))q+C\sqrt{tq\log(nm)}-(1-q)(k-1)}{t-(k-1)}\right) \\
&= q^k\left(1+\frac{C\sqrt{tq\log(nm)}}{tq}\right)\dots\left(1+\frac{C\sqrt{tq\log(nm)}-(1-q)(k-1)}{(t-(k-1))q}\right) \\
&\le q^k\left(1+\frac{C\sqrt{tq\log(nm)}}{(t-(k-1))q}\right)^k \\
&= q^k\left[1+O\left(k\sqrt{\frac{\log(nm)}{tq}}\right)\right],
\end{aligned}$$

where in the last step, we used that $t \ge \frac{Ck^2\log(nm)}{q}$.

Furthermore, for any y, observe also that

$$\begin{aligned}
\frac{\binom{t}{y}}{\binom{t-k}{y-k}} = \frac{t!(y-k)!}{y!(t-k)!} &= \frac{t(t-1)\dots(t-(k-1))}{y(y-1)\dots(y-(k-1))} = \left(1+\frac{t-y}{y}\right)\dots\left(1+\frac{t-y}{y-(k-1)}\right) \\
&\le \left(1+\frac{t-y}{y-(k-1)}\right)^k \le \left(1+\frac{t-tq+C\sqrt{tq\log(nm)}}{tq-C\sqrt{tq\log(nm)}-k+1}\right)^k \\
&\le \left(1+\frac{t-tq+C\sqrt{tq\log(nm)}}{tq-2C\sqrt{tq\log(nm)}}\right)^k \qquad \left(\text{since } k \le \sqrt{\frac{tq}{C\log(nm)}} < C\sqrt{tq\log(nm)}\right) \\
&= \left(\frac{1}{q}+\frac{2C\sqrt{tq\log(nm)}-Cq\sqrt{tq\log(nm)}}{q(tq-2C\sqrt{tq\log(nm)})}\right)^k = \frac{1}{q^k}\left(1+\frac{C(2-q)\sqrt{tq\log(nm)}}{tq-2C\sqrt{tq\log(nm)}}\right)^k
\end{aligned}$$

$$\leq \frac{1}{q^k}\left(1+\frac{2C\sqrt{tq\log(nm)}}{tq-2C\sqrt{tq\log(nm)}}\right)^k \leq \frac{1}{q^k}\left[1+O\left(k\sqrt{\frac{\log(nm)}{tq}}\right)\right],$$

where in the last step, we again used the assumption that $t \geq \frac{Ck^2\log(nm)}{q}$. Thus, we get that

$$\frac{\mu_1(x)}{\mu_0(x)} \leq q^k\left[1+O\left(k\sqrt{\frac{\log(nm)}{tq}}\right)\right]\cdot\frac{1}{q^k}\left[1+O\left(k\sqrt{\frac{\log(nm)}{tq}}\right)\right] \leq 1+O\left(k\sqrt{\frac{\log(nm)}{tq}}\right) = O(1).$$

□

We now restate and prove Theorem 5.2

Theorem 5.2 (Memory Lower Bound for Planted Bi-clique)**.** *Let $0 < q \leq 1/2$ and $0 < k < O\left(\sqrt{\frac{q\cdot n}{\log(nm)}}\right)$. Any p-pass streaming algorithm (using public as well as private randomness), that distinguishes between D_{uniform} and D_{planted} (as in Problem 5.1) when $x^1, x^2, \ldots, x^m$ arrive in a stream requires at least $\Omega\left(\frac{nmq}{pk^4\log(nm)}\right)$ bits of memory.*

Proof. Suppose there was a p-pass streaming algorithm $\mathcal{A}$ that solves Problem 5.1 using only $o\left(\frac{mnq}{pk^4\log n}\right)$ bits of memory. Our approach will be to use the existence of $\mathcal{A}$ to construct a p-pass streaming algorithm $\mathcal{A}'$ for Problem 5.5 that circumvents the lower bound of Lemma 5.6, yielding a contradiction.

For this, let $t = \left\lceil\frac{Ck^2\log(nm)}{q}\right\rceil$ for a suitably large constant C. Let $n' = t\cdot\left\lfloor\frac{n}{t}\right\rfloor$. The algorithm $\mathcal{A}'$ operates as follows: First, using public randomness, it draws $I \subseteq [n]$ of size $n-n'$ uniformly at random, and then draws $b^1, \ldots, b^m \in \{0,1\}^{n-n'}$, where every bit in every b^i is independently drawn as $\text{Ber}(q)$. Then, using public randomness again, it draws a uniformly random permutation π of $[n']$. Upon receiving a stream $z^1, \ldots, z^m$ of n'-bit vectors from an instance of Problem 5.5, algorithm $\mathcal{A}'$ translates this stream into a stream of n-bit vectors $y^1, \ldots, y^m$. Namely, y^i is constructed from z^i as follows: First, y^i_I is assigned to be b^i. Then, z^i is permuted according to π, yielding z^i_π. Finally, $y^i_{[n]\setminus I}$ is assigned to be z^i_π. Observe that $\mathcal{A}'$ can construct this stream $y^1, \ldots, y^m$ using only a constant memory overhead (since the public randomness does not contribute to the memory requirement). The algorithm $\mathcal{A}'$ then feeds this stream $y^1, \ldots, y^m$ to $\mathcal{A}$, and returns the output of $\mathcal{A}$. The total memory requirement of $\mathcal{A}'$ and $\mathcal{A}$ is thus the same, upto an additive constant.

We will now argue that $\mathcal{A}'$ correctly solves Problem 5.5.

Case 1: First, consider the case that $z^1, \ldots, z^m$ were draws from D_0 in Problem 5.5. We observe that on account of permuting uniformly at random according to π, the distribution of $y^1, \ldots, y^m$ that $\mathcal{A}'$ constructs is equivalent to the following random process:

(1) Draw a subset $I \subseteq [n]$ of size $n-n'$ uniformly at random.

(2) Draw $b^1, \ldots, b^m \in \{0,1\}^{n-n'}$, where every bit in every b^i is independently drawn as $\text{Ber}(q)$.

(3) Set $y^i_I = b^i$ for every $i \in [m]$.

(4) Draw a uniformly random partition $\mathcal{T} = \{T_{r'}\}_{r' \in [n'/t]}$ of $[n] \setminus I$, where $\forall r', |T_{r'}| = t$.

---

(5) For every $i \in [m], r' \in [n'/t]$, draw $y^i_{T_{r'}} \sim P^0_{\text{trunc}}$.

But on the other hand, notice also that a draw $x^1, \ldots, x^m$ from D_{uniform} in Problem 5.1 is equivalent to the same random process as above, but with (5) replaced as (5′) ahead:

(5′) For every $i \in [m], r' \in [n'/t]$, draw $x^i_{T_{r'}} \in \{0,1\}^t$, where every bit in $x^i_{T_{r'}}$ is drawn as $\text{Ber}(q)$.

We observe that the distributions of $y^1, \ldots, y^m$ and $x^1, \ldots, x^m$ can be decomposed respectively as as $D_y = \sum_{I,b,\mathcal{T}} D^{I,b,\mathcal{T}} \cdot D_y^{I,b,\mathcal{T}}$ and $D_x = \sum_{I,b,\mathcal{T}} D^{I,b,\mathcal{T}} \cdot D_x^{I,b,\mathcal{T}}$, where the marginal distribution over I, bit-strings $b = \{b^1, \ldots, b^m\}$ and partition $\mathcal{T}$, corresponding to Steps (1)-(4) above, is the *same* in both cases, and $D_y^{I,b,\mathcal{T}}$ and $D_x^{I,b,\mathcal{T}}$ are the conditional distributions according to Step (5) and (5′) respectively. The difference between $D_y^{I,b,\mathcal{T}}$ and $D_x^{I,b,\mathcal{T}}$ is that in $D_y^{I,b,\mathcal{T}}$, every $y^i_{T_{r'}}$ is drawn from P^0_{trunc}; on the other hand, in $D_x^{I,b,\mathcal{T}}$, every $x^i_{T_{r'}}$ is drawn such that every bit in it is an independent $\text{Ber}(q)$. Let A, B be random variables such that A has the distribution of $y^i_{T_{r'}}$ in the former case, whereas B has the distribution of $x^i_{T_{r'}}$ in the latter case. Observe that the distribution of A is identical to the distribution of B, *conditioned* on the event that the number of ones in B is in the range $tq \pm C\sqrt{tq \log(nm)}$. We therefore have that the TV distance between the distributions of A and B is at most the probability that a $Bin(t, q)$ random variable is not in the range $tq \pm C\sqrt{tq \log(nm)}$, which, by a Chernoff bound, is at most $1/(nm)^{10}$ (for suitably large C). Therefore, the TV distance between $D_y^{I,b,\mathcal{T}}$ and $D_x^{I,b,\mathcal{T}}$, which is the TV distance between the product distribution of $m \cdot n'/t$ such random variables, is at most $1/(nm)^9$.

Summarily, we have shown that the distribution of $y^1, \ldots, y^m$, in the case that $z^1, \ldots, z^m$ were drawn from D_0 in Problem 5.5 is $o(1)$ close in TV distance to the distribution D_{uniform} in Problem 5.1.

Case 2: Now, consider the case that $z^1, \ldots, z^m$ were draws from $D_1^{\mathcal{T}}$ in Problem 5.5. The distribution of $y^1, \ldots, y^m$ that $\mathcal{A}'$ constructs can then be described by the random process comprising of Steps (1)-(4) above in Case 1, followed by the steps ahead:

(5) Draw r uniformly at random from $[n'/t]$.

(6) Draw $S \subseteq T_r$ uniformly at random of size k, and $R \subseteq [m]$ uniformly at random of size k.

---

(7) For every $i \notin R$ and $r' \in [n'/t]$, draw $y^i_{T_{r'}} \sim P^0_{\text{trunc}}$.

(8) For every $i \in R$, draw $y^i_{T_r} \sim P^{1,S}_{\text{trunc}}$. Whereas for every $r' \neq r$, draw $y^i_{T_{r'}} \sim P^0_{\text{trunc}}$.

But on the other hand, notice also that a draw $x^1, \ldots, x^m$ from D_{planted} in Problem 5.1 is equivalent to the same random process as above, but with Steps (7) and (8) replaced as (7′) and (8′) ahead:

(7′) For every $i \notin R, r' \in [n'/t]$, draw $x^i_{T_{r'}} \in \{0,1\}^t$, where every bit in $x^i_{T_{r'}}$ is drawn as $\text{Ber}(q)$.

(8′) For every $i \in R$, for every $j \in T_r$, set $x^i_j = 1$ if $j \in S$, else set it to $\text{Ber}(q)$. Whereas for every $r' \neq r$, draw $x^i_{T_{r'}} \in \{0,1\}^t$, where every bit in $x^i_{T_{r'}}$ is drawn as $\text{Ber}(q)$.

Again, we observe that the distributions of $y^1, \ldots, y^m$ and $x^1, \ldots, x^m$ can be decomposed respectively as

$$D_y = \sum_{I,b,\mathcal{T},r,S,R} D^{I,b,\mathcal{T},r,S,R} \cdot D_y^{I,b,\mathcal{T},r,S,R}, \qquad D_x = \sum_{I,b,\mathcal{T},r,S,R} D^{I,b,\mathcal{T},r,S,R} \cdot D_x^{I,b,\mathcal{T},r,S,R},$$

where the marginal distribution over I, bit-strings $b = \{b^1, \ldots, b^m\}$, partition $\mathcal{T}$, planted partition r, planted columns S and planted rows R corresponding to Steps (1)-(6) above, is the *same* in both cases, and $D_y^{I,b,\mathcal{T},r,S,R}$ and $D_x^{I,b,\mathcal{T},r,S,R}$ are the conditional distributions corresponding to Steps (7), (8) and (7′), (8′) respectively. Furthermore, both these conditional distributions are product distributions on $m \cdot n'/t := M$ random variables—denote these as $A^1, \ldots, A^M$ and $B^1, \ldots, B^M$ respectively. Observe that all but k of the random variables A^i are distributed identically as the random variable A in Case 1 above, and the corresponding random variables B^i are distributed identically as the random variable B—the TV distance between the distribution of each such A^i and B^i is hence at most $1/(nm)^{10}$ as reasoned there. The distribution of each of the remaining k random variables A^i is identical to the distribution of the corresponding B^i, if we further condition on the number of ones in B^i to be in the range $tq \pm C\sqrt{tq\log(nm)}$. The probability that the number of ones in B^i is not in this range is the probability that a $Bin(t-k, q)$ random variable is not in the range $tq - k \pm C\sqrt{tq\log(nm)}$, which, by a Chernoff bound, is again at most $1/(nm)^{10}$. Thus, the TV distance between $D_y^{I,b,\mathcal{T},r,S,R}$ and $D_x^{I,b,\mathcal{T},r,S,R}$ is again at most $M \cdot (1/(nm)^{10}) \leq 1/(nm)^9$.

Summarily, we have shown that the distribution of $y^1, \ldots, y^m$, in the case that $z^1, \ldots, z^m$ were drawn from $D_1^{\mathcal{T}}$ in Problem 5.5, is $o(1)$ close in TV distance to the distribution D_{planted} in Problem 5.1.

To conclude, the analysis in Cases 1 and 2 above shows that if $\mathcal{A}$ distinguishes between D_{uniform} and D_{planted} with advantage 0.9 using only $o\left(\frac{nmq}{pk^4\log(nm)}\right) = o\left(\frac{mn'}{pk^2t}\right)$ bits of memory, $\mathcal{A}'$ distinguishes between D_0 and $D_1^{\mathcal{T}}$ with advantage 0.89 using (asymptotically) the same amount of memory. This contradicts Lemma 5.6. □

D Proofs from Section 6

We restate and prove Theorem 6.3 below.

Theorem 6.3. *[Memory Lower Bound for Pattern Planted Bi-Clique] Let $0 < k \leq n$. Any p-pass streaming algorithm that solves Problem 6.2, when $x^1, x^2, \ldots, x^n$ arrive in a stream, requires at least $\Omega\left(\frac{n^2}{pk^3}\right)$ bits of memory.*

Proof of Theorem 6.3. First, we consider the following distinguishing problem:

Problem D.1. *Let $0 < k, n' \leq n$. Let t divide n', $\mathcal{T} = \{T_r\}_{r \in [n'/t]}$ be a partition of $[n']$, where $\forall r, |T_r| = t$. The goal is to distinguish between the following joint distributions on n'-dimensional vectors $z^1, \ldots, z^n$:*

1. *D_0: $\forall i \in [n]$ and $\forall r' \in [n'/t]$, $z^i_{T_{r'}}$ is drawn uniformly at random from $\{0,1\}^t$.*
2. *$D_1^{\mathcal{T}}$: Pick r uniformly from $[n'/t]$. $\forall i \in [n]$ and $\forall r' \neq r$, $z^i_{T_{r'}}$ is drawn uniformly at random from $\{0,1\}^t$.*

   R is drawn uniformly at random from all subsets of $[n]$ of size k. Draw v uniformly at random from $\{0,1\}^t$.

   $\forall i \notin R$, $z^i_{T_r}$ is drawn uniformly at random from $\{0,1\}^t$. Whereas, $\forall i \in R$, $z^i_{T_r}$ is set to v.

We first note that Problem D.1 is a specific instantiation of Problem 4.1, with $n = n$ and $d = n'$. Furthermore, for the purposes of instantiating Theorem 4.2, if we denote by μ_v a point mass on a vector $v \in \{0,1\}^t$, we have that $\mu_1 = \mathbb{E}_v \mu_v = \mu_0$. Thus, Theorem 4.2 guarantees that any algorithm, which is allowed to use public randomness, that distinguishes between D_0 and $D_1^{\mathcal{T}}$ above requires at least $\Omega\left(\frac{nn'}{pk^2t}\right)$ bits of memory.

Now, suppose there was a p-pass streaming algorithm $\mathcal{A}$ that solves Problem 6.2 using only $o\left(\frac{n^2}{pk^3}\right)$ bits of memory. Our approach will be to use the existence of $\mathcal{A}$ to construct a p-pass streaming algorithm $\mathcal{A}'$ that circumvents the lower bound for Problem D.1, yielding a contradiction.

For this, let $t = k$ and $n' = t \cdot \left\lfloor \frac{n}{t} \right\rfloor$. The algorithm $\mathcal{A}'$ operates as follows: First, using public randomness, it draws $I \subseteq [n]$ of size $n - n'$ uniformly at random, and then draws $b^1, \ldots, b^n \in \{0,1\}^{n-n'}$, where every b^i is sampled uniformly at random from $\{0,1\}^{n-n'}$. Then, using public randomness again, it draws a uniformly random permutation π of $[n']$. Upon receiving a stream $z^1, \ldots, z^n$ of n'-bit vectors from an instance of Problem D.1, algorithm $\mathcal{A}'$ translates this stream into a stream of n-bit vectors $y^1, \ldots, y^n$. Namely, y^i is constructed from z^i as follows: First, y^i_I is assigned to be b^i. Then, z^i is permuted according to π, yielding z^i_π. Finally, $y^i_{[n]\setminus I}$ is assigned to be z^i_π. Observe that $\mathcal{A}'$ can construct this stream $y^1, \ldots, y^n$ using only a constant memory overhead (since the public randomness does not contribute to the memory requirement). The algorithm $\mathcal{A}'$ then feeds this stream $y^1, \ldots, y^n$ to $\mathcal{A}$, and returns the output of $\mathcal{A}$. The total memory requirement of $\mathcal{A}'$ and $\mathcal{A}$ is thus the same, upto an additive constant.

We will now argue that $\mathcal{A}'$ correctly solves Problem 5.5.

Case 1: First, consider the case that $z^1, \ldots, z^n$ were draws from D_0 in Problem D.1. The distribution of $y^1, \ldots, y^n$ constructed by $\mathcal{A}'$, since π is a random permutation, is equivalent to the following:

(1) Draw a subset $I \subseteq [n]$ of size $n - n'$ uniformly at random.

(2) Draw $b^1, \ldots, b^n \in \{0,1\}^{n-n'}$, where every b^i is sampled uniformly at random from $\{0,1\}^{n-n'}$.

(3) Set $y^i_I = b^i$ for every $i \in [n]$.

(4) Draw a uniformly random partition $\mathcal{T} = \{T_{r'}\}_{r' \in [n'/t]}$ of $[n] \setminus I$, where $\forall r', |T_{r'}| = t$.

---

(5) For every $i \in [n], r' \in [n'/t]$, draw $y^i_{T_{r'}}$ uniformly at random from $\{0,1\}^t$.

Observe that the distribution of each y_i thus drawn is simply the uniform distribution over $\{0,1\}^n$.

Case 2: We will now reason about the distribution of $y^1, \ldots, y^n$ above when $z^1, \ldots, z^n$ were draws from $D_1^{\mathcal{T}}$ in Problem D.1. The distribution of $y^1, \ldots, y^n$ that $\mathcal{A}'$ constructs can then be described by the random process comprising of Steps (1)-(4) above in Case 1, followed by the steps ahead:

(5) Draw r uniformly at random from $[n'/t]$, v uniformly at random from $\{0,1\}^t$, a subset $R \subseteq [n]$ uniformly at random of size k.

(6) For every $i \notin R, r' \in [n'/t]$, draw $y^i_{T_{r'}}$ uniformly at random from $\{0,1\}^t$.

(7) For every $i \in R$, set $y^i_{T_r} = v$, and for every $r' \neq r$, draw $y^i_{T_{r'}}$ uniformly at random from $\{0,1\}^t$.

Observe that $y_1, \ldots, y_n$ thus drawn is identical in distribution to a draw from D_{planted} in Problem 6.2.

Summarily, we have shown that the the distribution of $y^1, \ldots, y^n$, in the case that $z^1, \ldots, z^n$ were drawn from D_0 in Problem D.1, is identical to the distribution D_{uniform} in Problem 6.2. Similarly, we have also shown that the distribution of $y^1, \ldots, y^n$, in the case that $z^1, \ldots, z^n$ were drawn from $D_1^{\mathcal{T}}$ in Problem D.1, is identical to the distribution D_{planted} in Problem 6.2. So, it follows that if $\mathcal{A}$ distinguishes between D_{uniform} and D_{planted} with advantage 0.9 using only $o\left(\frac{n^2}{pk^3}\right) = o\left(\frac{nn'}{pk^2t}\right)$ bits of memory, $\mathcal{A}'$ distinguishes between D_0 and $D_1^{\mathcal{T}}$ with advantage 0.9 using the same amount of memory (asymptotically). This contradicts the lower bound that we derived in the first paragraph. □

E Proofs from Section 7

We first restate and prove Lemma 7.4

Lemma 7.4. *Let $\epsilon \in (0, 0.01)$ be a constant, d be sufficiently large, $\ell \leq d$, $n \leq d^{10}$ and $\alpha \in \left(\frac{1}{\ell\sqrt{\log d}}, 1\right]$. Let $\mathcal{A}$ be a p-pass streaming algorithm that uses s bits of memory and $n/400$ samples, and solves Problem 7.1 with probability 0.99 for every value of $\ell' \in [2\ell/3, 4\ell/3]$. Then, there exists a p-pass streaming algorithm $\mathcal{A}'$ that uses $s + \tilde{O}(1)$ bits of memory and n samples, and solves Problem 7.3 for $k = nq$ with probability 0.97*

Proof of Lemma 7.4. In this proof, we will construct a reduction between Problem 7.1 and Problem 7.3.

Consider first the intermediate Problem A of distinguishing between:

1. D_0 (no instance): $\forall i \in [n]$, x^i is drawn from $N(0, I_d)$.
2. D_1 (yes instance): Draw $s \sim Bin(t, \ell/t)$. Draw $S \subseteq [d]$ of size s. Obtain $v \in \mathbb{R}^d$, where $v_j = \alpha$ for every $j \in S$, and $v_j = 0$ otherwise. Draw $R \subseteq [n]$ uniformly at random of size nq. For every $i \in [n] \setminus R$, $x^i \sim N(0, I_d)$, whereas for every $i \in R$, $x^i \sim N(v, I_d)$.

The main reason to introduce Problem A above is to get rid of the partition in Problem 7.3—observe that there is no notion of such a "partition" in Problem 7.1. We will first relate the hardness of Problem A to Problem 7.3.

Suppose there were a p-pass algorithm $\mathcal{A}$ that uses s bits of memory and n samples to solve Problem A with probability 0.97. We will show that there exists a p-pass algorithm $\mathcal{A}'$ that uses $s + \tilde{O}(1)$ bits of memory and n samples to solve Problem 7.3 with probability 0.97. The algorithm $\mathcal{A}'$ operates on the input of Problem 7.3 as follows. First, using public randomness, it draws a uniformly random permutation π of $[d]$. Upon receiving the stream $x^1, \ldots, x^n$, it permutes each of $x^1, \ldots, x^n$ according to π, and feeds $x^1_\pi, \ldots, x^n_\pi$ to $\mathcal{A}$. Observe that if $x^1, \ldots, x^n$ were drawn from the no instance of Problem 7.3, then $x^1_\pi, \ldots, x^n_\pi \sim D_0$ above, and if they were drawn from the yes instance, $x^1_\pi, \ldots, x^n_\pi \sim D_1$ above.[8] Therefore, $\mathcal{A}'$ can simply return the output of $\mathcal{A}$, and solve Problem 7.3.

[8] Recall that we assumed for convenience that t divides d. We can handle the case of t not dividing d similarly as we did in the proof of Theorem 5.2. That is, we may instead consider Problem 7.3 with $d' = t \cdot \lfloor d/t \rfloor$. In order to prepare

In the yes instance of Problem A above, there are a fixed number nq of planted vectors; however, in Problem 7.1, the number of planted vectors is nq only *in expectation*. The next intermediate problem bridges this. Concretely, consider Problem B of distinguishing between:

1. D_0 (no instance): $\forall i \in [n/400]$, x^i is drawn from $N(0, I_d)$.
2. D_1 (yes instance): Draw $s \sim Bin(t, \ell/t)$. Draw $S \subseteq [d]$ of size s. Obtain $v \in \mathbb{R}^d$, where $v_j = \alpha$ for every $j \in S$, and $v_j = 0$ otherwise. For every $i \in [n/400]$, $x^i \sim N(v, I_d)$ with probability q, and $x^i \sim N(0, I_d)$ with probability $1 - q$.

Suppose there were a p-pass algorithm $\mathcal{A}$ that uses s bits of memory and $n/400$ samples to solve Problem B with probability 0.98. We will show that there exists a p-pass algorithm $\mathcal{A}'$ that uses $s + \tilde{O}(1)$ bits of memory and n samples to solve Problem A with probability 0.97. The algorithm $\mathcal{A}'$ operates as follows. Upon receiving an input stream $x^1, \ldots, x^n$ from Problem A, it feeds a uniformly random subset of $n/400$ of these vectors to $\mathcal{A}$. Observe first that if the input $x^1, \ldots, x^n$ was from the no instance of Problem A, then the input given to $\mathcal{A}$ is also distributed as the no instance of Problem B. On the other hand, if the input was drawn from the yes instance of Problem A, exactly nq of the vectors in the input were drawn from the planted distribution. Let X denote the number of vectors drawn from the planted distribution that get included in the uniformly random subset of $n/400$ vectors that $\mathcal{A}'$ feeds to $\mathcal{A}$. Now, let Y denote the number of vectors that get drawn from the planted distribution, when the input $x^1, \ldots, x^{n/400}$ is drawn from the yes instance of Problem B above. Observe that the distribution of X corresponds to the number of red balls drawn, when one draws $n/400$ balls uniformly at random from an urn containing n (red and blue) balls of which nq are red *without replacement*, while the distribution of Y corresponds to the number of red balls, when one draws $n/400$ balls uniformly at random from an urn containing n balls of which nq are red *with replacement*. From Theorem (4) in [DF80], we know that the TV distance between the distributions of X and Y is at most 0.01. That is, the input that $\mathcal{A}'$ feeds to $\mathcal{A}$ comprises of $n/400$ vectors, of which a uniformly random subset of X vectors are drawn from the planted distribution, whereas the input of the yes instance of Problem B corresponds to a uniformly random subset of Y vectors drawn from the planted distribution, where $TV(X, Y) \leq 0.01$. Summarily, we conclude that if $\mathcal{A}$ solves Problem B with probability 0.98, $\mathcal{A}'$ solves Problem A with probability 0.97.

Finally, we relate Problem 7.1 to Problem B above. Let $\mathcal{A}$ be a p-pass streaming algorithm that uses s bits of memory and $n/400$ samples, and solves Problem 7.1 with probability 0.99 for every value of $l' \in [2l/3, 4l/3]$. Namely, $\mathcal{A}$ processes $x^1, \ldots, x^{n/400}$ arriving in a stream, and satisfies that:

(1) If $x^1, \ldots, x^{n/400} \sim N(0, I_d)$, then $\mathcal{A}$ outputs no with probability at least 0.99.

(2) For *every* $\ell' \in \left[\frac{2\ell}{3}, \frac{4\ell}{3}\right]$: if $x^1, \ldots, x^{n/400} \sim D_{\text{planted}}$ in Problem 7.1 for sparsity ℓ', then $\mathcal{A}$ outputs yes with probability at least 0.99.

We will argue that $\mathcal{A}$ also solves Problem B. Notice that in the yes instance of Problem B, when $s \sim Bin(t, l/t)$, the probability that $s \in \left[\frac{2\ell}{3}, \frac{4\ell}{3}\right]$ is at least $1 - o(1)$. Together with (2) above, we conclude that, if we simply run $\mathcal{A}$ on input $x^1, \ldots, x^{n/400}$ from Problem B above, then $\mathcal{A}$ outputs the correct answer with probability at least 0.98. This concludes the sequence of reductions.

□

d-dimensional inputs to Problem A from d'-dimensional inputs of Problem 7.3, $\mathcal{A}'$ can first draw n i.i.d. vectors from $N(0, I_{d-d'})$, and assign these at coordinates corresponding to a uniformly random subset of $[d]$ of size $d - d'$. The rest of the d' coordinates may then be assigned to be $x^1_\pi, \ldots, x^n_\pi$. This generates an input for Problem A.

In what follows, we will make use of the elementary claim below at multiple places.

Claim E.1. *Let D be a distribution and let D_{trunc} be the restriction of D to the set T. Then,*

$$\|D - D_{trunc}\|_{TV} = \Pr_{x\sim D}[x \notin T].$$

Proof. Let f and g be the probability density functions for D and D_{trunc} respectively. Note that the definition of the truncated distributions implies that

$$g(x) = \begin{cases} \frac{f(x)}{\int_T f(y)\,dy} & \text{for } x \in T, \\ 0 & \text{for } x \notin T. \end{cases}$$

Let $\overline{T}$ denote the complement of the set T, and let $p_T = \int_T f(y)\,dy$. Note that $p_T < 1$. Then,

$$\begin{aligned}
\|D - D_{\text{trunc}}\|_{TV} &= \frac{1}{2}\int |f(x) - g(x)|\,dx \\
&= \frac{1}{2}\int_{\overline{T}} |f(x) - g(x)|\,dx + \frac{1}{2}\int_T |f(x) - g(x)|\,dx \\
&= \frac{1}{2}\int_{\overline{T}} f(x)\,dx + \frac{1}{2}\int_T \left|f(x) - \frac{f(x)}{p_T}\right|\,dx \\
&= \frac{1}{2}\int_{\overline{T}} f(x)\,dx + \frac{1}{2p_T}\int_T |p_T - 1|\, f(x)\,dx \\
&= \frac{1}{2}\int_{\overline{T}} f(x)\,dx + \frac{1}{2p_T}\int_T (1 - p_T)\, f(x)\,dx \\
&= \frac{1}{2}\int_{\overline{T}} f(x)\,dx + \frac{p_T - p_T^2}{2p_T} = \frac{1}{2}\int_{\overline{T}} f(x)\,dx + \frac{1}{2}(1 - p_T) \\
&= \frac{1}{2}\int_{\overline{T}} f(x)\,dx + \frac{1}{2}\int_{\overline{T}} f(x)\,dx = \Pr_{x\sim D}[x \notin T].
\end{aligned}$$

□

We restate and prove Lemma 7.6

Lemma 7.6. *Let $v \in V_{good}$ be arbitrary. The distributions $N(0, I_t)$ and $N(v, I_t)$ are close (in TV distance) to their respective truncations P^0_{trunc} and $P^{1,v}_{\text{trunc}}$:*

$$\begin{aligned}
\left\|P^0_{\text{trunc}} - N(0, I_t)\right\|_{TV} &\le 0.01/(nd/t), \\
\left\|P^{1,v}_{\text{trunc}} - N(v, I_t)\right\|_{TV} &\le 0.01/(nd/t).
\end{aligned}$$

Also, $\Pr_{v\sim D}[v \in V_{good}] \ge 0.99$.

Proof of Lemma 7.6. The claim that $\Pr_v[v \in V_{good}] \ge 0.99$ follows from Markov's inequality, as the expected sparsity of a vector drawn from D is ℓ, and V_{good} consists of vectors whose sparsity is at most 100ℓ. The remainder of this proof is devoted to bounding the TV distances for the truncated distributions. Since $0 \in V_{good}$, it suffices to show that $\|P^v_{\text{trunc}} - N(v, I_t)\|_{TV} \le 0.01/(nd/t)$ for an arbitrary vector $v \in V_{good}$

Computing $\|P^v_{\text{trunc}} - N(v, I_t)\|_{TV}$. By Claim E.1, the TV distance is precisely the probability $\Pr_{x\sim N(0,I_t)}[x \notin T]$, where T is the set defined in Equation (22). We split this probability into two terms and bound each separately. In particular, we define the set $T' = \left\{x \in \mathbb{R}^t : \forall j \in [t], |x_j| \le \sqrt{C_1 \log(200nd)}\right\}$ of values x with bounded coordinates (where C_1 is a positive constant that will be determined later). Note that

$$\begin{aligned}\Pr_{x\sim N(v,I_t)}[x \notin T] &= \Pr_{x\sim N(v,I_t)}[x \notin T, x \in T'] + \Pr_{x\sim N(v,I_t)}[x \notin T, x \notin T'] \\ &\le \Pr_{x\sim N(v,I_t)|x\in T'}[x \notin T] + \Pr_{x\sim N(v,I_t)}[x \notin T']. \end{aligned} \tag{47}$$

The majority of our analysis is devoted to obtaining a bound for the first term. Recalling the definition of T, we have

$$\Pr_{x\sim N(v,I_t)|x\in T'}[x \notin T] = \Pr_{x\sim N(v,I_t)|x\in T'}\left[\sum_{j=1}^{t} e^{\alpha x_j} \ge te^{\alpha^2/2} + \delta\right],$$

where $\delta = (C_1\alpha)\sqrt{t}d^{\epsilon/2}\log(200nd)$. We will apply a concentration inequality for the sum of independent, bounded random variables. First, we bound their expected sum.

Claim E.2. *For any vector $v \in V_{good}$,*

$$\mathbb{E}_{x\sim N(v,I_t)|x\in T'}\left[\sum_{j=1}^{t} e^{\alpha x_j}\right] \le te^{\alpha^2/2} + \delta/2.$$

Proof. Let $B = \sqrt{C_1 \log(200nd)}$. Since conditioning on the set T' preserves independence between the coordinates of x, we first derive an upper bound for the following quantity

$$\mathbb{E}_{X\sim N(v_j,1)}\left[e^{\alpha X} \,\middle|\, |X| \le B\right] = \frac{\int_{-B}^{B} e^{\alpha z} \cdot \frac{1}{\sqrt{2\pi}} e^{-(z-v_j)^2/2} dz}{\Pr_{X\sim N(v_j,1)}[|X| \le B]},$$

We now bound the numerator. Observe that

$$\begin{aligned}\int_{-B}^{B} e^{\alpha z} \cdot \frac{1}{\sqrt{2\pi}} e^{-(z-v_j)^2/2} dz &= \int_{-B}^{B} e^{\alpha^2/2+v_j\alpha} \cdot \frac{1}{\sqrt{2\pi}} e^{-(z-(\alpha+v_j))^2/2} dz \\ &= e^{\alpha^2/2+v_j\alpha}\left(\Pr_{X\sim N(\alpha+v_j,1)}[X \le B] - \Pr_{X\sim N(\alpha+v_j,1)}[X \le -B]\right) \\ &= e^{\alpha^2/2+v_j\alpha}\left(\Pr_{X\sim N(0,1)}[X \le B-\alpha-v_j] - \Pr_{X\sim N(0,1)}[X \le -B-\alpha-v_j]\right) \\ &= e^{\alpha^2/2+v_j\alpha}\left(\Phi(B-\alpha-v_j) - \Phi(-B-\alpha-v_j)\right),\end{aligned}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of a $N(0,1)$ random variable. Consider the function $f(y) = \Phi(B-y) - \Phi(-B-y)$, and observe that

$$f'(y) = -\phi(B-y) + \phi(-B-y) = \phi(B+y) - \phi(B-y),$$

where ϕ denotes the probability density function of a $N(0,1)$ random variable. The second equality follows from the symmetry of the density function ϕ. We claim that $f'(y) \le 0$ for $y \ge 0$, and hence f is non-increasing when $y \ge 0$. Indeed, if $y \le B$, then $B + y \ge B - y \ge 0$ and since ϕ is non-increasing for non-negative arguments, we in turn have $\phi(B-y) \ge \phi(B+y)$. On the other hand, if $y > B$, then $-B - y \le B - y < 0$ and since ϕ is increasing for negative arguments, we in turn have $\phi(B-y) > \phi(-B-y)$. Thus, f is non-increasing for non-negative arguments. Since $\alpha \ge 0$ and $v_j \ge 0$, we have $f(\alpha + v_j) \le f(v_j)$, which implies that

$$\begin{aligned}\Phi(B - \alpha - v_j) - \Phi(-B - \alpha - v_j) &\le \Phi(B - v_j) - \Phi(-B - v_j) \\ &= \Pr_{X \sim N(v_j,1)}[|X| \le B].\end{aligned}$$

It follows that our numerator is bounded above by $e^{\alpha^2/2 + v_j\alpha} \Pr_{X\sim N(v_j,1)}[|X| \le B]$ and thus, the conditional expectation is bounded above by $e^{\alpha^2/2+v_j\alpha}$. By linearity of expectation, we have that

$$\mathbb{E}_{x \sim N(v, I_t) | x \in T'}\left[\sum_{j=1}^{t} e^{\alpha x_j}\right] \le e^{\alpha^2/2} \sum_{j=1}^{t} e^{\alpha v_j}.$$

It remains to upper bound the right hand side of the above inequality. Since $v \in V_{good}$, it has sparsity $\kappa \le 100\ell$ and any nonzero coordinate is by definition equal to α. We therefore have

$$\begin{aligned}e^{\alpha^2/2} \sum_{j=1}^{t} e^{\alpha v_j} &= e^{\alpha^2/2}\left(t - \kappa + \kappa e^{\alpha^2}\right) \\ &= te^{\alpha^2/2} + \kappa e^{\alpha^2/2}(e^{\alpha^2} - 1) \\ &\le te^{\alpha^2/2} + \kappa e^{\alpha^2/2}(2\alpha^2) && \text{(since } e^x \le 1 + 2x \text{ for } x \in (0,1]\text{)} \\ &\le te^{\alpha^2/2} + 200\ell e^{\alpha^2/2}\alpha^2. && \text{(since } \kappa \le 100\ell\text{)}\end{aligned}$$

Furthermore, taking d sufficiently large and recalling that $t \ge (\ell\alpha)^2 d^\epsilon \log^2(200nd)$, we get

$$\begin{aligned}\delta/2 &= (C_1\alpha)\sqrt{t}d^{\epsilon/2}\log(200nd)/2 \\ &\ge C_1\ell\alpha^2 d^\epsilon \log^2(200nd)/2 \\ &\ge 200\ell e^{\alpha^2/2}\alpha^2,\end{aligned}$$

which proves our desired result. □

Next, we recall that conditioning on the set T' when x is drawn from an isotropic normal distribution preserves independence between the coordinates of x. Hence, we can apply Hoeffding's inequality to the random variables $Z_j = e^{\alpha x_j}$, where x is drawn from $N(v, I_t)$ restricted to the set T'. Note that the preceding Claim E.2 implies that

$$\Pr\left[\sum_{j=1}^{t} Z_j \ge te^{\alpha^2/2} + \delta\right] \le \Pr\left[\sum_{j=1}^{t} Z_j - \mathbb{E}\left[\sum_{j=1}^{t} Z_j\right] \ge \delta/2\right].$$

We will consider two cases based on the magnitude of α. First, suppose that $1/\sqrt{C_1 \log(200nd)} \le \alpha \le 1$. Then, we have that the variables Z_j are bounded as

$$0 \le e^{\alpha x_j} := Z_j \le e^{\sqrt{C_1 \log(200nd)}} \le e^{\sqrt{C_1 \log(200d^{11})}} \le e^{\frac{\epsilon}{2} \log d} = d^{\epsilon/2},$$

where we used that $n \le d^{10}$, and that d is sufficiently large. Thus, Hoeffding's inequality gives that

$$\begin{aligned}
\Pr\left[\sum_{j=1}^{t} Z_j - \mathbb{E}\left[\sum_{j=1}^{t} Z_j\right] \ge \delta/2\right] &\le \exp\left(-\frac{2(\delta/2)^2}{td^{\epsilon}}\right) \\
&= \exp\left(-\frac{C_1^2 \alpha^2 td^{\epsilon} \log^2(200nd)}{2td^{\epsilon}}\right) = \exp\left(-\frac{C_1^2 \alpha^2 \log^2(200nd)}{2}\right) \\
&\le \exp\left(-\frac{C_1 \log(200nd)}{2}\right) \qquad \text{(since } \alpha \ge 1/\sqrt{C_1 \log(200nd)}\text{)} \\
&\le 0.005/(nd) \le 0.005/(nd/t).
\end{aligned}$$

The second-to-last inequality holds whenever $C_1 \ge 2$.

Now suppose that $0 < \alpha \le 1/\sqrt{C_1 \log(200nd)}$. In this case, we use the following bound on the variables Z_j:

$$e^{-\alpha\sqrt{C_1 \log(200nd)}} \le e^{\alpha x_j} := Z_j \le e^{\alpha\sqrt{C_1 \log(200nd)}}.$$

Applying Hoeffding's inequality then gives that

$$\begin{aligned}
\Pr\left[\sum_{j=1}^{t} Z_j - \mathbb{E}\left[\sum_{j=1}^{t} Z_j\right] \ge \delta/2\right] &\le \exp\left(-\frac{2(\delta/2)^2}{t\left(e^{\alpha\sqrt{C_1 \log(200nd)}} - e^{-\alpha\sqrt{C_1 \log(200nd)}}\right)^2}\right) \\
&\le \exp\left(-\frac{\delta^2}{2t(3\alpha\sqrt{C_1 \log(200nd)})^2}\right) \qquad (e^y - e^{-y} \le 3y \text{ if } 0 \le y \le 1) \\
&= \exp\left(-\frac{C_1^2 \alpha^2 td^{\epsilon} \log^2(200nd)}{18t\alpha^2 C_1 \log(200nd)}\right) = \exp\left(-\frac{C_1 d^{\epsilon} \log(200nd)}{18}\right) \\
&\le 0.005/(nd) \le 0.005/(nd/t).
\end{aligned}$$

The second-to-last inequality holds whenever $C_1 \ge 18$ and d is sufficiently large.

In both cases, we have obtained an upper bound of $0.005/(nd/t)$ for the first term $\Pr_{x \sim N(v, I_t) | x \in T'}[x \notin T]$ in (47). Finally, we compute an upper bound on the second term $\Pr_{x \sim N(v, I_t)}[x \notin T']$.

Claim E.3. *For sufficiently large d,*

$$\Pr_{x \sim N(v, I_t)}[x \notin T'] \le 0.005/(nd/t).$$

Proof. By a union bound, we note that the left hand side is at most

$$\sum_{j=1}^{t} \Pr_{x_j \sim N(v_j, 1)}\left[|x_j| \ge \sqrt{C_1 \log(200nd)}\right] \le 2 \sum_{j=1}^{t} \Pr_{x_j \sim N(v_j, 1)}\left[x_j \ge \sqrt{C_1 \log(200nd)}\right]$$

$$
\begin{aligned}
&\leq 2t \Pr_{z\sim N(1,1)}\left[z \geq \sqrt{C_1 \log(200nd)}\right] = 2t \Pr_{z\sim N(0,1)}\left[z \geq \sqrt{C_1 \log(200nd)} - 1\right] \\
&\leq \frac{2t}{\sqrt{2\pi}\left(\sqrt{C_1 \log(200nd)} - 1\right)} \exp\left(-\frac{\left(\sqrt{C_1 \log(200nd)} - 1\right)^2}{2}\right) && \text{(Mill's inequality)} \\
&= \frac{\sqrt{2}t}{\sqrt{\pi}\left(\sqrt{C_1 \log(200nd)} - 1\right)} \exp\left(\frac{-C_1 \log(200nd) - 1 + 2\sqrt{C_1 \log(200nd)}}{2}\right) \\
&= \frac{\sqrt{2}t(200nd)^{-C_1/2}}{\sqrt{\pi e}\left(\sqrt{C_1 \log(200nd)} - 1\right)} \exp\left(\sqrt{C_1 \log(200nd)}\right) \\
&\leq \frac{t(200d^{11})^{-C_1/2}}{2\left(\sqrt{C_1 \log(200nd)} - 1\right)} \exp\left(\sqrt{C_1 \log(200d^{11})}\right) \leq \frac{td^{-6.5C_1+\epsilon}}{\sqrt{C_1 \log(200nd)}} && (d \text{ sufficiently large}) \\
&\leq td^{-6.5C_1+0.01} \leq 0.005/(d^{11}/t) \leq 0.005/(nd/t).
\end{aligned}
$$

The inequality in the first line follows from the fact that $v_j \in [0,1]$ (and hence the right tail has more probability mass). The second inequality follows from the fact that each $v_j \leq 1$. The final inequalities use that d is sufficiently large, and that $C_1 > 5$ (say). □

To conclude, we have upper bounded the sum of the two terms on the right in (47) by $0.01/(nd/t)$ as desired. Note that to resolve all the dependencies on C_1, we can take $C_1 = 20$ (say).

□

We now restate and prove Claim 7.7.

Claim 7.7. *Let μ_0 and μ_1^v be the probability density functions of P^0_{trunc} and $P^{1,v}_{\text{trunc}}$ respectively. Then, there exists a positive constant C such that*

$$\mathbb{E}_{v\sim D_{good}}[\mu_1^v] \leq C\mu_0.$$

Proof of Claim 7.7. Since μ_0 and μ_1^v are distributions truncated to the set T defined in Equation (22), the Gaussian densities need to normalized with the appropriate normalizing constants. However, we will first get a bound for the unnormalized densities, and then deal with the normalization.

Let f_0 and f_v respectively be the probability density functions for the (non-truncated) Gaussian distributions $N(0, I_t)$ and $N(v, I_t)$. Notice that for any x, we have

$$
\begin{aligned}
f_v(x) = (2\pi)^{-t/2} \cdot \exp\left(-\frac{1}{2}(x-v)^\intercal(x-v)\right) &= (2\pi)^{-t/2} \cdot \exp\left(-\frac{1}{2}x^\intercal x\right) \cdot \exp\left(x^\intercal v - \frac{v^\intercal v}{2}\right) \\
&= f_0(x) \cdot \exp\left(x^\intercal v - \frac{v^\intercal v}{2}\right).
\end{aligned}
$$

This in turn implies that, for $v \sim D$ as defined in Equation (21) and $x \in T$,

$$\mathbb{E}_{v\sim D}\frac{f_v(x)}{f_0(x)} = \mathbb{E}_v \exp\left(x^\intercal v - \frac{v^\intercal v}{2}\right)$$

$$
\begin{aligned}
&= \prod_{j=1}^{t} \mathbb{E}_{v_j} \exp\left(x_j v_j - \frac{v_j^2}{2}\right) && \text{(coordinates of } v \sim D \text{ are independent)} \\
&= \prod_{j=1}^{t} \left(1 - \frac{\ell}{t} + \frac{\ell}{t} \cdot e^{-\alpha^2/2} e^{\alpha x_j}\right) && (v_j \text{ is } \alpha \text{ w.p. } \ell/t \text{ and 0 otherwise}) \\
&\le \exp(-\ell) \cdot \exp\left(\frac{\ell}{t} \cdot e^{-\alpha^2/2} \sum_{j=1}^{t} e^{\alpha x_j}\right) \\
&\le \exp(-\ell) \cdot \exp\left(\frac{\ell}{t} e^{-\alpha^2/2} \left(t e^{\alpha^2/2} + \delta\right)\right) && (\text{since } x \in T) \\
&= \exp\left(\frac{\ell\delta}{t} e^{-\alpha^2/2}\right),
\end{aligned}
$$

where $\delta = (C_1\alpha)\sqrt{t} d^{\epsilon/2} \log(200nd)$. Since $t \ge (\alpha\ell)^2 d^{\epsilon} \log^2(200nd)$, we further have that

$$
\begin{aligned}
\exp\left(\frac{\ell}{t} e^{-\alpha^2/2} \delta\right) &= \exp\left(\frac{\ell e^{-\alpha^2/2} (C_1\alpha) d^{\epsilon/2} \log(200nd)}{\sqrt{t}}\right) \\
&\le \exp\left(\frac{\ell e^{-\alpha^2/2} (C_1\alpha) d^{\epsilon/2} \log(200nd)}{(\alpha\ell) d^{\epsilon/2} \log(200nd)}\right) \le C',
\end{aligned}
$$

for some constant C'. Now, by the law of total probability, we have

$$
\begin{aligned}
\mathbb{E}_{v\sim D}\left[\frac{f_v}{f_0}\right] &= \Pr_v[v \in V_{good}] \cdot \mathbb{E}_{v|v\in V_{good}}\left[\frac{f_v}{f_0}\right] + \Pr_v[v \notin V_{good}] \cdot \mathbb{E}_{v|v\notin V_{good}}\left[\frac{f_v}{f_0}\right] \\
&\ge \Pr_v[v \in V_{good}] \cdot \mathbb{E}_{v|v\in V_{good}}\left[\frac{f_v}{f_0}\right] = \Pr_v[v \in V_{good}] \cdot \mathbb{E}_{v\sim D_{good}}\left[\frac{f_v}{f_0}\right].
\end{aligned}
$$

The inequality above follows since probability density functions are non-negative. The last equality follows since the distribution of v conditioned on $v \in V_{good}$ is precisely the distribution D_{good}.

Next, since $\mathbb{E}_{v\sim D}[\|v\|_0] = \ell$, we note by Markov's inequality that $\Pr_v[v \in V_{good}] \ge 0.99$. Therefore,

$$
\mathbb{E}_{D_{good}}\left[\frac{f_v}{f_0}\right] \le 2 \cdot \mathbb{E}_{v\sim D}\left[\frac{f_v}{f_0}\right].
$$

We will now tackle our normalizing constants. Let $f_0(T) = \Pr_{x\sim N(0,I_t)}[x \in T]$ and $f_v(T) = \Pr_{x\sim N(v,I_t)}[x \in T]$. From Claim E.1 and Lemma 7.6, we know that $f_v(T) \ge 1 - \frac{0.01t}{nd} \ge 0.99$, which immediately gives that $f_0(T)/f_v(T) \le 1/0.99 \le 2$. Finally,

$$
\begin{aligned}
\mathbb{E}_{v\sim D_{good}}\left[\frac{\mu_1^v}{\mu_0}\right] &= \mathbb{E}_{v\sim D_{good}}\left[\frac{f_v}{f_0} \cdot \frac{f_0(T)}{f_v(T)}\right] \\
&\le 2 \cdot \mathbb{E}_{v\sim D_{good}}\left[\frac{f_v}{f_0}\right] \\
&\le 4 \cdot \mathbb{E}_{v\sim D}\left[\frac{f_v}{f_0}\right]
\end{aligned}
$$

$$\leq 4C'.$$

The desired result follows by taking $C = 4C'$.

□

We restate and prove Claim 7.8.

Claim 7.8. *Fix a constant $\delta \in (0,1)$ and let $C_{\delta,\alpha} = \left(\frac{8+4\log(4/\delta)}{\alpha^2}\right)$. For all n, d sufficiently large that satisfy $nq \geq 2C_{\delta,\alpha}\log(nd)$, the following holds. If $|R| = 2C_{\delta,\alpha}(d/\ell)\log(nd/\delta)\log(nd)$, $\ell \geq s_1 = s_2 = C_{\delta,\alpha}\log(nd)$ and $\tau = \sqrt{2s_1s_2\log\left(2\binom{n}{s_1}\binom{|R|}{s_2}/\delta\right)}$, then*

$$\max\left\{\Pr_{D_{\text{null}}}\left[\max_{\substack{S_1\subseteq[n],|S_1|=s_1\\ S_2\subseteq R,|S_2|=s_2}}\sum_{j\in S_1}\sum_{i\in S_2}x_i^j \geq \tau\right], \Pr_{D_{\text{planted}}}\left[\max_{\substack{S_1\subseteq[n],|S_1|=s_1\\ S_2\subseteq R,|S_2|=s_2}}\sum_{j\in S_1}\sum_{i\in S_2}x_i^j \leq \tau\right],\right\} \leq \delta.$$

Proof of Claim 7.8. Throughout this proof, we will let $Y_{S_1,S_2} = \sum_{j\in S_1}\sum_{i\in S_2}x_i^j$ for ease of exposition. We will bound the failure probability for D_{null} and D_{planted} separately.

We first bound the probability that the test fails to detect the null distribution D_{null}. It is straightforward to verify that when $x^1, \ldots x^n \sim D_{\text{null}}$, we have $Y_{S_1,S_2} \sim N(0, s_1s_2)$ for every s_1-sized subset $S_1 \subseteq \{x^j\}_{j=1}^n$ and s_2-sized subset $S_2 \subseteq R$. . By a union bound, the probability that one of the test statistics exceeds τ is at most

$$\begin{aligned}\binom{n}{s_1}\binom{|R|}{s_2}\Pr_{Y\sim N(0,s_1s_2)}[Y \geq \tau] &\leq \binom{n}{s_1}\binom{|R|}{s_2}\exp\left(-\frac{\tau^2}{2s_1s_2}\right)\\ &\leq \binom{n}{s_1}\binom{|R|}{s_2}\cdot\exp\left(-\frac{\left(\sqrt{2s_1s_2\log\left(2\binom{n}{s_1}\binom{|R|}{s_2}/\delta\right)}\right)^2}{2s_1s_2}\right)\\ &\leq \delta/2.\end{aligned}$$

Hence the failure probability for the null distribution is bounded as desired.

We now bound the probability that the test fails to detect the planted distribution D_{planted}. It will be convenient for us to define two events. Let $\mathcal{E}_1$ be the event that at least s_1 of the samples are drawn from the distribution $N(v, I_d)$. Let $\mathcal{E}_2$ be the event that $|A \cap R| \geq s_2$, where A is the support of the planted vector v. Note that if the event $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs, then there will be some pair of subsets S_1, S_2 that contain signal from the planted vector. If the statistic Y_{S_1,S_2} exceeds the threshold, then we would correctly detect the planted distribution. It follows that the failure probability is at most

$$\Pr\left[\neg(\mathcal{E}_1 \cap \mathcal{E}_2)\right] + \Pr_{D_{\text{planted}}}\left[\max_{\substack{S_1\subseteq[n],|S_1|=s_1\\ S_2\subseteq R,|S_2|=s_2}}\sum_{j\in S_1}\sum_{i\in S_2}x_i^j < \tau \,\middle|\, \mathcal{E}_1 \cap \mathcal{E}_2\right]. \tag{48}$$

We will bound each of these terms separately. We begin with the term $\neg(\mathcal{E}_1 \cap \mathcal{E}_2)$. Note that the number of samples from the distribution $N(v, I_d)$ will follow a binomial distribution $\text{Bin}(n, q)$ (and have mean nq). Hence, we can show that

$$\Pr[\neg\mathcal{E}_1] \leq \Pr_{Z\sim\text{Bin}(n,q)}[Z \leq s_1]$$

$$
\begin{aligned}
&\leq \Pr_{Z\sim\mathrm{Bin}(n,q)}\left[Z \leq \frac{1}{2}nq\right] && \text{(since } nq \geq 2s_1\text{)}\\
&\leq \exp\left(-\frac{(nq)(1/2)^2}{2}\right) && \text{(Chernoff bound)}\\
&\leq \exp\left(-\frac{C_{\delta,\alpha}\log(nd)}{4}\right) && \text{(since } nq \geq C_{\delta,\alpha}\log(nd)\text{)}\\
&\leq (nd)^{-2} && \text{(since } C_{\delta,\alpha} \geq 8\text{)}\\
&\leq \delta/4 && \text{(for sufficiently large } n, d\text{)}
\end{aligned}
$$

Next, note that the number of coordinates in the intersection $|A \cap R|$ will follow the hypergeometric distribution Hypergeometric$(d, \ell, |R|)$ (and have mean $|R|(\ell/d) = 2C_{\delta,\alpha}\log(nd/\delta)\log(nd)$). Hence, we can show that

$$
\begin{aligned}
\Pr[\neg\mathcal{E}_2] &\leq \Pr_{Z\sim\mathrm{Hyp}(d,\ell,|R|)}[Z \leq s_2]\\
&\leq \Pr_{Z\sim\mathrm{Bin}(|R|,\ell/d)}[Z \leq s_2]\\
&= \Pr_{Z\sim\mathrm{Bin}(|R|,\ell/d)}\left[Z \leq \frac{1}{2\log(nd/\delta)}\cdot|R|(\ell/d)\right]\\
&\leq \Pr_{Z\sim\mathrm{Bin}(|R|,\ell/d)}\left[Z \leq \frac{1}{2}\cdot|R|(\ell/d)\right] && \text{(for sufficiently large } n, d\text{)}\\
&\leq \exp\left(-\frac{|R|(\ell/d)(1/2)^2}{2}\right) && \text{(Chernoff bound)}\\
&= \exp\left(-\frac{C_{\delta,\alpha}\log(nd)\log(nd/\delta)}{4}\right)\\
&\leq \exp\left(-\frac{C_{\delta,\alpha}\log(nd)}{4}\right) && \text{(for sufficiently large } n, d\text{)}\\
&\leq \delta/4
\end{aligned}
$$

In the second line we made use of the well-known fact that the binomial distribution stochastically dominates the hypergeomtric distribution. The final inequality follows from observing a similar expression in the calculation for $\Pr[\neg\mathcal{E}_1]$.

Note that by a union bound we have $\Pr[\neg(\mathcal{E}_1 \cap \mathcal{E}_2)] \leq \delta/2$. It remains to show that the second term in Equation (48) is also upper bounded by $\delta/2$. Recall that if the event $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs, then then there will be some pair of subsets S_1, S_2 such that every coordinate contains signal from the planted vector. Note also that sum of these entries follows the Gaussian distribution $N(\alpha s_1 s_2, s_1 s_2)$. It is not hard to see that

$$
\Pr_{D_{\text{planted}}}\left[\max_{\substack{S_1\subseteq[n],|S_1|=s_1\\ S_2\subseteq R,|S_2|=s_2}}\sum_{j\in S_1}\sum_{i\in S_2}x_i^j < \tau \;\middle|\; \mathcal{E}_1\cap\mathcal{E}_2\right] \leq \Pr_{Z\sim N(\alpha s_1 s_2, s_1 s_2)}[Z < \tau] = \Pr_{Z\sim N(0,1)}\left[Z < \frac{\tau - s_1 s_2\alpha}{\sqrt{s_1 s_2}}\right].
$$

Let $t = \sqrt{2\log(4/\delta)}$. It suffices to show that $s_1 s_2\alpha \geq \tau + t\sqrt{s_1 s_2}$. Indeed, the application of standard Gaussian tail bound would imply that the second term in Equation (48) is at most $\delta/4$. This in turn would confirm that the total failure probability is at most $3\delta/4$.

We now show through a series of calculations that the desired inequality holds. We will in fact show that

$$\alpha^2 \geq 2\left(\frac{\tau}{s_1 s_2}\right)^2 + 2\left(\frac{t}{\sqrt{s_1 s_2}}\right)^2 \geq \left(\frac{\tau + t\sqrt{s_1 s_2}}{s_1 s_2}\right)^2$$

Note that the second inequality follows from the fact that $2a^2 + 2b^2 \geq (a+b)^2$, so we only need to establish the first inequality.

Let $s = s_1 = s_2$ and $N = \binom{n}{s_1}\binom{|R|}{s_2}$. Since $t^2 = 2\log(4/\delta)$, we can rewrite the constant $C_{\delta,\alpha} = \frac{8+2t^2}{\alpha^2}$. Equivalently, we have

$$\alpha^2 = \frac{8+2t^2}{C_{\delta,\alpha}} = \frac{4}{C_{\delta,\alpha}} + \frac{4+2t^2}{C_{\delta,\alpha}}.$$

We will show that this value of α is in fact sufficient. That is, we will show that

$$\alpha^2 \geq \frac{2\tau^2}{s^4} + \frac{2t^2}{s^2}$$

Substituting the definition of $\tau^2 = 2s^2\log(2N/\delta)$ into the expression on the right hand side of our target inequality, we can derive that

$$\begin{aligned}
\frac{2\tau^2}{s^4} + \frac{2t^2}{s^2} &= \frac{2\left(2s^2\log(2N/\delta)\right)}{s^4} + \frac{2t^2}{s^2} \\
&= \frac{4\log(N)}{s^2} + \frac{4\log(2/\delta) + 2t^2}{s^2} \\
&\leq \frac{4\left(\log\binom{n}{s} + \log\binom{|R|}{s}\right)}{s^2} + \frac{4\log(2/\delta) + 2t^2}{s^2} \\
&\leq \frac{4\log(n|R|)}{s} + \frac{4\log(2/\delta) + 2t^2}{s^2} \\
&\leq \frac{4\log(nd)(1+o(1))}{s} + \frac{4\log(2/\delta) + 2t^2}{s^2} \\
&= \frac{4(1+o(1))}{C_{\delta,\alpha}} + \frac{4\log(2/\delta) + 2t^2}{s^2} \\
&= \frac{4}{C_{\delta,\alpha}} + \left(\frac{4\cdot o(1)}{C_{\delta},\alpha} + \frac{4\log(2/\delta) + 2t^2}{s^2}\right)
\end{aligned}$$

We note that for sufficiently large n, d the second term in the above expression tends to 0 and in particular is less than the constant $\frac{4+2t^2}{C_{\delta,\alpha}}$. Thus, our value of α is indeed sufficient, as desired.

□

F Proofs from Section 8

In what follows, we will make use of the following observation at multiple points.

Claim F.1. *For any distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$, where $\Sigma_S = I_t + \alpha vv^\intercal$ and $v = \frac{1}{\sqrt{\ell}} 1_S$, and any set $R \in \mathcal{S} = \{[1, \ell], [\ell + 1, 2\ell], \ldots, [t - \ell + 1, t]\}$, if $x \sim D$, the random variable $Y_R = x^\intercal 1_R$ follows a Gaussian distribution with mean $\mathbb{E}[Y_R] = 0$ and variance*

$$\sigma^2 = 1_R^\intercal \mathbb{C}ov(x, x) 1_R = \begin{cases} \ell & x \sim N(0, I_t) \\ \ell & x \sim N(0, \Sigma_S), \quad R \neq S \\ (1 + \alpha)\ell & x \sim N(0, \Sigma_S), \quad R = S \end{cases}.$$

Moreover, $\mathbb{E}[Y_R^2] = \sigma^2 + \mathbb{E}[Y_R]^2 = \sigma^2$.

We first restate and prove Lemma 8.6.

Lemma 8.6. *For any set $S \in \mathcal{S}$, the distributions $N(0, I_t)$ and $N(0, \Sigma_S)$ are close (in TV distance) to their respective truncations P^0_{trunc} and $P^{1,S}_{\text{trunc}}$:*

$$\left\| P^0_{\text{trunc}} - N(0, I_t) \right\|_{TV} \leq 0.01/(nd/t),$$
$$\left\| P^{1,S}_{\text{trunc}} - N(0, \Sigma_S) \right\|_{TV} \leq 0.01/(nd/t).$$

Proof of Lemma 8.6. We proceed in a similar way as in the proof of Lemma 7.6. For each distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$, Claim E.1 tells us that the TV distance is precisely the probability $\Pr_{x \sim D}[x \notin T]$, where T is the set defined in Equation (23). We split this probability into two terms and bound each separately. In particular, we define the set

$$T' = \left\{ x \in \mathbb{R}^t : |x^\intercal 1_R| \leq \sqrt{2(1 + \alpha)\ell \log(400nd)} \quad \forall R \in \mathcal{S} \right\}$$

of values x whose sums over ℓ-sized blocks are bounded. Note that the variables $Y_R = x^\intercal 1_R$ are mutually independent when $x \sim D$ and that further conditioning on T' preserves independence between those blocks. Note also that

$$\begin{aligned} \Pr_{x \sim D}[x \notin T] &= \Pr_{x \sim D}[x \notin T, x \in T'] + \Pr_{x \sim D}[x \notin T, x \notin T'] \\ &\leq \Pr_{x \sim D | x \in T'}[x \notin T] + \Pr_{x \sim D}[x \notin T']. \end{aligned} \tag{49}$$

The majority of our analysis is devoted to obtaining a bound for the first term. Recalling the definition of T, we have

$$\Pr_{x \sim D | x \in T'}[x \notin T] = \Pr_{x \sim D | x \in T'} \left[\sum_R \exp\left(\frac{\alpha}{2(\alpha + 1)} \cdot \frac{1}{\ell} (x^\intercal 1_R)^2 \right) > (t/\ell)(1 - \alpha)^{-1/2} + \delta \right],$$

where $\delta = d^{\epsilon/2} \sqrt{(t/\ell) \log(400nd)}$. For this term, we will apply a concentration inequality for the sum of independent, bounded random variables. For the second term, we will use a standard tail bound. We begin by bounding the first term. We will make use of the following claim, which bounds the expectation of the key term in our analysis.

Claim F.2. *For any distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$ and any set $R \in \mathcal{S}$,*

$$\mathbb{E}_{x \sim D | x \in T'} \left[\exp\left(\frac{\alpha}{2(\alpha + 1)} \cdot \frac{1}{\ell} (x^\intercal 1_R)^2 \right) \right] \leq (1 - \alpha)^{-1/2}.$$

Proof. We consider the random variable $Y_R = x^\intercal 1_R$, where $x \sim D$. Let $c = \frac{\alpha}{2(\alpha+1)} \cdot \frac{1}{\ell}$. First note that we can rewrite the left hand side of our target expression as follows

$$\begin{aligned}
&\mathbb{E}\left[\exp(cY_R^2) \;\middle|\; |Y_A| \le \sqrt{2(1+\alpha)\ell \log(400nd)} \quad \forall A \in \mathcal{S}\right] \\
&= \mathbb{E}\left[\exp(cY_R^2) \;\middle|\; |Y_R| \le \sqrt{2(1+\alpha)\ell \log(400nd)}\right]. \qquad \text{(by independence of blocks)}
\end{aligned}$$

Next, observe that

$$\begin{aligned}
&\mathbb{E}\left[\exp(cY_R^2)\right] \\
&= \mathbb{E}\left[\exp(cY_R^2) \;\middle|\; |Y_R| > \sqrt{2(1+\alpha)\ell \log(400nd)}\right]\left(\Pr\left[|Y_R| > \sqrt{2(1+\alpha)\ell \log(400nd)}\right]\right) + \\
&\quad \mathbb{E}\left[\exp(cY_R^2) \;\middle|\; |Y_R| \le \sqrt{2(1+\alpha)\ell \log(400nd)}\right]\left(\Pr\left[|Y_R| \le \sqrt{2(1+\alpha)\ell \log(400nd)}\right]\right) \\
&\ge \mathbb{E}\left[\exp(cY_R^2) \;\middle|\; |Y_R| \le \sqrt{2(1+\alpha)\ell \log(400nd)}\right]. \qquad \text{(since } e^{cy^2} \text{ is monotone)}
\end{aligned}$$

Thus, it suffices to find an upper bound for $\mathbb{E}[\exp(cY_R^2)]$. By Claim F.1, we know that Y_R follows a Gaussian distribution. Therefore, by standard properties of the Gaussian distribution we can show that if $c < \frac{1}{2\sigma^2}$, then

$$\begin{aligned}
\mathbb{E}_{Y_R}\left[\exp(cY_R^2)\right] &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \cdot \exp(cy^2)\, dy \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \int_{-\infty}^{\infty} \exp\left(-y^2\left(\frac{1}{2\sigma^2} - c\right)\right) dy \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \sqrt{\frac{\pi}{\frac{1}{2\sigma^2} - c}} \\
&= \sqrt{\frac{1}{1 - 2\sigma^2 c}}.
\end{aligned}$$

Since $c = \frac{\alpha}{2(\alpha+1)} \cdot \frac{1}{\ell}$ and $\alpha < 1$, the required condition on c holds and thus, we have

$$\mathbb{E}_{Y_R}\left[\exp(cY_R^2)\right] = \begin{cases} (1+\alpha)^{1/2} & \sigma^2 = \ell \\ (1-\alpha)^{-1/2} & \sigma^2 = (1+\alpha)\ell. \end{cases} \tag{50}$$

Note that since $\alpha < 1$, we have $(1+\alpha)^{1/2} \le (1-\alpha)^{-1/2}$. Our desired result immediately follows. □

Next, we recall that conditioning on the set T' when x is drawn from a distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$ preserves independence between the ℓ-sized blocks of coordinates of x. Hence, we can apply Hoeffding's inequality to the random variables

$$Z_R = \exp\left(\frac{\alpha}{2(\alpha+1)} \cdot \frac{1}{\ell}(x^\intercal 1_R)^2\right),$$

where x is drawn from the appropriate Gaussian distribution D further truncated on the set T'. The random variable is clearly bounded as we show below:

$$0 \le Z_R \le \exp\left(\frac{\alpha}{2(\alpha+1)} \cdot \frac{1}{\ell}\left(\sqrt{2(1+\alpha)\ell\log(400nd)}\right)^2\right) = (400nd)^{\alpha} \le (400d^{11})^{\alpha}.$$

The final inequality follows from the fact that $n \le d^{10}$. Now, notice that the preceding Claim F.2 implies that for $\delta = d^{\epsilon/2}\sqrt{(t/\ell)\log(400nd)}$,

$$\begin{aligned}
\Pr\left[\sum_R Z_R \ge (t/\ell)(1-\alpha)^{-1/2} + \delta\right] &\le \Pr\left[\sum_R Z_R - \mathbb{E}\left[\sum_R Z_R\right] \ge \delta\right] \\
&\le \exp\left(-\frac{2\left(d^{\epsilon/2}\sqrt{(t/\ell)\log(400nd)}\right)^2}{(t/\ell)(400d^{11})^{2\alpha}}\right) \\
&= \exp\left(-\frac{d^{\epsilon-22\alpha}}{400^{2\alpha}} \cdot 2\log(400nd)\right) \\
&\le (1/(400nd))^2 \qquad \text{(since } \alpha < \epsilon/22\text{, and } d \text{ is sufficiently large)} \\
&\le 0.005/(nd/t).
\end{aligned}$$

The fourth line follows from the fact that $\alpha < \epsilon/22$ and taking d sufficiently large. Thus, we have obtained an upper bound of $0.005/(nd/t)$ for the first term $\Pr_{x\sim D|x\in T'}[x \notin T]$ in (49). Finally, we compute an upper bound on the second term $\Pr_{x\sim D}[x \notin T']$ for each distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$.

Claim F.3. *For each distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$,*

$$\Pr_{x\sim D}[x \notin T'] \le 0.005/(nd/t).$$

Proof. By a union bound over the sets $R \in \{[1, \ell], [\ell+1, 2\ell], \ldots, [t-\ell+1, t]\}$, the left side is at most $\sum_R \Pr_x[|x^\intercal 1_R| \ge \sqrt{2(1+\alpha)\ell\log(400nd)}]$. We again consider the random variable $Y_R = x^\intercal 1_R$ and recall Claim F.1. Hence, it follows that for each distribution $D \in \{N(0, I_t), N(0, \Sigma_S)\}$, we have

$$\begin{aligned}
\sum_R \Pr_{x\sim D}\left[|x^\intercal 1_R| \ge \sqrt{2(1+\alpha)\ell\log(400nd)}\right] &\le (t/\ell) \Pr_{Y\sim N(0,(1+\alpha)\ell)}\left[|Y| \ge \sqrt{2(1+\alpha)\ell\log(400nd)}\right] \\
&\le (t/\ell)\exp\left(-\frac{\left(\sqrt{2(1+\alpha)\ell\log(400nd)}\right)^2}{2(1+\alpha)\ell}\right) \\
&= (t/\ell)\exp(-\log(400nd)) \\
&\le 0.005/(nd(\ell/t)) \\
&\le 0.005/(nd/t).
\end{aligned}$$

□

To conclude, we have bounded the sum of the two terms in (49) by $0.01/(nd/t)$ as desired. □

We now restate and prove Claim 8.7 below.

Claim 8.7. *Let μ_0 and μ_1^S be the probability density functions of P^0_{trunc} and $P^{1,S}_{\text{trunc}}$ respectively. Then, there exists a positive constant C such that*

$$\mathbb{E}_{S\sim\mathcal{S}}[\mu_1^S] \le C\mu_0.$$

Proof of Claim 8.7. Let $f_S(x)$ be the probability density function for $N(0,\Sigma_S)$. We first apply standard matrix identities to the matrix $\Sigma_S = I_d + \alpha vv^\top$, where $v = \frac{1}{\sqrt{\ell}}1_S$, to derive that

$$\begin{aligned}
f_S(x) &= (2\pi)^{-t/2}|\Sigma_S|^{-1/2}\exp\left(-\frac{1}{2}x^\top\Sigma_S^{-1}x\right)\\
&= (2\pi)^{-t/2}|\Sigma_S|^{-1/2}\exp\left(-\frac{1}{2}x^\top\left(I_d - \frac{\alpha vv^\top}{\alpha+1}\right)x\right) && \text{(Sherman-Morrison identity)}\\
&= (2\pi)^{-t/2}(1+\alpha)^{-1/2}\exp\left(-\frac{1}{2}x^\top\left(I_d - \frac{\alpha vv^\top}{\alpha+1}\right)x\right) && \text{(Matrix-determinant lemma)}\\
&= f_0(x)(1+\alpha)^{-1/2}\exp\left(\frac{\alpha}{2(\alpha+1)}(x^\top v)^2\right)\\
&= f_0(x)(1+\alpha)^{-1/2}\exp\left(\frac{\alpha}{2(\alpha+1)}\cdot\frac{1}{\ell}(x^\top 1_S)^2\right).
\end{aligned}$$

For every $x\in T$, we take expectation over $S\in\mathcal{S}$ to get

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{S}}\left[\frac{f_v(x)}{f_0(x)}\right] &= (1+\alpha)^{-1/2}\cdot\frac{1}{(t/\ell)}\sum_S\exp\left(\frac{\alpha}{2(\alpha+1)}\cdot\frac{1}{\ell}(x^\top 1_S)^2\right)\\
&\le (1+\alpha)^{-1/2}\cdot\frac{1}{(t/\ell)}\left((t/\ell)(1-\alpha)^{-1/2} + d^{\epsilon/2}\sqrt{(t/\ell)\log(400nd)}\right) && \text{(since } x\in T\text{)}\\
&= (1+\alpha)^{-1/2}((1-\alpha)^{-1/2}+1). && \text{(since } t\ge \ell d^\epsilon\log(400nd)\text{)}
\end{aligned}$$

Since α is a constant, the above expression is bounded above by a constant, as desired. □

We restate and prove Claim 8.8 below.

Claim 8.8. *Fix a constant $\delta\in(0,1)$ and suppose that $n\ge\log\left(\frac{2}{\delta}\right)\left[\frac{4C_1^2(1+\alpha)^2}{c\alpha^2}\cdot\frac{d}{\ell}\right]$. Then,*

$$\max\left\{\Pr_{D_{\text{null}}}\left[\sum_{j=1}^n\sum_{R\in\mathcal{S}}\left(\sum_{i\in R}x_i^j\right)^2\ge\tau\right], \Pr_{D_{\text{planted}}}\left[\sum_{j=1}^n\sum_{R\in\mathcal{S}}\left(\sum_{i\in R}x_i^j\right)^2\le\tau\right]\right\}\le\delta$$

Proof of Claim 8.8. For each block $R\in\{[1,\ell],[\ell+1,2\ell],\dots,[d-\ell+1,d]\}$ and each sample $j\in[m]$, we define the variable $Y_{R,j} = \sum_{i\in R}x_i^j$. Since the samples are drawn independently and the covariance matrix of the underlying Gaussian distributions are such that that coordinates from different ℓ-sized blocks are independent, the variables $Y_{R,j}$ are all independent. We also recall properties of $Y_{R,j}$ given by Claim F.1.

Our approach is to apply Bernstein's inequality to the random variables $Z_{R,j} = Y_{R,j}^2 - \mathbb{E}[Y_{R,j}^2]$.

Proposition F.4 (Bernstein's inequality)**.** *Let $Z_1, \ldots, Z_N$ be independent, mean-zero, sub-exponential random variables. Then for every $t \geq 0$, we have*

$$\Pr\left[\left|\sum_{i=1}^{N} Z_i\right| \geq t\right] \leq 2\exp\left[-c\min\left(\frac{t^2}{\sum_{i=1}^{n}\|Z_i\|^2}, \frac{t}{\max_i \|Z_i\|}\right)\right],$$

where $c > 0$ is an absolute constant, and $\|X\| = \inf\{K > 0 : \mathbb{E}[\exp(|X|/K)] \leq 2\}$.

It is straightforward to verify that the variables $Z_{R,j}$ are independent and mean-zero. We also remark that $Z_{R,j} \sim \mathbb{V}ar(Y_{S,j})(X-1)$ where X is a chi-squared random variable with one degree of freedom. By standard properties of the chi-squared distribution, it follows that $Z_{R,j}$ is a sub-exponential random variable and $\|Z_{R,j}\| \leq \mathbb{V}ar(Y_{R,j})\|X-1\| = C_1\mathbb{V}ar(Y_{R,j})$, where $C_1 > 1$ is an absolute constant that depends on the the chi-squared distribution.

Finally, we are ready to apply Bernstein's inequality. Let $\tau = nd + n\alpha\ell/2$. The probability that the test fails to detect D_{null} is at most

$$\begin{aligned}
\Pr_{D_{\text{null}}}\left[\sum_j\sum_R Y_{R,j}^2 \geq \tau\right] &\leq \Pr_{D_{\text{null}}}\left[\sum_j\sum_R Z_{R,j} \geq \tau - \sum_j\sum_R \mathbb{E}[Y_{R,j}^2]\right] \\
&= \Pr_{D_{\text{null}}}\left[\sum_j\sum_R Z_{R,j} \geq \tau - n(d/\ell)\ell\right] && \text{(by Claim F.1)} \\
&\leq \Pr_{D_{\text{null}}}\left[\left|\sum_j\sum_R Z_{R,j}\right| \geq n\alpha\ell/2\right].
\end{aligned}$$

Similarly, the probability that the test fails to detect D_{planted} is at most

$$\begin{aligned}
\Pr_{D_{\text{planted}}}\left[\sum_j\sum_R Y_{R,j}^2 < \tau\right] &\leq \Pr_{D_{\text{planted}}}\left[\sum_j\sum_R Z_{R,j} < \tau - \sum_j\sum_R \mathbb{E}[Y_{R,j}^2]\right] \\
&= \Pr_{D_{\text{planted}}}\left[\sum_j\sum_R Z_{R,j} < \tau - n\left((d/\ell - 1)\ell + (1+\alpha)\ell\right)\right] && \text{(by Claim F.1)} \\
&\leq \Pr_{D_{\text{planted}}}\left[\left|\sum_j\sum_R Z_{R,j}\right| \geq n\alpha\ell/2\right].
\end{aligned}$$

In either case, taking $n \geq \log\left(\frac{2}{\delta}\right)\left[\frac{4C_1^2(1+\alpha)^2}{c\alpha^2}\cdot\frac{d}{\ell}\right]$ the failure probability is at most

$$\begin{aligned}
&2\exp\left[-c\min\left(\frac{n^2(\alpha\ell/2)^2}{n(d/\ell)\cdot C_1^2((1+\alpha)\ell)^2}, \frac{n(\alpha\ell/2)}{C_1(1+\alpha)\ell}\right)\right] \\
&= 2\exp\left[-\frac{c\alpha}{2C_1(1+\alpha)}n\min\left(\frac{\alpha\ell}{2C_1 d(1+\alpha)}, 1\right)\right] \\
&\leq 2\exp\left[-\frac{c\alpha}{2C_1(1+\alpha)}n\cdot\left(\frac{\alpha\ell}{2C_1 d(1+\alpha)}\right)\right] && \text{(since } \alpha < 1, \ell \leq d \text{ and } C_1 > 1\text{)}
\end{aligned}$$

$$= 2\exp\left[-\frac{c\alpha^2\ell}{4C_1^2(1+\alpha)^2 d}n\right]$$
$$\leq \delta$$

as desired. □

References

[AAK+07] Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing k-wise and almost k-wise independence. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 496–505, 2007. 1

[Abb18] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018. 1

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010. 2

[AITT00] Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000. 7

[AKS98] 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. 1

[ALPA17] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017. 10

[AMOP08] Alexandr Andoni, Andrew McGregor, Krzysztof Onak, and Rina Panigrahy. Better bounds for frequency moments in random-order streams. *arXiv preprint arXiv:0808.2222*, 2008. 1, 8, 10

[AS15] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. *Advances in neural information processing systems*, 28, 2015. 9

[Ass23] Sepehr Assadi. Recent advances in multi-pass graph streaming lower bounds. *ACM SIGACT News*, 54(3):48–75, 2023. 10

[AV11] Brendan PW Ames and Stephen A Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical programming*, 129(1):69–89, 2011. 2

[AWZ23] Gérard Ben Arous, Alexander S Wein, and Ilias Zadik. Free energy wells and overlap gap property in sparse pca. *Communications on Pure and Applied Mathematics*, 76(10):2410–2473, 2023. 1

[BAR02] YANNICK BARAUD. Non-asymptotic minimax rates of testing in signal detection. *Bernoulli*, 8(5):577–606, 2002. 7

[BB20] Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020. 1

[BBFM12] Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, pages 26–1. JMLR Workshop and Conference Proceedings, 2012. 9

[BBH+21] Matthew S Brennan, Guy Bresler, Sam Hopkins, Jerry Li, and Tselil Schramm. Statistical query algorithms and low degree tests are almost equivalent. In *Conference on Learning Theory*, pages 774–774. PMLR, 2021. 1

[BBS22] Gavin Brown, Mark Bun, and Adam Smith. Strong memory lower bounds for learning natural models. In *Conference on Learning Theory*, pages 4989–5029. PMLR, 2022. 9

[BGL+24] Mark Braverman, Sumegha Garg, Qian Li, Shuo Wang, David P Woodruff, and Jiapeng Zhang. A new information complexity measure for multi-pass streaming with applications. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1781–1792, 2024. 1, 9, 10, 11, 13, 14, 18, 19, 20, 24, 42, 43, 45

[BGM+16] 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. 1, 8, 10, 11, 25, 26, 27, 28, 49

[BGW20] Mark Braverman, Sumegha Garg, and David P Woodruff. The coin problem with applications to data streams. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 318–329. IEEE, 2020. 1, 13

[BKS23] Rares-Darius Buhai, Pravesh K Kothari, and David Steurer. Algorithms approaching the threshold for semi-random planted clique. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1918–1926, 2023. 2

[BKV12] Bahman Bahmani, Ravi Kumar, and Sergei Vassilvitskii. Densest subgraph in streaming and mapreduce. *Proceedings of the VLDB Endowment*, 5(5), 2012. 7

[BLS+18] Vladimir Braverman, Zaoxing Liu, Tejasvam Singh, NV Vinodchandran, and Lin F Yang. New bounds for the clique-gap problem using graph decomposition theory. *Algorithmica*, 80:652–667, 2018. 5, 6

[BM15] Rémi Bardenet and Odalric-Ambrym Maillard. Concentration inequalities for sampling without replacement. 2015. 32, 35

[BMR20] Jean Barbier, Nicolas Macris, and Cynthia Rush. All-or-nothing statistical and computational phase transitions in sparse spiked matrix estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 14915–14926, 2020. 1

[BR13] Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Conference on learning theory*, pages 1046–1066. PMLR, 2013. 1, 9

[BYJKS04] 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. 12, 27, 49

[CCM08] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 641–650, 2008. 8, 10

[CCT17] O Collier, L Comminges, and AB Tsybakov. Minimax estimation of linear and quadratic functionals on sparsity classes. *Annals of Statistics*, 45(3):923–958, 2017. 7

[CCTV18] Olivier Collier, Laëtitia Comminges, Alexandre B Tsybakov, and Nicolas Verzelen. Optimal adaptive estimation of linear functionals under sparsity. *The Annals of Statistics*, 46(6A):3130–3150, 2018. 7

[CDK18] Graham Cormode, Jacques Dark, and Christian Konrad. Approximating the caro-wei bound for independent sets in graph streams. In *International Symposium on Combinatorial Optimization*, pages 101–114. Springer, 2018. 6

[CDK19] Graham Cormode, Jacques Dark, and Christian Konrad. Independent sets in vertex-arrival streams. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, pages 1–14. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Germany, 2019. 6

[CG19] Lijie Chen and Ofer Grossman. Broadcast congested clique: Planted cliques and pseudorandom generators. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 248–255, 2019. 5, 13

[CMVW16] Michael Crouch, Andrew McGregor, Gregory Valiant, and David P Woodruff. Stochastic streams: Sample complexity vs. space complexity. In *24th Annual European Symposium on Algorithms (ESA 2016)*, pages 32–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016. 1

[CX16] Yudong Chen and Jiaming Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *Journal of Machine Learning Research*, 17(27):1–57, 2016. 1

[DF80] Persi Diaconis and David Freedman. Finite exchangeable sequences. *The Annals of Probability*, pages 745–764, 1980. 57

[DH24] Rishabh Dudeja and Daniel Hsu. Statistical-computational trade-offs in tensor pca and related problems via communication complexity. *The Annals of Statistics*, 52(1):131–156, 2024. 1

[DJ04] David DONOHO and JIASHUN JIN. Higher criticism for detecting sparse heterogeneous mixtures. *Annals of statistics*, 32(3):962–994, 2004. 7

[DJ08] David Donoho and Jiashun Jin. Higher criticism thresholding: Optimal feature selection when useful features are rare and weak. *Proceedings of the National Academy of Sciences*, 105(39):14790–14795, 2008. 7

[DJ15] David Donoho and Jiashun Jin. Higher criticism for large-scale inference, especially for rare and weak effects. *Statistical Science*, 30(1):1–25, 2015. 7

[DJW13] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013. 9

[dKNS20] Tommaso d'Orsi, Pravesh K Kothari, Gleb Novikov, and David Steurer. Sparse pca: algorithms, adversarial perturbations and certificates. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 553–564. IEEE, 2020. 1

[DKS19] Yuval Dagan, Gil Kur, and Ohad Shamir. Space lower bounds for linear prediction in the streaming model. In *Conference on Learning Theory*, pages 929–954. PMLR, 2019. 9

[DKWB24] Yunzi Ding, Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Subexponential-time algorithms for sparse pca. *Foundations of Computational Mathematics*, 24(3):865–914, 2024. 1

[DR19] John Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. In *Conference on Learning Theory*, pages 1161–1191. PMLR, 2019. 11

[DS18] Yuval Dagan and Ohad Shamir. Detecting correlations with little memory and communication. In *Conference On Learning Theory*, pages 1145–1198. PMLR, 2018. 9

[FGR+17] 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. 2, 10

[FK98] Uriel Feige and Joe Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 674–683. IEEE, 1998. 5

[FK00] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms*, 16(2):195–208, 2000. 1, 5

[FP16] Laura Florescu and Will Perkins. Spectral thresholds in the bipartite stochastic block model. In *Conference on Learning Theory*, pages 943–959. PMLR, 2016. 2

[GM07] Sudipto Guha and Andrew McGregor. Space-efficient sampling. In *Artificial Intelligence and Statistics*, pages 171–178. PMLR, 2007. 1

[GMN14] Ankit Garg, Tengyu Ma, and Huy L Nguyen. On communication cost of distributed statistical estimation and dimensionality. *Advances in Neural Information Processing Systems*, 27, 2014. 8

[GRT18] Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 990–1002, 2018. 10

[HJ10] Peter Hall and Jiashun Jin. Innovated higher criticism for detecting sparse signals in correlated noise. *The Annals of Statistics*, pages 1686–1732, 2010. 7

[HKP+17] Samuel B Hopkins, Pravesh K Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 720–731. IEEE, 2017. 1

[HSSW12] 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. 5, 6

[Ing96] Yuri I Ingster. On some problems of hypothesis testing leading to infinitely divisible distributions. 1996. 7

[IS03] Yu. I. Ingster and I. A. Suslina. *Nonparametric goodness-of-fit testing under Gaussian models*, volume 169 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 2003. 7

[Jay09] TS Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 562–573. Springer, 2009. 28

[Jer92] Mark Jerrum. Large cliques elude the metropolis process. *Random Structures & Algorithms*, 3(4):347–359, 1992. 1

[JJK+16] Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm. In *Conference on learning theory*, pages 1147–1164. PMLR, 2016. 8

[JL09] Iain M Johnstone and Arthur Yu Lu. Sparse principal components analysis. *arXiv preprint arXiv:0901.4392*, 2009. 9

[JW07] Leah Jager and Jon A Wellner. Goodness-of-fit tests via phi-divergences. *The Annals of Statistics*, 35(5), 2007. 7

[Kap21] Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1874–1893. SIAM, 2021. 6

[KKS14] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 734–751. SIAM, 2014. 1

[KLP22] Akash Kumar, Anand Louis, and Rameesh Paul. Exact recovery algorithm for planted bipartite graph in semi-random graphs. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, pages 84–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022. 2

[KMM12] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 231–242. Springer, 2012. 1

[KMPV19] John Kallaugher, Andrew McGregor, Eric Price, and Sofya Vorotnikova. The complexity of counting cycles in the adjacency list streaming model. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 119–133, 2019. 6

[KS13] Bernd Klaus and Korbinian Strimmer. Signal identification for rare and weak features: higher criticism or false discovery rates? *Biostatistics*, 14(1):129–143, 2013. 7

[KS24] Syamantak Kumar and Purnamrita Sarkar. Oja's algorithm for streaming sparse pca. *Advances in Neural Information Processing Systems*, 37:74528–74578, 2024. 8

[Kuč95] Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995. 1, 5

[KVV90] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990. 6

[LKZ15] Thibault Lesieur, Florent Krzakala, and Lenka Zdeborová. Phase transitions in sparse pca. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1635–1639. IEEE, 2015. 1

[LKZ17] Thibault Lesieur, Florent Krzakala, and Lenka Zdeborová. Constrained low-rank matrix estimation: Phase transitions, approximate message passing and applications. *Journal of Statistical Mechanics: Theory and Experiment*, 2017(7):073403, 2017. 1

[LMFB24] Tommaso Lanciano, Atsushi Miyauchi, Adriano Fazzone, and Francesco Bonchi. A survey on the densest subgraph problem and its variants. *ACM Computing Surveys*, 56(8):1–40, 2024. 7, 31

[LWZ23] Tianyuan Lu, Lei Wang, and Xiaoyong Zhao. Review of anomaly detection algorithms for data streams. *Applied Sciences*, 13(10):6353, 2023. 10

[LWZ25] Qian Li, Shuo Wang, and Jiapeng Zhang. Multi-pass memory lower bounds for learning problems. In *Proceedings of Thirty Eighth Conference on Learning Theory*. PMLR, 2025. 9

[LZ23] Shachar Lovett and Jiapeng Zhang. Streaming lower bounds and asymmetric set-disjointness. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 871–882. IEEE, 2023. 8, 10

[Mar21a] Jay Mardia. Is the space complexity of planted clique recovery the same as that of detection? In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, pages 34–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021. 5

[Mar21b] Jay Mardia. Logspace reducibility from secret leakage planted clique. *arXiv preprint arXiv:2107.11886*, 2021. 9

[MBPS10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1), 2010. 8

[McG14] Andrew McGregor. Graph stream algorithms: a survey. *ACM SIGMOD Record*, 43(1):9–20, 2014. 10

[MCJ13] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming pca. *Advances in neural information processing systems*, 26, 2013. 8

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537. IEEE, 2001. 1

[MMA16] Emaad Manzoor, Sadegh M Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1035–1044, 2016. 10

[MSSV24] Annie Marsden, Vatsal Sharan, Aaron Sidford, and Gregory Valiant. Efficient convex optimization requires superlinear memory. *Journal of the ACM*, 71(6):1–37, 2024. 1

[MW15a] Tengyu Ma and Avi Wigderson. Sum-of-squares lower bounds for sparse PCA. *Advances in Neural Information Processing Systems*, 28, 2015. 1, 9

[MW15b] Zongming Ma and Yihong Wu. Computational barriers in minimax submatrix detection. *The Annals of Statistics*, 43(3):1089–1116, 2015. 1

[Rag16] Maxim Raginsky. Strong data processing inequalities and *phi*-sobolev inequalities for discrete channels. *IEEE Transactions on Information Theory*, 62(6):3355–3389, 2016. 11

[Raz18] Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *Journal of the ACM (JACM)*, 66(1):1–18, 2018. 1, 9

[RWYZ21] Cyrus Rashtchian, David Woodruff, Peng Ye, and Hanlin Zhu. Average-case communication complexity of statistical problems. In *Conference on Learning Theory*, pages 3859–3886. PMLR, 2021. 5

[SD15] Jacob Steinhardt and John Duchi. Minimax rates for memory-bounded sparse linear regression. In *Conference on Learning Theory*, pages 1564–1587. PMLR, 2015. 1, 9

[SGW18] Vatsal Sharan, Parikshit Gopalan, and Udi Wieder. Efficient anomaly detection via matrix sketching. *Advances in neural information processing systems*, 31, 2018. 10

[Sha14] Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. *Advances in Neural Information Processing Systems*, 27, 2014. 1, 9

[SSV19] Vatsal Sharan, Aaron Sidford, and Gregory Valiant. Memory-sample tradeoffs for linear regression with small error. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 890–901, 2019. 1

[SVW16] Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Conference on Learning Theory*, pages 1490–1516. PMLR, 2016. 1

[Tre17] Luca Trevisan. U.C. Berkeley — CS294: Beyond Worst-Case Analysis: Lecture 1. https://lucatrevisan.github.io/teaching/bwca17/lectures/lecture01.pdf, 2017. 35

[TTL11] Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. Fast anomaly detection for streaming data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1511–1516. AAAI Press, 2011. 10

[WBSS21] Blake E Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory*, pages 4386–4437. PMLR, 2021. 9

[WL16] Chuang Wang and Yue M Lu. Online learning for sparse pca in high dimensions: Exact dynamics and phase transitions. In *2016 IEEE Information Theory Workshop (ITW)*, pages 186–190. IEEE, 2016. 8

[YX15] Wenzhuo Yang and Huan Xu. Streaming sparse principal component analysis. In *International Conference on Machine Learning*, pages 494–503. PMLR, 2015. 8

[ZDJW13] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. *Advances in Neural Information Processing Systems*, 26, 2013. 8

[ZHT06] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006. 1, 8, 9

[ZX18] Hui Zou and Lingzhou Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018. 8