Electronic Colloquium on Computational Complexity, Report No. 72 (2019)

Broadcast Congested Clique: Planted Cliques and Pseudorandom Generators

Lijie Chen* MIT

Ofer Grossman† MIT

Abstract

Consider the multiparty communication complexity model where there are n processors, each receiving as input a row of an $n \times n$ matrix M with entries in $\{0,1\}$, and in each round each party can broadcast a single bit to all other parties (this is known as the BCAST(1) model). There are many lower bounds known for the number of rounds necessary for certain problems in this model, but they are all worst case lower bounds which apply only for very specifically constructed input distributions. We develop a framework for showing lower bounds in this setting for more natural input distributions, and apply the framework to show:

- A lower bound for finding planted cliques in random inputs (i.e., each entry of the matrix is random, except there is a random subset $a_1, \ldots, a_k \in [n]$ where $M_{a_i a_j} = 1$ for all i and j). Specifically, we show that if $k = O(n^{1/4-\varepsilon})$, this problem requires a number of rounds polynomial in n.
- A pseudo-random generator which fools the BCAST(1) model. That is, we show a distribution which is efficiently samplable using few random bits, and which is indistinguishable from uniform by a low-round BCAST(1) protocol. This allows us to show that every $t = \Omega(\log n)$ round randomized algorithm in which each processor uses up to n random bits can be efficiently transformed into an $O(t)$-round randomized algorithm in which each processor uses only up to $O(t)$ random bits, while maintaining a high success probability.

As a corollary of the pseudo-random generator, we also prove the first average case lower bound for the model (specifically, for the problem of determining whether the input matrix is full rank), as well as an average-case time hierarchy.

1 Introduction

In the Broadcast Congested Clique model (BCAST(1)), there are n processors numbered 1 through n, (each with unlimited local computation power), which each receive as input a row of an $n \times n$ bit matrix M. Computation proceeds in rounds, and in each round each processor broadcasts a single bit to all other processors (within a single round, a processor must broadcast the same bit to all other processors), and the goal is to compute some function of the input.

*Supported by NSF CCF-1741615 (CAREER: Common Links in Algorithms and Complexity). This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

†Supported by the Fannie and John Hertz Foundation Fellowship, an NSF Graduate Research Fellowship, NSF CNS-1413920, DARPA 491512803, Sloan 996698, and MIT/IBM W1771646.

ISSN 1433-8092

This model is actively studied, for example in [BARR15, dERRU16, CKK+15, GHM18, DKO14, Gal16, NY19, HP15, CPS17, BMRT18, JN17b, MT16, JN17a] to list a few. In general, lower bounds for the model take the following approach: a specific carefully chosen family of input matrices is constructed, and the processors are partitioned into two parts (or, in some rare cases, three parts), and a reduction from classical two-party (or three-party) communication complexity is used.

A main downside of this approach is that it merely proves worst case lower bounds. For all we know, for more natural input distributions (instead of the artificially constructed input distributions used for these communication complexity reductions), many problems which are worst-case hard may become easy. One of our main goals in this paper is to develop techniques that prove lower bounds for more natural input distributions.

Our results include a lower bound for the planted clique problem. Additionally, we show how to use our techniques to construct a pseudo-random generator that fools the Broadcast Congested Clique model. This is the first pseudo-random generator which fools a distributed message passing model. As simple corollaries, we obtain an average-case lower bound for the model, and an average-case time hierarchy for the model.

A Note on the Broadcast Congested Clique: In the study of the Broadcast Congested Clique model, it is more common to view the input as a graph (instead of a matrix as we define it here), where each vertex of the graph corresponds to a processor, and each processor gets a list of the edges incident to it. This definition is similar to ours – instead of the graph one can view the input as the adjacency matrix of the graph, where each processor receives one row of the matrix. Note that there is a difference between our definition and that classic definition for undirected graphs in that an adjacency matrix is always symmetric, whereas our input matrix need not be symmetric.

Additionally, it is common to consider the case where instead of sending a single bit in each round, instead each processor can send up to $O(\log n)$ bits per round. When this is the case, we refer to the model as $\mathsf{BCAST}(\log n)$.

Another way to view a BCAST(1) protocol is as a protocol for number-in-hand communication complexity in the blackboard model with the additional constraint that in each round every processor must write exactly one bit on the blackboard.

1.1 Our Approach

How to deal with distributions Consider the (directed) planted clique problem[1]. In this problem, the input is a directed graph which is either a random graph (each directed edge is included with probability $1/2$), or a random graph where k of the vertices are chosen at random and all edges within this set of k vertices are included (these k vertices are called the planted clique). The goal is to distinguish between these two distributions (or, one can consider the search version of the problem, in which a graph with a planted clique is given, and the goal is to find the clique). For now, one can think of k as approximately $n^{1/4}$.

When trying to prove a lower bound for the problem, one's first approach may be to try reducing to two-party communication complexity: split the vertices into two parts, and argue that a lot of information must pass between the parts. This approach will not work for the problem, since no matter how the graph is split up, at least one of the parts will be able to detect the presence of the clique, since at least one of the parts must have many of the clique's vertices.

One's next approach may be the following: show that for any Congested Clique algorithm, after t rounds, the distribution of transcripts of the Congested Clique algorithm (the "transcript" is the history of

[1]Note that in the broadcast congested clique model, as opposed to the CONGEST or unicast congested clique model, it is not possible to reduce from directed to undirected in one round.

the algorithm; that is, the "transcript" is a list of all messages sent so far as well as who sent which message and when the message was sent) if the input were uniform is close to the distribution of transcripts if the input had a planted clique. To prove this, one may take an inductive approach: show that if the transcripts were similar for $t-1$ rounds, the next round can distinguish between the distributions with only low probability. In this round, all vertices are broadcasting, so one may try to use another approach: show that each vertex reveals little information about whether the graph has a planted clique or not, and therefore, the whole round reveals little information about whether there is a clique. This way, we would only need to analyze a single broadcast at a time, a much simpler task than considering a whole round at once.

There is an issue, however. The problem is that the inputs of different nodes may not be independent. Therefore, it is possible that while each processor's broadcast on its own will not reveal substantial information about whether the graph contains a clique, when we consider many processors' broadcasted bits the information revealed may be more than the sum of the information of the individual broadcasts.

To get around this, we instead split the planted clique distributions into a sum of many distributions, such that for each of these distributions, all processors' inputs are independent. Specifically, we can write the planted clique distribution as a sum over all possible cliques C of a random graph with a clique planted at C. Notice that after fixing C, each vertex's input is independent of all other vertices' inputs. Now, when considering whether an algorithm distinguishes between a random graph with a clique at C and a truly random graph, we can consider each node's output on its own, instead of trying to deal with all nodes at the same time. This is one of our main high level ideas: splitting the distribution into many distributions where in each one, different vertices' inputs are independent. This greatly simplifies the analysis by letting us avoid having to deal with many messages at once, and is what makes proving the lower bounds possible.

Statistical Inequalities The idea of partitioning a distribution into distributions with independent vertex inputs makes proving the lower bounds possible, but there is still lots of technical work to do. Specifically, we now have a bunch of distributions, and we need to show that any algorithm can distinguish only few of those distributions from uniform. At the high level, the idea to do this is to show that for any algorithm, for almost all cliques, when a vertex broadcasts a bit, the probability of that bit being broadcast with the clique vs without the clique is similar. This is basically a problem about Boolean functions: if we let f be the function which takes in the node's input, and outputs what the node will broadcast, we wish to show that for almost all cliques C, when f's input is uniform, the output distribution is similar to the distribution when f's input is chosen with a planted clique at C. The inequalities we show are at the high level similar to this, but many more technical issues arise. For example, since we have to prove a multi-round lower bound, we have to condition on what a node broadcasted in previous rounds, so instead of proving the statistical inequalities for total functions, we instead have to prove the inequalities for functions defined on only part of $\{0,1\}^n$. These inequalities are not true for all partial functions, but we manage to prove that they are true for all functions which are defined on a large enough subset. So then, there is another challenge of proving that with high probability, after conditioning on a transcript, the set of possible inputs to a node is large.

1.2 Our Results

Lower bounds for the planted clique problem One of the problems we consider is the (directed) planted clique problem. In this problem, the input is a directed graph which is either a random graph (each directed edge is included with probability $1/2$), or a random graph where k of the vertices are chosen and all edges within this set of k vertices are added. The goal is to distinguish between these two distributions (or, one can consider the search version of the problem, in which a graph with a planted clique is given, and the goal is to find the clique). Because a random graph contains cliques of size $\Theta(\log n)$, the problem makes sense for larger values of k. Once k goes substantially above $\sqrt{n}$, it is possible to find the clique by considering the vertices with highest degree. Hence, the interesting values of k are between approximately $\log n$ and $\sqrt{n}$.

In the classical (non-distributed) setting, the planted clique problem is very well studied. There exists a spectral algorithm solving the problem when $k = O(\sqrt{n})$, and it remains a major open problem in complexity theory to understand whether the problem is hard when k is smaller.

It has been shown that proving lower bounds in the unicast CONGESTED-CLIQUE[2] would imply some strong circuit lower bounds [DKO14], so finding lower bounds for the problem in this setting is quite a challenge. Finding upper bounds for planted clique in the unicast CONGESTED-CLIQUE model is an interesting problem, but it seems difficult – maybe impossible – to improve upon simple sampling-based algorithms for the problem. The next model one can look at is the broadcast CONGESTED-CLIQUE model, which is the model we consider. Specifically, we prove that for cliques of size $O(n^{1/4-\varepsilon})$, the planted clique problem requires polynomially many rounds:

Theorem 1.1 (Planted Clique lower bound)**.** *When $k = n^{1/4-\varepsilon}$ for a constant ε, no $n^{o(1)}$ round $\mathsf{BCAST}(1)$ protocol Π can distinguish between $\mathcal{A}_{\mathsf{rand}}$ and $\mathcal{A}_k$ with advantage*[3] $\Omega(1)$.

Pseudo-random generators for distributed computation: Historically, pseudo-random generators were used to fool adversaries modeled as Turing Machines or Circuits. One of our main contributions is constructing the first pseudo-random generator which fools a distributed message passing setting. That is, we show how within the BCAST(1) Congested Clique model, the processors can each sample a small random seed, and end with each processor having a longer string than it started with, such that these longer strings *look random* to the system. That is, no low-round BCAST(1) protocol can distinguish between these pseudo-random strings and truly uniformly random strings within few rounds (except with some low probability). Thus, any algorithm can use a pseudo-random string instead of true randomness, thereby saving random bits.

Pseudo-random generators have been used to derandomize specific problems in message passing models, for example in [PY18]. Our construction is the first pseudo-random generator which fools *all* low-round algorithms in the BCASTmodel (as opposed to just fooling a specific algorithm).

Definition 1.2 (BCAST(1) pseudo-random generator)**.** A (k, m, n, ℓ) BCAST(1) pseudo-random generator (PRG) is an n-processor BCAST(1) protocol Π such that:

- At the beginning every processor independently gets k private uniform and independent random bits as input.
- After participating in protocol Π, each processor outputs m bits (these bits are not broadcasted: they are the node's pseudo-random bits).
- The joint distribution of all processors' output bits cannot be distinguished (with better than $\frac{1}{n}$ probability[4]) from a truly uniform random distribution by any ℓ round BCAST(1) protocol. Specifically, the statistical distance between the distribution of the protocol's transcript when using a pseudo-random generator and the distribution of the transcript when using true randomness is small.

Theorem 1.3. *For all $m = O(n)$ and $k = \Omega(\log n)$, there exists an $(O(k), m, n, \Theta(k))$ BCAST(1) PRG that can be constructed within $O(k)$ rounds. In particular, the PRG works as follows*

[2] In the unicast model, in each round each processor can send one bit to each other processor without the requirement that the same bit is broadcast to all other processors; that is, within a single round a processor may choose to send the message "1" to some processors, and "0" to others.

[3] An algorithm distinguishing between two distributions D_1 and D_2 with advantage ε is an algorithm A which, when given a random sample s which with probability $1/2$ is drawn from D_1 and probability $1/2$ is drawn from D_2, then A can successfully guess from which distribution s was drawn with probability $1/2 + \varepsilon$.

[4] All of our constructions in the paper can achieve arbitrarily low inverse polynomial probability.

- *Each processor gets $k + k \cdot \frac{(m-k)}{n} = O(k)$ private random bits.*
- *Then in $O\left(\frac{m-k}{n} \cdot k\right) = O(k)$ rounds, all processors broadcast their last $k \cdot \frac{(m-k)}{n}$ random bits. And they use that to construct a random matrix $M \in \{0,1\}^{k \times (m-k)}$.*
- *Each processor's output is simply the concatenation of its first k random bits x and $x^T M$.*

That is, with k random bits per processor as a seed, for every constant c, within $O(k)$ rounds we can turn them into cn pseudo-random bits per processor which require $\Omega(k)$ rounds to be distinguished from random.

So, for example, within $O(\log^2(n))$ round (In the $\mathsf{BCAST}(\log n)$ model, $O(\log n)$ rounds would suffice), one can construct a pseudo-random generator which is indistinguishable from random for any $\log^2 n$-round algorithm, except with low probability (the seed size for each processor would be $O(\log^2 n)$, while the size of the pseudo-random string is $\Theta(n)$).

The PRG is very simple to describe (we describe it here with seed size $2k$): first each processor shares k random bits to create k public random elements from $\mathbb{F}_2^n$ (i.e., k random n-bit vectors). Then, each processor uses its remaining k random bits to pick a random linear combination of those vectors (which requires k bits to sample), and the result of this linear combination is the node's pseudo-random bits. So, essentially the pseudo-random generator is a distribution of low-rank matrices (which is very close to the uniform distribution matrices of rank up to k). Although the description and construction of the PRG are simple, and the seed size is tight up to a constant factor, the analysis is quite technical and involves new techniques. We remark that the pseudo-random generator has the additional nice property that it is computationally cheap; the only operations done by the processors is computing dot products of vectors over $\mathbb{F}_2$.

Efficiently saving random bits It is possible to show that in the broadcast congested clique there is a randomized-deterministic separation (by reductions from two-player communication complexity for equality). That is, there are certain problems with faster randomized algorithms than the best possible deterministic algorithms. So, there is no hope for a general derandomization theorem. However, one can ask: in general, what is the fewest number of random bits needed to efficiently solve problems in the Broadcast Congested Clique model? Using a technique of Newman [New91] from communication complexity, it is possible to show that for any protocol with output size k per processor, $O(k)$ random bits per processor is enough (see Appendix A). The main downside of Newman's technique is that it is computationally inefficient: it holds in the case where all the processors have unbounded computational power, and is not a practical tool for saving random bits.

In this work, we ask the following question: In the Broadcast Congested Clique with *computationally bounded (polynomial time)* processors, how many random bits are needed to perform general randomized computation? We can use our pseudo-random generator to show that every k-round algorithm where $k = \Omega(\log n)$ in which every processor uses up to $O(n)$ random bits can be transformed into an $O(k)$-round algorithm in which every processor uses up to k random bits (that is, we can show that each processor needs to use at most 1 random bit per round, while only increasing the run-time by a constant factor). Furthermore, this transformation is efficient. That is, if in the original algorithm all processors work in polynomial time, they also work in polynomial time in the new algorithm (in fact, there is only an *additive* overhead of $O(kn)$ computation time for each processor. This overhead is the time required to compute the pseudo-random bits from the seed).

Stated in the setting of $\mathsf{BCAST}(\log n)$, where messages are of size $O(\log n)$ instead of 1, we show that every k round randomized algorithm using up to n random bits per processor can be transformed into an $O(k)$ round algorithm using up to $O(k \log n)$ random bits per processor.

First BCAST(1) Average Case Lower Bound and Hierarchy: As a simple corollary of our PRG construction, we prove a $\mathsf{BCAST}(1)$ average case lower bound, which is the first average case lower bound proven in the model. Specifically, we show that when each processor receives a row of a sample from a certain close-to-uniform distribution of $n \times n$ matrices of rank $n-1$, this cannot be distinguished from each processor receiving n uniformly random bits. We show that this implies that determining whether an input has rank n or not is hard (it takes $\Omega(n)$ rounds), even when the input is chosen uniformly at random:

Theorem 1.4. *Let n be a large enough integer and $F_{\text{full-rank}} : \{0,1\}^{n\times n} \to \{0,1\}$ be the indicator function which indicates whether the given matrix has full rank. Suppose there are n processors, and the i-th processor is given the i-th row of the input matrix. For all $n/20$-round BCAST(1) protocols and all processors i in it, i cannot compute F correctly with probability better than 0.99 when the input is a uniformly chosen random matrix from $\{0,1\}^{n\times n}$.*

We also obtain an average-case time hierarchy theorem for the model:

Theorem 1.5. *For any $\omega(\log n) \le k \le n$, there is a function F such that a k-round $\mathsf{BCAST}(1)$ protocol can compute exactly, while any $k/20$-round $\mathsf{BCAST}(1)$ protocols cannot compute F correctly with probability 0.99 over the uniform distribution.*

1.3 Toy Example: One Round Lower Bound for Planted Clique

As a toy example to illustrate our proof framework, in the following we prove that the planted clique problem is hard for one-round $\mathsf{BCAST}(1)$ protocols when $k = o(n^{1/4})$. We begin with some notations.

Notations. Let $\mathcal{U}_m$ denote the uniform distribution on $\{0,1\}^m$. For a function $f : \{0,1\}^* \to \{0,1\}^*$ and a distribution $\mathcal{D}$ on $\{0,1\}^*$, we use $f(\mathcal{D})$ to denote the distribution of the output of f when the input is drawn from $\mathcal{D}$. For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we use $\|\mathcal{D}_1 - \mathcal{D}_2\| = \frac{1}{2}\sum_{x\in\{0,1\}^*} |\mathcal{D}_1(x) - \mathcal{D}_2(x)|$ to denote their statistical distance (where $\mathcal{D}(x)$ is the probability that a sample from $\mathcal{D}$ equals x).

Let $\mathcal{A}^n_{\mathsf{rand}}$ be the distribution on $\{0,1\}^{n\times n}$ such that for a sample A from $\mathcal{A}^n_{\mathsf{rand}}$, for all $i \neq j$, $A_{i,j}$ is an independent uniform random bit in $\{0,1\}$, and $A_{i,i}$ is always 0 for all i. Let C be a subset of $[n]$. We use $\mathcal{A}^n_C$ to denote the conditional distribution of $\mathcal{A}^n_{\mathsf{rand}}$ on the event that for all $i, j \in C$ and $i \neq j$, $A_{i,j} = 1$ (that is, C is a clique). We also use $\mathcal{A}^n_k$ to be the mixed distribution of $\mathcal{A}^n_C$'s when C is a uniform random subset of $[n]$ of size k. When the meaning is clear, we often drop the superscripts of the aforementioned distributions for simplicity.

So, to summarize, $\mathcal{A}^n_{\mathsf{rand}}$ is the uniform distribution over a random directed graph, $\mathcal{A}^n_C$ is the distribution where the vertices of C are in a clique, and the rest of the edges are uniformly random, and $\mathcal{A}^n_k$ is the distribution where a random k vertices are chosen to be a clique, and the rest of the edges are chosen uniformly at random.

By Yao's principle [Yao77], we can assume all processors are deterministic as we are trying to prove a lower bound for distinguishing two input distributions. Processor i can then be defined by a function $f_i : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}$, such that $f_i(z,p)$ is the bit that processor i outputs when it gets the input z and transcript p.[5] We use $f_i^{|p}$ to denote the function $f_i(\cdot, p)$ for simplicity. If transcript p is incompatible with processor i having input z, then we set $f_i(z,p)$ arbitrarily.

Given a $\mathsf{BCAST}(1)$ protocol Π and an input distribution $\mathcal{D}$, we use $\mathcal{P}(\Pi, \mathcal{D})$ to denote the distribution of the transcripts of the protocol Π running on a input drawn from $\mathcal{D}$ (that is, given a matrix A which is

[5] In a zero-round protocol, the processor's f_i does not take in an input p, since there is no transcript yet, just an input. However, in our proof, we assume that the processors broadcast their messages sequentially (that is, first the first processor speaks, then the second, and so on). In this stronger model, all but one of the processors do see a transcript before they broadcast their first bit.

drawn from the distribution $\mathcal{D}$, the processor i gets the i-th row of A, and all processors act according to the protocol Π).

In this section we prove the following theorem.

Theorem 1.6. *Let n be the number of processors and k be an integer. For any one round BCAST(1) protocol Π, we have*

$$\|\mathcal{P}(\Pi, \mathcal{A}_{\mathsf{rand}}) - \mathcal{P}(\Pi, \mathcal{A}_k)\| \leq O\left(\frac{k^2}{\sqrt{n}}\right).$$

That is, for any one-round protocol, the distribution of transcripts when run on a uniformly random input is statistically close to the distribution on an input with a planted clique. As a simple corollary, we immediately have:

Corollary 1.7. *When $k = o(n^{1/4})$, no one-round BCAST(1) protocol Π can distinguish between $\mathcal{A}_{\mathsf{rand}}$ and $\mathcal{A}_k$ with advantage $\Omega(1)$ (that is, any protocol which accepts on $\mathcal{A}_{\mathsf{rand}}$ with probability p, must accept on $\mathcal{A}_k$ with probability $p \pm o(1)$).*

That is, there is no way to solve the planted clique problem for cliques of size $o(n^{1/4})$ within one round of the Broadcast Congested Clique.

Let $\mathcal{S}_k^T$ be the uniform distribution on all size-k subsets of T. To prove Theorem 1.6, we need the following technical lemma, whose proof is deferred to the end of this section. In this lemma $f(x)$ represents the bit broadcasted by a processor, and x represents the input to the processor. The lemma states that when picking a random clique C of size k, then any function f behaves similarly when the input is sampled from the uniform distribution vs. when the input is sampled from the uniform distribution with a clique planted on C. Basically, this means that for almost all possible cliques, f does not substantially help distinguish between the clique existing, or the input being uniform.

Lemma 1.8. *Let n, k be integers such that $k \leq n^{1/4}$, and $\mathcal{U}_n^C$ be the uniform distribution on $\{x : x \in \{0,1\}^n, x_i = 1$ for all $i \in C\}$. For all function $f : \{0,1\}^n \to \{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{C \sim \mathcal{S}_k^{[n]}} \left[\left\| f(\mathcal{U}_n) - f(\mathcal{U}_n^C)\right\|\right] \leq O\left(\frac{k}{\sqrt{n}}\right).$$

We also need the following lemma to bound the increase of the statistical distance when a processor speaks, The proof can be found in the preliminaries (Section 2).

Lemma 1.9. *Let X and Y be two sets, $\mathcal{D}$ and $\mathcal{D}'$ be two distributions on $X \times Y$. Let $\mathcal{D}_{|X}$ and $\mathcal{D}'_{|X}$ be the respective marginal distribution of $\mathcal{D}$ and $\mathcal{D}'$ on set X. For $a \in X$, we use $\mathcal{D}_{X=a}$ and $\mathcal{D}'_{X=a}$ to denote the respective conditional distribution of $\mathcal{D}$ and $\mathcal{D}'$ on Y conditioning on $X = a$.[6] We have*

$$\|\mathcal{D} - \mathcal{D}'\| \leq \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\| + \mathop{\mathbb{E}}_{a \sim \mathcal{D}_{|X}} \left[\|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|\right].$$

Now we are ready to prove Theorem 1.6.

Proof of Theorem 1.6. Instead of viewing the algorithm as a single round algorithm, we will prove a slightly stronger lower bound. Consider the model where we have n turns. On the t^{th} turn, processor t gets to send a single bit. This model is stronger than one round of the BCAST(1) model, since it allows the later processors to condition their outputs on earlier the processors' messages. Hence, our lower bound implies a lower bound for the BCAST(1) model as well.

[6]For simplicity, we let $\mathcal{D}_{X=a}$ be the uniform distribution on Y if $\Pr_{(x,y)\sim\mathcal{D}}[x = a] = 0$.

Let $\mathcal{P}^{(t)}_{\mathsf{rand}}$ and $\mathcal{P}^{(t)}_{C}$ be the distributions of the transcript of the first t turns when the input is drawn from $\mathcal{A}_{\mathsf{rand}}$ or $\mathcal{A}_C$, respectively. Note that to prove the theorem, it suffices to show that $\mathcal{P}^{(n)}_{\mathsf{rand}}$ is close to most $\mathcal{P}^{(n)}_{C}$. For this purpose, we are going to prove the following inequality holds for any $t \leq n$:

$$\mathop{\mathbb{E}}_{C\sim\mathcal{S}^{[n]}_k}\left[\left\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{C}\right\|\right] \leq t \cdot c_1 \cdot \frac{k^2}{n} \cdot \frac{1}{\sqrt{n}}, \tag{1}$$

where c_1 is a large enough universal constant. This inequality states that when picking a clique at random, the distribution of transcripts is similar to the distribution of transcripts when the input is chosen uniformly at random. It is easy to see that plugging in $t = n$, (1) implies the theorem.

We prove the inequality above inductively. Clearly, (1) holds when $t = 0$. So it suffices to show that when it holds for $t-1$, it also holds for t.

Let $\mathcal{D}_t$ and $\mathcal{D}^C_t$ be the input distributions to processor t in $\mathcal{A}_{\mathsf{rand}}$ and $\mathcal{A}_C$, respectively. For a fixed $C \subseteq [n]$, by Lemma 1.9, we have:

$$\left\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{C}\right\| \leq \left\|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{C}\right\| + \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f^{|p}_t(\mathcal{D}_t) - f^{|p}_t(\mathcal{D}^C_t)\right\|\right]. \tag{2}$$

We think of $\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{C}\|$ as the amount of evidence the transcript is giving us about whether the distribution is uniform or if it has C as a clique. (If this value were 0, that would mean the transcript gives us no evidence. If the value were 1, that would mean we have "full" evidence and could distinguish between the two with no error given the transcript). So, with this interpretation, the inequality above is basically stating that the amount of evidence we have after round t is the sum of the evidence from all rounds up to $t-1$, plus the extra evidence we get from the broadcast of the processor in round t.

By definition, $\mathcal{D}_t$ is the uniform distribution on the set $\{x : x \in \{0,1\}^n, x_t = 0\}$. And $\mathcal{D}^C_t = \mathcal{D}_t$ if $t \notin C$, and is the uniform distribution on the set $\{x : x \in \{0,1\}^n, x_t = 0, x_j = 1 \text{ for all } j \in C \setminus \{t\}\}$ otherwise.

We care not about the probability of distinguishing a particular clique existing, but about whether any clique exists, so we take the expected value over all possible cliques of both sides of (2) gives:

$$\mathop{\mathbb{E}}_{C\sim\mathcal{S}^{[n]}_k}\left[\left\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{C}\right\|\right] \leq \mathop{\mathbb{E}}_{C\sim\mathcal{S}^{[n]}_k}\left[\left\|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{C}\right\|\right] + \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\mathop{\mathbb{E}}_{C\sim\mathcal{S}^{[n]}_k}\left[\left\|f^{|p}_t(\mathcal{D}_t) - f^{|p}_t(\mathcal{D}^C_t)\right\|\right]. \tag{3}$$

We can bound $\mathbb{E}_{C\sim\mathcal{S}^{[n]}_k}\left[\left\|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{C}\right\|\right]$ by the inductive hypothesis, so it suffices to bound $\mathbb{E}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\mathbb{E}_{C\sim\mathcal{S}^{[n]}_k}\left[\left\|f^{|p}_t(\mathcal{D}_t) - f^{|p}_t(\mathcal{D}^C_t)\right\|\right]$. For $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$, there are two cases:

- When $t \notin C$, which happens with probability $1 - \frac{k}{n}$, we have

  $$\left\|f^{|p}_t(\mathcal{D}_t) - f^{|p}_t(\mathcal{D}^C_t)\right\| = 0,$$

  as $\mathcal{D}^C_t = \mathcal{D}_t$. That is, since t is not in C, what t says gives us no information about whether C is a clique (whether there is a clique or not does not affect the input of t, and therefore does not affect their message).

- When $t \in C$, which happens with probability $\frac{k}{n}$, by Lemma 1.8, we have

  $$\mathop{\mathbb{E}}_{C'\sim\mathcal{S}^{[n]\setminus\{t\}}_{k-1}}\left[\left\|f^{|p}_t(\mathcal{D}_t) - f^{|p}_t(\mathcal{D}^{C'\cup\{t\}}_t)\right\|\right] \leq O\left(\frac{k}{\sqrt{n}}\right).$$

That is, when t is in C, while t might give information about whether C is a clique, there are many cliques that may include t, and the inequality states that t cannot give too much information about many of cliques (the expected amount of information revealed about a randomly chosen clique of size k containing t is bounded by $O\left(\frac{k}{\sqrt{n}}\right)$).

So, now we have bounded how much evidence the t-th processor reveals when broadcasting. We know that when the clique is chosen randomly, with probability $1 - k/n$ no information is revealed, and with probability k/n at most $O(k/\sqrt{n})$ information is revealed in expectation. Combining these facts gives:

$$\mathop{\mathbb{E}}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}} \mathop{\mathbb{E}}_{C\sim\mathcal{S}_k} \left[\left\| f_t^{|p}(\mathcal{D}_t) - f_t^{|p}(\mathcal{D}_t^C)\right\|\right] \leq \frac{k}{n}\cdot O\left(\frac{k}{\sqrt{n}}\right),$$

which, plugging into (3) and using the inductive hypothesis proves inequality (1) for t. □

1.3.1 Proof for Lemma 1.8

We need the following lemma first, whose proof is based on tools from information theory, and is deferred to the end of this subsection.[7]

Lemma 1.10. *Let n be an integer, and $\mathcal{U}_n^{[i]}$ be the uniform distribution on $\{x : x \in \{0,1\}^n, x_i = 1\}$. For all function $f : \{0,1\}^n \to \{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{i\leftarrow[n]} \left[\left\| f(\mathcal{U}) - f(\mathcal{U}^{[i]})\right\|\right] \leq O\left(\frac{1}{\sqrt{n}}\right).$$

That is, if we consider a function f, suppose that on a uniform distribution, the probability it outputs 1 is p. Then, if we pick a random index and set it to 1, we still expect that if we randomly pick the rest of the coordinates, the output will be 1 with probability approximately p.

Now we are ready to prove Lemma 1.8 (restated below).

Reminder of Lemma 1.8 *Let n, k be integers such that $k \leq n^{1/4}$, and $\mathcal{U}_n^C$ be the uniform distribution on $\{x : x \in \{0,1\}^n, x_i = 1$ for all $i \in C\}$. For all function $f : \{0,1\}^n \to \{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}} \left[\left\| f(\mathcal{U}_n) - f(\mathcal{U}_n^C)\right\|\right] \leq O\left(\frac{k}{\sqrt{n}}\right).$$

The idea of the proof is that each bit we set to 1, by Lemma 1.10, will change the expected output of f by $O\left(\frac{1}{\sqrt{n}}\right)$. Hence, if we set k of those bits to 1, that will change the expected outcome by $O\left(\frac{1}{\sqrt{n}}\right)$ at most k times, for a total of $O\left(\frac{k}{\sqrt{n}}\right)$. A formal proof is included below:

Proof of Lemma 1.8. Instead of choosing C from $\mathcal{S}_k^{[n]}$, we choose an ordered k-tuple of $a = (a_1, a_2, \ldots, a_k)$ of k distinct elements in $[n]$ uniformly at random. Let the distribution be $\mathcal{T}_k^{[n]}$.

[7] This lemma is standard and can be proved in various ways. We present a proof based on information theory because it can be easily generalized to a proof for Lemma 4.4, which is used in Section 4.

We have

$$\begin{aligned}\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|f(\mathcal{U}_n)-f(\mathcal{U}_n^C)\right\|\right] &= \mathop{\mathbb{E}}_{a\sim\mathcal{T}_k^{[n]}}\left[\left\|f(\mathcal{U}_n)-f(\mathcal{U}_n^{\{a_i\}_{i=1}^k})\right\|\right]\\ &\le \sum_{\ell=1}^{k}\mathop{\mathbb{E}}_{a\sim\mathcal{T}_\ell^{[n]}}\left[\left\|f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell-1}})-f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell}})\right\|\right]\\ &\le \sum_{\ell=0}^{k-1}\mathop{\mathbb{E}}_{a\sim\mathcal{T}_\ell^{[n]}}\mathop{\mathbb{E}}_{j\leftarrow[n]\setminus\{a_i\}_{i=1}^{\ell}}\left[\left\|f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell}})-f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell}\cup\{j\}})\right\|\right]. \qquad (4)\end{aligned}$$

Now we are to bound the right side of (4) for each $0\le\ell\le k-1$ separately. Applying Lemma 1.10 on the restriction of f such that all bits in $\{a_i\}$ are set to 1, we have

$$\mathop{\mathbb{E}}_{j\leftarrow[n]\setminus\{a_i\}_{i=1}^{\ell}}\left[\left\|f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell}})-f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell}\cup\{j\}})\right\|\right]\le O\left(\frac{1}{\sqrt{n-\ell}}\right).$$

Plugging the above in (4) and noting that $k\le n^{1/4}$ completes the proof. □

Now we prove Lemma 1.10. The proof makes use several tools from information theory, see Section 2.4 for the details.

Proof of Lemma 1.10. Throughout the proof we will assume X is a random variable drawn uniformly from $\{0,1\}^n$. For $i\in[n]$, let X_i be the random variable of the i-th bit of X.

We have

$$I(X_i;f(X))=H(X_i)-H(X_i|f(X))=1-H(X_i|f(X)).$$

And by the sub-additivity of conditional entropy, we have

$$\sum_{i=1}^{n}H(X_i|f(X))\ge H(X|f(X))\ge n-1.$$

Therefore,

$$\sum_{i=1}^{n}I(X_i;f(X))\le n-(n-1)\le 1.$$

or equivalently,

$$\mathop{\mathbb{E}}_{i\leftarrow[n]}I(X_i;f(X))\le\frac{1}{n}.$$

Note that by Fact 2.1,

$$I(X_i;f(X)):=\mathop{\mathbb{E}}_{x\sim X_i}D(f(X)_{X_i=x}||f(X)).$$

Taking expected values over i of both sides and using $\mathbb{E}_{i\leftarrow[n]}I(X_i;f(X))\le\frac{1}{n}$ gives

$$\mathop{\mathbb{E}}_{i\leftarrow[n]}I(X_i;f(X))=\mathop{\mathbb{E}}_{i\leftarrow[n]}\mathop{\mathbb{E}}_{x\sim X_i}D(f(X)_{X_i=x}||f(X))\le\frac{1}{n}.$$

By Pinsker's inequality (Lemma 2.2) and the fact that $\sqrt{x}$ is a concave function, we have

$$\mathop{\mathbb{E}}_{i\leftarrow[n]}\mathop{\mathbb{E}}_{x\sim X_i}\|f(X)_{X_i=x}-f(X)\|\le\sqrt{\frac{1}{n}},$$

and

$$\mathop{\mathbb{E}}_{i\leftarrow[n]}\frac{1}{2}\cdot\|f(X)_{X_i=1}-f(X)\|\le\sqrt{\frac{1}{n}}.$$

Note that by definition, $f(X)_{X_i=1}$ is distributed identically to $f(\mathcal{U}^{[i]})$, which completes the proof. □

1.4 Related Work

BCAST(1) Congested Clique: The specific distributed model we investigate is the Broadcast Congested Clique. In this model, there are n processors, and computation proceeds in rounds. In each round, each processor broadcasts a short message to all other processors. It has recently been studied in [BARR15, dERRU16, CKK$^+$15, GHM18, DKO14, Gal16, NY19, HP15, CPS17, BMRT18, JN17b, MT16, JN17a], among others. It has been used to study other areas in computer science such as streaming algorithms [AMS99] and mechanism design [DNO14].

Complexity Theoretic approaches in Distributed Computing: Recently, more complexity theoretic approaches and results have been made in the congested clique and distributed computation in general, for example in [KS18, FKP13, GKM17, CP17].

Pseudo-randomness and Distributed Computing: In [BGR96], the authors construct a pseudo-random generator which creates additional shared random bits in a distributed system. Specifically, the authors work in a setting where every pair of processors can privately communicate with each other (whereas we work in the broadcast model), and some of the processors may be adversarially faulty. They show how to use few shared random bits, and unlimited private random bits to efficiently compute more shared random bits. In our setting, we are saving on private random bits flipped (when all processors are non-faulty, it is easy to turn a private random bit into a public random bit – simply broadcast it).

In [INW94], the authors construct pseudo-randomness for a different distributed system, in which the network has a topology. Their main application is constructing pseudo-randomness that fools all low-space computation.

In [NPR99], a different setting than ours is considered, in which the processors are computationally bounded, and a cryptographic pseudo-random function is being evaluated.

In [PY18], a pseudo-random generator that fools DNFs is used to deterministically construct spanners in the congested clique. In that work, the pseudo-random generator is used for the specific problem considered, as opposed to being a pseudo-random generator which fools all algorithms in the model.

Pseudo-randomness in the context of complexity theory has been very widely studied. See Vadhan's survey [Vad12].

In [GHK18], the authors introduce general methods for derandomizing algorithms in the LOCAL model to obtain better deterministic algorithms.

Planted Clique: The planted clique problem (or hidden clique problem) was introduced in [Jer92] and [Kuc95]. The best known classical algorithm [FK00, DGP14] can find the hidden clique when its size is $k = \Omega(\sqrt{n})$ in near linear-time. For $k \ll \sqrt{n}$, the naïve algorithm (looking for a clique of size $10 \log n$ with brute force, and then extending that clique to the whole clique) can solve it in $n^{O(\log n)}$ time, and the problem is conjectured to be not solvable in polynomial time. However, since it is an average-case problem, it is unlikely that the hardness of this problem can be derived from standard complexity assumptions such as $\mathsf{P} \neq \mathsf{NP}$ [FF93, BT06]. Therefore, much work has been put into trying to show limitations for certain classes of algorithm on this problem [FK03, MPW15, DM15, HKP$^+$18, BHK$^+$16], or showing tight hardness for closely-related worst-case problems under standard assumptions [BKRW17].

Distributed Clique lower bounds There is some literature on lower bounds for finding cliques in the congested clique model [DKO14], as well as the standard CONGEST model [CK18]. These lower bounds hold in the worst case, and have no direct implications about the hardness of the planted clique problem.

Organization of the Paper

In Section 2, we introduce the needed preliminaries for this paper. Section 3 we present an abstract framework for our approach. In Section 4 we prove the lower bound for planted clique in $\mathsf{BCAST}(1)$. In Section 5 we give an overview of the proofs for the PRG construction for $\mathsf{BCAST}(1)$, starting with a one-round toy example. Then in Section 6, we show how to create a single pseudo-random bit for each processor, which also implies our average case lower bound for $\mathsf{BCAST}(1)$. Next, in Section 7 we show how to create many pseudo-random bits. Finally, in Section 8 we show our pseudo-random generator's parameters are optimal.

2 Preliminaries

2.1 Notations

Here we summarize some standard notations which are used in this paper.

For integers n and m, we use $\mathcal{U}_m$ to denote the uniform distribution on $\{0,1\}^m$, and $\mathcal{U}_{n\times m}$ to denote the uniform distribution on $\{0,1\}^{n\times m}$.

Let X, Y be two sets. For a function $f : X \to Y$ and a distribution $\mathcal{D}$ on X, we use $f(\mathcal{D})$ to denote the distribution of the output of f when the input is drawn from $\mathcal{D}$. For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ on a set X, we use $\|\mathcal{D}_1 - \mathcal{D}_2\| = \frac{1}{2}\sum_{x\in X} |\mathcal{D}_1(x) - \mathcal{D}_2(x)|$ to denote their statistical distance (where $\mathcal{D}(x)$ is the probability that a sample from $\mathcal{D}$ equals x).

2.2 Analysis of Boolean Functions

Our proofs make use of some well-known facts from analysis of Boolean functions[8].

For any function $f : \{0,1\}^n \to \mathbb{R}$, its *Fourier coefficient* at a set S is defined as

$$\widehat{f}(S) := \mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\left[f(x)\cdot(-1)^{\sum_{i\in S} x_i}\right].$$

Parsevals Identity states

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\left[f(x)^2\right] = \sum_{S\subseteq[n]} \widehat{f}(S)^2.$$

2.3 Probability Theory

The following lemma is standard. We provide a proof here for completeness.

Reminder of Lemma 1.9 *Let X and Y be two sets, and $\mathcal{D}$ and $\mathcal{D}'$ be two distributions on $X \times Y$. Let $\mathcal{D}_{|X}$ and $\mathcal{D}'_{|X}$ be the respective marginal distribution of $\mathcal{D}$ and $\mathcal{D}'$ on set X. For $a \in X$, we use $\mathcal{D}_{X=a}$ and $\mathcal{D}'_{X=a}$ to denote the respective conditional distribution of $\mathcal{D}$ and $\mathcal{D}'$ on Y conditioning on $X = a$.[9] We have*

$$\|\mathcal{D} - \mathcal{D}'\| \le \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\| + \mathop{\mathbb{E}}_{a\sim\mathcal{D}_{|X}}\left[\|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|\right].$$

[8] Some nice references can be found in [DW08, O'D14]

[9] For simplicity, we let $\mathcal{D}_{X=a}$ be the uniform distribution on Y if $\Pr_{(x,y)\sim\mathcal{D}}[x = a] = 0$.

Proof. We first define an auxiliary distribution $\mathcal{D}_{\mathsf{aux}}$ as follows: for $(a,b) \in X \times Y$, if $\mathcal{D}'_{|X}(a) > 0$, (we use $\mathcal{D}'_{|X}(a)$ to denote the probability that a sample from $\mathcal{D}'_{|X}$ equal a).

$$\mathcal{D}^{\mathsf{aux}}(a,b) := \mathcal{D}'(a,b) \cdot \frac{\mathcal{D}_{|X}(a)}{\mathcal{D}'_{|X}(a)}.$$

Otherwise, we set $\mathcal{D}^{\mathsf{aux}}(a,b) := \frac{1}{|Y|} \cdot \mathcal{D}_{|X}(a)$. It is easy to verify that $\mathcal{D}^{\mathsf{aux}}_{|X} = \mathcal{D}_{|X}$ and for all a $\mathcal{D}^{\mathsf{aux}}_{X=a} = \mathcal{D}'_{X=a}$, and therefore it is a distribution.

Now, it is easy to see that

$$\|\mathcal{D}^{\mathsf{aux}} - \mathcal{D}'\| = \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\|.$$

Moreover, we have

$$\|\mathcal{D}^{\mathsf{aux}} - \mathcal{D}\| = \mathop{\mathbb{E}}_{a \sim \mathcal{D}_{|X}} \left[\|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|\right].$$

Putting everything together, we have

$$\begin{aligned}\|\mathcal{D} - \mathcal{D}'\| &\leq \|\mathcal{D}^{\mathsf{aux}} - \mathcal{D}'\| + \|\mathcal{D}^{\mathsf{aux}} - \mathcal{D}\|. \\ &\leq \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\| + \mathop{\mathbb{E}}_{a \sim \mathcal{D}_{|X}} \left[\|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|\right].\end{aligned}$$

□

2.4 Information Theory

In this paper we need some definitions and facts from information theory. For an excellent introduction to information theory, one is referred to the textbook by Cover and Thomas [CT06]. We consider discrete random variables in this paper.

Let X, Y be random variables in the same probability space Ω. The entropy of X, denoted by $H(X)$, is defined as $H(X) := \Pr_{a \sim X} \log \frac{1}{\Pr[X=a]}$. The conditional entropy of X given Y, denoted as $H(X|Y)$, is defined as $H(X|Y) := \mathbb{E}_{y \sim Y} H(X|Y=y)$.

The mutual information between X and Y, denoted by $I(X;Y)$, is defined as $I(X;Y) := H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y)$.

For two distributions $\mathcal{P}$ and $\mathcal{Q}$ on the same set S, their Kullback-Leibler (KL) divergence is defined as

$$D(\mathcal{P}||\mathcal{Q}) := \sum_{s \in S} \mathcal{P}(s) \cdot \log \frac{\mathcal{P}(s)}{\mathcal{Q}(s)}.$$

KL divergence is related to mutual information in the following way.

Fact 2.1.

$$I(X;Y) := \mathop{\mathbb{E}}_{x \sim X} D(Y|X=x||Y),$$

Lemma 2.2 (Pinsker's Inequality)**.** *For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we have*

$$\|\mathcal{D}_1 - \mathcal{D}_2\| \leq \sqrt{\frac{1}{2} \cdot D(\mathcal{D}_1||\mathcal{D}_2)}.$$

For a real $p \in [0,1]$, we use $\mathsf{Ber}(p)$ to denote the binary Bernoulli random variable with expectation p. We also use $H(p)$ to denote the $H(\mathsf{Ber}(p))$. We have the following fact.

Fact 2.3. *If $H(p) \geq 0.9$, we have $p \in [0.3, 0.7]$, and*

$$\frac{1 - H(p)}{(p - 1/2)^2} \in [2,3].$$

3 Abstract Framework

In this section, we present an abstraction of our framework. Understanding this section is not necessary to understand the rest of the sections of the paper. It is included to make it easier to understand the structure of the proof without having to dig through the problem-specific technical parts.

In the following we exhibit an abstract framework for showing a certain input distribution $\mathcal{A}_{\mathsf{pseudo}}$ is indistinguishable from the uniform random input distribution $\mathcal{A}_{\mathsf{rand}}$ by a low round $\mathsf{BCAST}(1)$ protocol[10].

For the simplicity of discussion. We assume each of the n processors gets n bits as its input. We also use a matrix $A \in \{0,1\}^{n\times n}$ to denote their inputs collectively, where the i-th player gets the i-th row of A. Then $\mathcal{A}_{\mathsf{rand}}$ is simply the uniform distribution over $\{0,1\}^{n\times n}$.

Notations. We first recall and introduce some notations. Let $\mathcal{U}_m$ denote the uniform distribution on $\{0,1\}^m$. For a function $f : \{0,1\}^* \to \{0,1\}^*$ and a distribution $\mathcal{D}$ on $\{0,1\}^*$, we use $f(\mathcal{D})$ to denote the distribution of the output of f when the input is drawn from $\mathcal{D}$. For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we use $\|\mathcal{D}_1 - \mathcal{D}_2\| = \frac{1}{2}\sum_{x\in\{0,1\}^*} |\mathcal{D}_1(x) - \mathcal{D}_2(x)|$ to denote their statistical distance (where $\mathcal{D}(x)$ is the probability that a sample from $\mathcal{D}$ equals x).

By Yao's principle [Yao77], we can assume all processors are deterministic as we are trying to prove a lower bound for distinguishing two input distributions. Processor i can then be defined by a function $f_i : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}$, such that $f_i(z,p)$ is the bit that player i outputs when it gets the input z and transcript p. We use $f_i^{|p}$ to denote the function $f_i(\cdot, p)$ for simplicity. If transcript p is incompatible with player i having input z, then we set $f_i(z,p)$ arbitrarily.

Given a $\mathsf{BCAST}(1)$ protocol Π and an input distribution $\mathcal{D}$, we use $\mathcal{P}(\Pi, \mathcal{D})$ to denote the distribution of the transcripts of the protocol Π running on a input drawn from $\mathcal{D}$. We also use $\mathcal{P}^{(t)}(\Pi, \mathcal{D})$ to denote the distribution of the same transcript in first t turns.

A Relaxation

Instead of viewing the algorithm as a single round algorithm, we will prove a slightly stronger lower bound. Consider the model where we have $j \cdot n$ turns instead of j rounds. On the t^{th} turn, processor $(t-1) \bmod n+1$ gets to send a single bit. This model is stronger than j rounds of the $\mathsf{BCAST}(1)$ model, since it allows the later processors to condition their outputs on earlier processors' messages. Hence, lower bounds for this relaxed model imply lower bounds for the $\mathsf{BCAST}(1)$ model as well.

Decomposition into Row-Independent Distributions

We first write $\mathcal{A}_{\mathsf{pseudo}}$ as an average of many row-independent distributions. Let $\mathcal{I}$ be an index set, and $\{\mathcal{A}_I\}_{I\in\mathcal{I}}$ be a family of distributions, we need the following two properties:

- $\mathcal{A}_{\mathsf{pseudo}} = \frac{1}{|\mathcal{I}|}\sum_{I\in\mathcal{I}} \mathcal{A}_I$. That is, $\mathcal{A}_{\mathsf{pseudo}}$ can be written as an average of all distributions in $\{\mathcal{A}_I\}_{I\in\mathcal{I}}$.
- For each $I \in \mathcal{I}$, $\mathcal{A}_I = \bigoplus_{i=1}^n \mathcal{A}_I^{[i]}$, where $\oplus$ means concatenation and all $\mathcal{A}_I^{[i]}$'s are independent. Equivalently, rows in $\mathcal{A}_I$ are *independent*. (Each row is a single node's input).

[10] It is not imperative that one of the distributions is uniform. We decide to present the framework with one of the distributions as uniform for the sake of simplicity, and since our two main applications of the framework in this paper involve distinguishing distributions from uniform.

Progress Function

We first fix a $\mathsf{BCAST}(1)$ protocol Π. For simplicity, we define $\mathcal{P}^{(t)}_{\mathsf{rand}} = \mathcal{P}^{(t)}(\Pi, \mathcal{A}_{\mathsf{rand}})$, $\mathcal{P}^{(t)}_{\mathsf{pseudo}} = \mathcal{P}^{(t)}(\Pi, \mathcal{A}_{\mathsf{pseudo}})$ and $\mathcal{P}^{(t)}_I = \mathcal{P}^{(t)}(\Pi, \mathcal{A}_I)$ for $I \in \mathcal{I}$.

Ideally, we would like to bound

$$\mathcal{L}^{(t)}_{\mathsf{real\text{-}dist}} := \left\| \mathcal{P}^{(t)}_{\mathsf{pseudo}} - \mathcal{P}^{(t)}_{\mathsf{rand}} \right\|$$

round by round. But as discussed in the introduction, the above is very hard to work with, so we try to bound the following progress function instead:

$$\mathcal{L}^{(t)}_{\mathsf{progress}} := \mathop{\mathbb{E}}_{I \leftarrow \mathcal{I}} \left[\left\| \mathcal{P}^{(t)}_I - \mathcal{P}^{(t)}_{\mathsf{rand}} \right\| \right].$$

It is not hard to see that $\mathcal{L}^{(t)}_{\mathsf{real\text{-}dist}} \le \mathcal{L}^{(t)}_{\mathsf{progress}}$: we know that $\mathcal{L}^{(t)}_{\mathsf{progress}} = \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \left\| \mathcal{P}^{(t)}_I - \mathcal{P}^{(t)}_{\mathsf{rand}} \right\|$, which by a triangle inequality is greater than or equal to $\left\| \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \left[\mathcal{P}^{(t)}_I - \mathcal{P}^{(t)}_{\mathsf{rand}} \right] \right\| = \left\| \mathcal{P}^{(t)}_{\mathsf{pseudo}} - \mathcal{P}^{(t)}_{\mathsf{rand}} \right\|$ so showing an upper bound on $\mathcal{L}^{(t)}_{\mathsf{progress}}$ is sufficient for upper bounding $\mathcal{L}^{(t)}_{\mathsf{real\text{-}dist}}$.

Upper Bounding the Progress Made in Turn t

Now suppose we are at the t-th turn. Let j be the current round number, and i be the broadcasting processor of this turn. By Lemma 1.9, for all $I \in \mathcal{I}$, we have

$$\left\| \mathcal{P}^{(t)}_I - \mathcal{P}^{(t)}_{\mathsf{rand}} \right\| \le \left\| \mathcal{P}^{(t-1)}_I - \mathcal{P}^{(t-1)}_{\mathsf{rand}} \right\| + \mathop{\mathbb{E}}_{p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}} \left[\left\| f_i^{|p}(\mathcal{D}_i|p) - f_i^{|p}(\mathcal{D}_i^I|p) \right\| \right]. \tag{5}$$

In above, $\mathcal{D}_i|p$ and $\mathcal{D}_i^I|p$ are the input distributions to player i conditioning on seeing the transcript p of the previous $t-1$ rounds, in $\mathcal{A}_{\mathsf{rand}}$ and $\mathcal{A}_I$ respectively.

First, since $\mathcal{D}_i$ is just $\mathcal{U}_n$, we can see $\mathcal{D}_i|p$ is simply the uniform distribution on the set of inputs which is consistent with the transcript p. Formally, let $t_1, t_2, \ldots, t_{j-1}$ be the indices of all previous $j-1$ turns with processor i broadcasting, before the current t-th turn. For $x \in \{0,1\}^n$, we say that x is consistent with transcript p, if for all $\ell \in [j-1]$, we have

$$f_i^{|p^{(t_\ell - 1)}}(x) = p_{t_\ell},$$

where $p^{(t_\ell - 1)}$ denotes the first $t_\ell - 1$ bits of p. That is, simulating f_i with respect to p on x gives the same outputs in p.

Let $D_p^{(t-1)}$ denote the set of inputs to f_i which are consistent with the transcript p. Then we can see $\mathcal{D}_i|p$ is the uniform distribution on $D_p^{(t-1)}$. Similarly, since $\mathcal{A}_I$ is row-independent, $\mathcal{D}_i^I|p$ is just $\mathcal{A}_I^{[i]}$ conditioning on $D_p^{(t-1)}$. We denote this as $\mathcal{A}_I^{[i]}|D_p^{(t-1)}$.

Plugging in (5), and taking an expectation for all $I \in \mathcal{I}$, we have

$$\mathop{\mathbb{E}}_{I \leftarrow \mathcal{I}} \left\| \mathcal{P}^{(t)}_I - \mathcal{P}^{(t)}_{\mathsf{rand}} \right\| \le \mathop{\mathbb{E}}_{I \leftarrow \mathcal{I}} \left\| \mathcal{P}^{(t-1)}_I - \mathcal{P}^{(t-1)}_{\mathsf{rand}} \right\| + \mathop{\mathbb{E}}_{p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}} \mathop{\mathbb{E}}_{I \leftarrow \mathcal{I}} \left[\left\| f_i^{|p}(\mathcal{U}_{D_p^{(t-1)}}) - f_i^{|p}(\mathcal{A}_I^{[i]}|D_p^{(t-1)}) \right\| \right]. \tag{6}$$

A key observation here is that $\mathcal{D}_p^{(t-1)}$ is usually a large set over $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$. The proof of the following claim is essentially the same as the proof for Claim 2 in Section 4, so we omit it here.

Claim 1. *For all $\varepsilon > 0$,*

$$\Pr_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}}\left[D_p^{(t-1)} \geq 2^{n-j}\cdot\varepsilon\right] \geq 1-\varepsilon.$$

Therefore, in order to bound the second term of the right hand side of (6), we can assume $|D_p^{(t-1)}| \geq 2^{n-\Theta(\beta)}$, where β is roughly the round lower bound we wish to prove.

Statistical Inequality Task

Now we are finally able to specify the statistical inequality task we need to prove. We want to show that for almost all $p \sim \mathcal{P}_{\mathsf{rand}}^{(t-1)}$, their contribution to the second term of the right side of (6),

$$\mathop{\mathbb{E}}_{I\leftarrow\mathcal{I}}\left[\left\|f_i^{|p}(\mathcal{U}_{D_p^{(t-1)}}) - f_i^{|p}(\mathcal{A}_I^{[i]}|D_p^{(t-1)})\right\|\right],$$

is small. We obviously have no control over the set $D_p^{(t-1)}$ except for that it is large, so we want the following type of statistically inequality.

Required Lemma Format. *Let $D \subseteq \{0,1\}^n$ with $|D| \geq 2^{n-\beta}$, $\mathcal{U}_D$ be the uniform distribution on D. For all function $f : D \to \{0,1\}$ and $i \in [n]$, we have*

$$\mathop{\mathbb{E}}_{I\leftarrow\mathcal{I}}[\|f(\mathcal{U}_D) - f(\mathcal{A}_I^{[i]}|D)\|] \leq \varepsilon(n,\beta).$$

In above $\varepsilon(n,\beta)$ is some error function which is increasing in β.

It will be helpful to observe that Lemma 1.8, Lemma 4.3, Lemma 5.2, Lemma 6.1, and Lemma 7.2 are all instantiations of the above required lemma (for proving the one-round lower bound we can simply assume $D = \{0,1\}^n$).

Once we have the required lemma, then by a simple induction, we have

$$\mathcal{L}_{\mathsf{progress}}^{(j\cdot n)} \leq \sum_{\ell=1}^{j}\varepsilon(n,\Theta(\ell))) \leq (j\cdot n)\cdot\varepsilon(n,\Theta(j)).$$

From which we can deduce the needed lower bound, if $(j\cdot n)\cdot\varepsilon(n,\Theta(j)) \ll 1$.

4 Lower Bound for Planted Clique

In this section we prove that the planted clique problem is hard for $\mathsf{BCAST}(1)$ when $k = n^{1/4-\varepsilon}$. We encourage the reader to read Section 1.3 before this section. That subsection contains a one-round lower bound for the problem, which involves a similar yet much less technical proof.

Notations. We first recall some notations. Let $\mathcal{A}_{\mathsf{rand}}^n$ be the distribution on $\{0,1\}^{n\times n}$ such that for a sample A from $\mathcal{A}_{\mathsf{rand}}^n$, for all $i \neq j$, $A_{i,j}$ is an independent uniform random bit in $\{0,1\}$, and $A_{i,i}$ is always 0 for all i. Let C be a subset of $[n]$. We use $\mathcal{A}_C^n$ to denote the conditional distribution of $\mathcal{A}_{\mathsf{rand}}^n$ on the event that for all $i,j \in C$ and $i \neq j$, $A_{i,j} = 1$ (that is, C is a clique). We also use $\mathcal{A}_k^n$ to denote the mixed distribution of $\mathcal{A}_C^n$'s when C is a uniformly chosen random subset of $[n]$ of size k.

For a distribution $\mathcal{A}$, we use $\mathcal{A}^{[i]}$ to denote it's marginal distribution on the i-th row. Note that $\mathcal{A}^n_{\mathsf{rand}}$ and $\mathcal{A}^n_C$ have independent rows[11].

When the meaning is clear, we often drop the superscripts of the above distributions for simplicity.

Given a $\mathsf{BCAST}(1)$ protocol Π and an input distribution $\mathcal{D}$, we use $\mathcal{P}(\Pi, \mathcal{D})$ to denote the distribution of the transcripts of the protocol Π running on an input drawn from $\mathcal{D}$ (that is, given a matrix A which is drawn from the distribution $\mathcal{D}$, the processor i gets the i-th row of A, and all processors act according to the protocol Π).

In this section we prove the following theorem:

Theorem 4.1. *Let n be the number of processors. For any j-round $\mathsf{BCAST}(1)$ protocol Π, we have*

$$\|\mathcal{P}(\Pi, \mathcal{A}_{\mathsf{rand}}) - \mathcal{P}(\Pi, \mathcal{A}_k)\| \leq O\left(j \cdot k^2 \cdot \sqrt{\frac{j + \log n}{n}}\right).$$

As a simple corollary, we immediately have:

Corollary 4.2 ($\mathsf{BCAST}(1)$ Lower Bound for Planted Clique)**.** *For any constant $\varepsilon > 0$, if $k = n^{1/4-\varepsilon}$ then no $n^{o(1)}$ round $\mathsf{BCAST}(1)$ protocol Π can distinguish between $\mathcal{A}_{\mathsf{rand}}$ and $\mathcal{A}_k$ with advantage $\Omega(1)$.*

Let $\mathcal{S}^T_k$ be the uniform distribution on all size-k subsets of T. To prove Theorem 4.1, we need the following technical lemma, whose proof is deferred to the end of this section.

Lemma 4.3. *Let n, t, k be integers such that $t, k \leq n^{1/4}$ and $t \geq 10 \log n$, D be a subset of $\{0,1\}^n$ with $|D| \geq 2^{n-t}$, $\mathcal{U}_D$ be the uniform distribution on D, and $\mathcal{U}^C_D$ be the uniform distribution on $\{x : x \in D, x_i = 1 \text{ for all } i \in C\}$. For all functions $f : D \to \{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{C \sim \mathcal{S}^{[n]}_k}[\|f(\mathcal{U}_D) - f(\mathcal{U}^C_D)\|] \leq O\left(k \cdot \sqrt{\frac{t}{n}}\right).$$

(If $\mathcal{U}^C_D$ is empty, we define $\|f(\mathcal{U}_D) - f(\mathcal{U}^C_D)\| = 1$).

Intuitively speaking, the D in the lemma above corresponds to the set of inputs to a certain node which are consistent with the current transcript. Each time the node broadcasts a bit, the size of D is expected to reduce by at most a constant factor, so after r rounds one would expect D to be larger than $2^{n-\Theta(r)}$.

Now we are ready to prove Theorem 4.1.

Proof of Theorem 4.1. Instead of viewing the algorithm as a standard j round algorithm, we will prove a slightly stronger lower bound. Consider the model where during each round we have n turns. On the t^{th} turn, processor $(t-1) \bmod n + 1$ gets to send a single bit. So, essentially, instead of all processors broadcasting their bit at the same time, they take turns. This model is stronger than one round of the $\mathsf{BCAST}(1)$ model, since it allows the later processors to condition their outputs on earlier the processors' messages. Hence, our lower bound implies a lower bound for the $\mathsf{BCAST}(1)$ model as well.

Let $\mathcal{P}^{(t)}_{\mathsf{rand}}$ and $\mathcal{P}^{(t)}_C$ be the distributions of the transcript of the first t turns when the input is drawn from $\mathcal{A}_{\mathsf{rand}}$ or $\mathcal{A}_C$, respectively. Note that to prove the theorem, it suffices to show that the distribution $\mathcal{P}^{(j \cdot n)}_{\mathsf{rand}}$ is close to $\mathcal{P}^{(j \cdot n)}_C$ for most choices of C. For this purpose, we are going to prove the following inequality holds for any $t \leq j \cdot n$:

[11] Fixing a clique C, all entries of the distribution $\mathcal{A}^n_C$ are independent: each edge outsize of C is an independent coin flip with probability 1/2, and each edge in C is an independent coin flip with probability 1. In particular, note that every two edges in the clique are independent, since they are both 1 with probability 1, and therefore the mutual information between the two entries is 0.

$$\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|\mathcal{P}_{\mathsf{rand}}^{(t)}-\mathcal{P}_C^{(t)}\right\|\right]\le t\cdot\left(1/n^2+c_1\cdot\frac{k^2}{n}\cdot\sqrt{\frac{j+\log n}{n}}\right),\tag{7}$$

where c_1 is a large enough universal constant. It is easy to see that plugging in $t=j\cdot n$, (7) implies the theorem.

To prove (7), we induct on t. Clearly, (7) holds when $t=0$. So it suffices to show that when it holds for $t-1$, it also holds for t. Let i be the processor who is broadcasting at the t-th turn.

For a fixed $C\subseteq[n]$, by Lemma 1.9, we have:

$$\left\|\mathcal{P}_{\mathsf{rand}}^{(t)}-\mathcal{P}_C^{(t)}\right\|\le\left\|\mathcal{P}_{\mathsf{rand}}^{(t-1)}-\mathcal{P}_C^{(t-1)}\right\|+\mathop{\mathbb{E}}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}}\left[\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^C|p)\right\|\right].\tag{8}$$

In above, $\mathcal{D}_i|p$ and $\mathcal{D}_i^C|p$ are the input distributions to player i conditioning on seeing the transcript p of the previous $t-1$ rounds. Let $D_p^{(t-1)}$ denote the set of inputs from $\{x:x\in\{0,1\}^n,x_i=0\}$ to f_i which are consistent with the transcript p. We can see $\mathcal{D}_i|p$ is the uniform distribution on $D_p^{(t-1)}$, while $\mathcal{D}_i^C|p$ is the same as $\mathcal{D}_i|p$ when $i\notin C$, and is the uniform distribution on $\{x:x\in D_p^{(t-1)},x_j=1 \text{ for all } j\in C\setminus\{i\}\}$.

Taking the expected value over all cliques of both sides of (8) gives

$$\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|\mathcal{P}_{\mathsf{rand}}^{(t)}-\mathcal{P}_C^{(t)}\right\|\right]\le\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|\mathcal{P}_{\mathsf{rand}}^{(t-1)}-\mathcal{P}_C^{(t-1)}\right\|\right]+\mathop{\mathbb{E}}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}}\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^C|p)\right\|\right].\tag{9}$$

So, to prove (7), since we can bound $\mathbb{E}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|\mathcal{P}_{\mathsf{rand}}^{(t-1)}-\mathcal{P}_C^{(t-1)}\right\|\right]$ by the inductive hypothesis, it suffices to bound $\mathbb{E}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}}\mathbb{E}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^C|p)\right\|\right]$. We first show that for most $p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}$, $D_p^{(t-1)}$ is large (that is, after $t-1$ turns, we expect the set of inputs consistent with the transcript to be large). The proof for the following claim is deferred to the end of the whole proof.

Claim 2. *For $t\le j\cdot n\le\frac{k\cdot n}{10}$, with probability $1-1/n^2$ over $p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}$, we have $|D_p^{(t-1)}|\ge 2^{n-j}/n^3$.*

Now, given a p with $|D_p^{(t-1)}|\ge 2^{n-j}/n^3=2^{n-j-3\log n}$, we want to bound $\mathbb{E}_{C\sim\mathcal{S}_k^{[n]}}\left[\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^C|p)\right\|\right]$. There are two cases:

- When $i\notin C$, which happens with probability $1-\frac{k}{n}$, we have

  $$\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^C|p)\right\|=0,$$

  as $\mathcal{D}_i^C|p=\mathcal{D}_i|p$.

- When $i\in C$, which happens with probability $\frac{k}{n}$, by Lemma 4.3, we have

  $$\mathop{\mathbb{E}}_{C'\sim\mathcal{S}_{k-1}^{[n]\setminus\{i\}}}\left[\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^{C'\cup\{i\}}|p)\right\|\right]\le O\left(k\cdot\sqrt{\frac{j+\log n}{n}}\right).$$

Putting them together, we have

$$\mathop{\mathbb{E}}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}}\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k}\left[\left\|f_i^{|p}(\mathcal{D}_i|p)-f_i^{|p}(\mathcal{D}_i^C|p)\right\|\right]\le 1/n^2+\frac{k}{n}\cdot O\left(k\cdot\sqrt{\frac{j+\log n}{n}}\right),$$

which proves (7) for t. □

Finally, we prove Claim 2.

Proof of Claim 2. Let $t_1, t_2, \ldots, t_\ell$ be the indices of all previous ℓ turns with processor i broadcasting, before the current t-th turn. We have $\ell \le j$. Let $x \in \{z : z \in \{0,1\}^n, z_i = 0\}$, note that x is consistent with transcript p, if for all $a \in [\ell]$, we have

$$f_i^{|p^{(t_a-1)}}(x) = p_{t_a},$$

where $p^{(t_a-1)}$ denotes the first $t_a - 1$ bits of p. We set $F_i(x,p) = 1$ if x and p are consistent, and 0 otherwise.

Consider the random process of generating $p \sim \mathcal{P}_{\mathsf{rand}}^{(t-1)}$, suppose inputs to all processors other than i are fixed, let $x^{-i} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \{0,1\}^{(n-1)\times n}$ be those fixed input. Let $P_{x^{-i}}^{(t)}$ be the distribution of the transcript when $x_i \sim \mathcal{A}_{\mathsf{rand}}^{[i]}$, and all other processors get (fixed) input according to x^{-i}.

For a fixed x^{-i}, note that there are only 2^ℓ possible transcripts p from $P_{x^{-i}}^{(t-1)}$, as the transcript is determined after fixing the output of processor i at all ℓ rounds. Therefore, let $T(x^{-i}, x_i)$ be the transcript when all processors get inputs according to x^{-i} and x_i, we can see when $p \sim P_{x^{-i}}^{(t-1)}$, $F_i(x,p) = 1$ if and only if $T(x^{-i}, x) = p$. That is,

$$P_{x^{-i}}^{(t-1)}(p) = \Pr_{x_i \sim \mathcal{A}_{\mathsf{rand}}^{[i]}}[T(x^{-i}, x_i) = p] = D_p^{(t-1)}/2^{n-1}.$$

In above $P_{x^{-i}}^{(t-1)}(p)$ is the probability that getting p from distribution $P_{x^{-i}}^{(t-1)}$. Then we have

$$\begin{aligned}
&\Pr_{p \sim P_{x^{-i}}^{(t-1)}}\left[D_p^{(t-1)} < 2^{-j-3\log n} \cdot 2^n\right] \\
=&\Pr_{p \sim P_{x^{-i}}^{(t-1)}}\left[P_{x^{-i}}^{(t-1)}(p) < 2^{-j-3\log n+1}\right] \\
\le& 2^{-j-3\log n+1} \cdot 2^\ell = 1/n^2.
\end{aligned}$$

The last inequality holds because the support size of $P_{x^{-i}}^{(t-1)}$ is at most 2^ℓ and $\ell \le j$.

Hence, we have

$$\begin{aligned}
\Pr_{p \sim \mathcal{P}_{\mathsf{rand}}^{(t-1)}}\left[D_p^{(t-1)} < 2^{-j-3\log n} \cdot 2^n\right] &= \mathop{\mathbb{E}}_{x^{-i} \sim \mathcal{A}_{\mathsf{rand}}^{[-i]}}\left[\Pr_{p \sim P_{x^{-i}}^{(t-1)}}\left[D_p^{(t-1)} < 2^{-j-3\log n} \cdot 2^n\right]\right] \\
&\le 1/n^2.
\end{aligned}$$

In above $\mathcal{A}_{\mathsf{rand}}^{[-i]}$ denotes the marginal distribution of $\mathcal{A}_{\mathsf{rand}}$ on all rows except the i-th row. □

4.1 Proof for Lemma 4.3

We need the following lemma first, which is proved using tools from information theory.

Lemma 4.4. *Let n, t, k be integers such that $t, k \le n/10$, D be a subset of $\{0,1\}^n$ with $|D| \ge 2^{n-t}$, $\mathcal{U}_D$ be the uniform distribution on D, and $\mathcal{U}_D^{[i]}$ be the uniform distribution on $\{x : x \in D \text{ and } x_i = 1\}$, for all function $f : D \to \{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{i \leftarrow [n]}\left[\left\|f(\mathcal{U}_D) - f(\mathcal{U}_D^{[i]})\right\|\right] \le O\left(\sqrt{\frac{t}{n}}\right).$$

Proof. Let $D^{[i]} := \{x : x \in D \text{ and } x_i = 1\}$. Throughout the proof we will assume X is a random variable drawn uniformly from D. For $i \in [n]$, let X_i be the random variable of the i-th bit of X.

We have $\frac{|D^{[i]}|}{|D|} = \Pr[X_i = 1]$. By the sub-additivity of entropy, it follows that $\sum_{i=1}^{n} H(X_i) \geq H(X) \geq n - t$.

That is, $\mathbb{E}_{i \leftarrow [n]}[1 - H(X_i)] = \frac{t}{n}$. By a simple Markov's inequality, with probability at least $1 - \frac{2t}{n}$ over $i \leftarrow [n]$, we have $H(X_i) \geq 1/2$. Note that $H(X_i) \geq 1/2$ implies $\Pr[X_i = 1] \geq 0.1$.

Also,

$$I(X_i; f(X)) = H(X_i) - H(X_i|f(X)) \leq 1 - H(X_i|f(X)).$$

And by the sub-additivity of conditional entropy, we have

$$\sum_{i=1}^{n} H(X_i|f(X)) \geq H(X|f(X)) \geq n - t - 1.$$

Therefore,

$$\sum_{i=1}^{n} I(X_i; f(X)) \leq n - (n - t - 1) \leq t + 1,$$

or equivalently,

$$\mathop{\mathbb{E}}_{i \leftarrow [n]} I(X_i; f(X)) \leq \frac{t+1}{n}.$$

Note that by Fact 2.1,

$$I(X_i; f(X)) := \mathop{\mathbb{E}}_{x \sim X_i} D(f(X)_{X_i = x} || f(X)).$$

Taking expected values over i of both sides and using $\mathbb{E}_{i \leftarrow [n]} I(X_i; f(X)) \leq \frac{t+1}{n}$ gives

$$\mathop{\mathbb{E}}_{i \leftarrow [n]} I(X_i; f(X)) = \mathop{\mathbb{E}}_{i \leftarrow [n]} \mathop{\mathbb{E}}_{x \sim X_i} D(f(X)_{X_i = x} || f(X)) \leq \frac{t+1}{n}.$$

By Pinsker's inequality (Lemma 2.2) and the fact that $\sqrt{x}$ is a concave function, we have

$$\mathop{\mathbb{E}}_{i \leftarrow [n]} \mathop{\mathbb{E}}_{x \sim X_i} \|f(X)_{X_i = x} - f(X)\| \leq \sqrt{\frac{t+1}{n}},$$

and

$$\mathop{\mathbb{E}}_{i \leftarrow [n]} \Pr[X_i = 1] \cdot \|f(X)_{X_i = 1} - f(X)\| \leq \sqrt{\frac{t+1}{n}}.$$

Finally, note that with probability at least $1 - \frac{2t}{n}$ over $i \leftarrow [n]$, we have $\Pr[X_i = 1] \geq 0.1$. Putting everything together, we have

$$\mathop{\mathbb{E}}_{i \leftarrow [n]} \|f(X)_{X_i = 1} - f(X)\| \leq \frac{2t}{n} + 10 \cdot \sqrt{\frac{t+1}{n}} \leq O\left(\sqrt{\frac{t}{n}}\right). \qquad \square$$

Now we are ready to prove Lemma 4.3 (restated below).

Reminder of Lemma 4.3 *Let n, t, k be integers such that $t, k \leq n^{1/4}$ and $t \geq 10 \log n$, D be a subset of $\{0,1\}^n$ with $|D| \geq 2^{n-t}$, $\mathcal{U}_D$ be the uniform distribution on D, and $\mathcal{U}_D^C$ be the uniform distribution on $\{x : x \in D, x_i = 1 \text{ for all } i \in C\}$. For all function $f : D \to \{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{C \sim \mathcal{S}_k^{[n]}} [\|f(\mathcal{U}_D) - f(\mathcal{U}_D^C)\|] \leq O\left(k \cdot \sqrt{\frac{t}{n}}\right).$$

Proof of Lemma 4.3. Instead of choosing C from $\mathcal{S}_k^{[n]}$, we choose an ordered k-tuple of $a = (a_1, a_2, \ldots, a_k)$ of k distinct elements in $[n]$ uniformly at random. Let the distribution be $\mathcal{T}_k^{[n]}$.

We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}[\|f(\mathcal{U}_D) - f(\mathcal{U}_D^C)\|] &= \mathop{\mathbb{E}}_{a\sim\mathcal{T}_k^{[n]}}[\|f(\mathcal{U}_D) - f(\mathcal{U}_D^{\{a_i\}_{i=1}^k})\|] \\
&\le \sum_{\ell=1}^{k} \mathop{\mathbb{E}}_{a\sim\mathcal{T}_\ell^{[n]}}[\|f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell-1}}) - f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}})\|] \\
&\le \sum_{\ell=0}^{k-1} \mathop{\mathbb{E}}_{a\sim\mathcal{T}_\ell^{[n]}} \mathop{\mathbb{E}}_{j\leftarrow[n]\setminus\{a_i\}_{i=1}^{\ell}}[\|f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}}) - f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}\cup\{j\}})\|] \qquad (10)
\end{aligned}$$

Now we are to bound the right side of (10) for each $0 \le \ell \le k-1$ separately. For a subset $S \subseteq [n]$, let $D^S = \{x : x \in D \wedge x_i = 1 \text{ for all } i \in S\}$.

We first show with high probability, for $a \sim \mathcal{T}_\ell^{[n]}$, we have $D^{\{a_i\}_{i=1}^{\ell}}$ is very large. The proof of the following claim is deferred to end of the whole proof.

Claim 3. *For an integer $\ell \le n^{1/4}$,*

$$\Pr_{a\sim\mathcal{T}_\ell^{[n]}}[|D^{\{a_i\}_{i=1}^{\ell}}| \ge 2^{(n-\ell)-3t}] \ge 1 - O\left(\frac{t\cdot\ell}{n}\right).$$

Now, by Claim 3 and Lemma 4.4, with probability at least $1 - O\left(\frac{t\cdot\ell}{n}\right)$ over $a \sim \mathcal{T}_\ell^{[n]}$, we have

$$\mathop{\mathbb{E}}_{j\leftarrow[n]\setminus\{a_i\}_{i=1}^{\ell}}[\|f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}}) - f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}\cup\{j\}})\| \le O\left(\sqrt{\frac{3t}{n-\ell}}\right) = O\left(\sqrt{\frac{t}{n}}\right).$$

Putting them together, we have

$$\mathop{\mathbb{E}}_{a\sim\mathcal{T}_\ell^{[n]}} \mathop{\mathbb{E}}_{j\leftarrow[n]\setminus\{a_i\}_{i=1}^{\ell}}[\|f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}}) - f(\mathcal{U}_D^{\{a_i\}_{i=1}^{\ell}\cup\{j\}})\|] \le O\left(\frac{t\cdot\ell}{n} + \sqrt{\frac{t}{n}}\right)$$

Summing everything up for $0 \le \ell \le k-1$, we have

$$\mathop{\mathbb{E}}_{C\sim\mathcal{S}_k^{[n]}}[\|f(\mathcal{U}_D) - f(\mathcal{U}_D^C)\|] \le O\left(k^2\cdot\frac{t}{n} + k\sqrt{\frac{t}{n}}\right) = O\left(k\sqrt{\frac{t}{n}}\right).$$

□

Finally, we prove Claim 3, which is the most technical proof of this section.

Proof of Claim 3. We begin with some notations.

Subset Tree. We can view the process of choosing the k-tuples as growing a tree. For each $0 \le \ell \le k$ and each sequence $\{a_i\}_{i=1}^{\ell}$ from $\mathcal{T}_\ell^{[n]}$, we build a tree node $T_{\{a_i\}_{i=1}^{\ell}}$. For each $j \in [n] \setminus \{a_i\}_{i=1}^{\ell}$, we say node $T_{\{a_i\}_{i=1}^{\ell}\cup\{j\}}$ is a child of the node $T_{\{a_i\}_{i=1}^{\ell}}$, and denote the edge between them as $E_{\{a_i\}_{i=1}^{\ell}\to j}$. With this interpretation, the process of choosing $a \sim \mathcal{T}_k^{[n]}$ can be seen as starting from the root $T_\emptyset$, and descending to a random child for k times.

We also define

$$Z_{\{a_i\}_{i=1}^{\ell}} = (n-\ell) - \log_2 |D^{\{a_i\}_{i=1}^{\ell}}|,$$

and

$$Y_{\{a_i\}_{i=1}^{\ell}} = Z_{\{a_i\}_{i=1}^{\ell}} - Z_{\{a_i\}_{i=1}^{\ell-1}}.$$

That is, $Z_{\{a_i\}_{i=1}^{\ell}}$ is the gap between the entropy of the set corresponding to the node and the "full entropy" $n-\ell$, while $Y_{\{a_i\}_{i=1}^{\ell}}$ is the increase of that entropy gap on its parent.

Note that the claim asks to upper bound

$$\Pr_{a \sim T_{\ell}^{[n]}}[Z_{\{a_i\}_{i=1}^{\ell}} > 3t],$$

and we have

$$Z_{\emptyset} = t.$$

Good Nodes, Good Edges, Bad Nodes, Bad Edges, and Edge Labels. We next define when a node (or an edge) is good or bad. The root $T_{\emptyset}$ is a good node. If the parent of the node is a bad node then it is also a bad node. If a node is a bad node, then all edges in its sub-tree are bad edges.

If $T_{\{a_i\}_{i=1}^{\ell}}$ is a good node, we look at all $j \in [n] \setminus \{a_i\}_{i=1}^{\ell}$. We say the edge from $T_{\{a_i\}_{i=1}^{\ell}}$ to $T_{\{a_i\}_{i=1}^{\ell} \cup \{j\}}$, denoted as $E_{\{a_i\}_{i=1}^{\ell} \to j}$ is a good edge, if

$$H_{X \sim D^{\{a_i\}_{i=1}^{\ell}}}(X_j) \geq 0.9,$$

otherwise it is a bad edge. The above basically guarantees to us that a large enough subset (at least a constant fraction) of $D^{\{a_i\}_{i=1}^{\ell}}$ contains a 1 as its jth entry.

We mark $T_{\{a_i\}_{i=1}^{\ell} \cup \{j\}}$ as a bad node if $E_{\{a_i\}_{i=1}^{\ell} \to j}$ is a bad edge, or $Z_{\{a_i\}_{i=1}^{\ell} \cup \{j\}} > 3t$, otherwise it is a good node.

For a good edge $E_{\{a_i\}_{i=1}^{\ell} \to j}$, we say it has label k if $|Y_{\{a_i\}_{i=1}^{\ell} \cup \{j\}}| \in (2^{-k}, 2^{-k+1}]$.

Since it is an good edge, we have

$$H_{X \sim D^{\{a_i\}_{i=1}^{\ell}}}(X_j) \geq 0.9,$$

and by Fact 2.3, it follows

$$\Pr_{X \sim D^{\{a_i\}_{i=1}^{\ell}}}[X_j = 1] \geq 0.3.$$

Therefore,

$$|D^{\{a_i\}_{i=1}^{\ell} \cup \{j\}}| \geq 0.3 \cdot |D^{\{a_i\}_{i=1}^{\ell}}|.$$

Therefore, $Z^{\{a_i\}_{i=1}^{\ell} \cup \{j\}} \leq Z^{\{a_i\}_{i=1}^{\ell}} + \log(1/0.3) - 1 \leq Z^{\{a_i\}_{i=1}^{\ell}} + 1$, which means $Y_{\{a_i\}_{i=1}^{\ell} \cup \{j\}} \leq 1$. Hence, a good edge's label is at least 1.

Basic Facts. We need the following two basic facts, whose proof can be found at the end of the proof.

Fact 4.5. *Let $T_{\{a_i\}_{i=1}^{\ell}}$ be a good node, we have*

$$\Pr_{j \in [n] \setminus \{a_i\}_{i=1}^{\ell}}[E_{\{a_i\}_{i=1}^{\ell} \to j} \textit{ is good}] \geq 1 - O\left(\frac{t}{n}\right).$$

Fact 4.6. *Let $T_{\{a_i\}_{i=1}^{\ell}}$ be a good node and k be an integer, we have*

$$\Pr_{j \in [n] \setminus \{a_i\}_{i=1}^{\ell}}[E_{\{a_i\}_{i=1}^{\ell} \to j} \textit{ has label } k] \leq O\left(\frac{4^k \cdot t}{n}\right).$$

The Bound. Now we are going to lower bound the probability of the event that all nodes $T_{\{a_i\}_{i=1}^d}$ for $0 \le d \le \ell$ are good, denoted as event $\mathcal{E}_{\mathsf{good}}$. Clearly this provides a lower bound on $\Pr_{a \sim T_\ell^{[n]}}[Z_{\{a_i\}_{i=1}^\ell} > 2t]$.

Suppose $\mathcal{E}_{\mathsf{good}}$ doesn't happen, let d be the first index such that $T_{\{a_i\}_{i=1}^d}$ is a bad node, let this event be $\mathcal{E}_{\mathsf{bad}}^d$. Clearly we have

$$\Pr[\mathcal{E}_{\mathsf{good}}] = 1 - \sum_{d=0}^{\ell} \Pr[\mathcal{E}_{\mathsf{bad}}^d].$$

Therefore it suffices to provide an upper bound for each $\Pr[\mathcal{E}_{\mathsf{bad}}^d]$, note that $\mathcal{E}_{\mathsf{bad}}^d$ is defined as

$$\left[T_{\{a_i\}_{i=1}^j} \text{ is good for all } 0 \le j \le d-1 \text{ and } T_{\{a_i\}_{i=1}^d} \text{ is bad}\right].$$

There are two possible cases, the first case is that the edge $E_{\{a_i\}_{i=1}^{d-1} \to a_d}$ is an bad edge, which happens with probability at most $O\left(\frac{t}{n}\right)$ by Fact 4.5.

The second case is that the edge $E_{\{a_i\}_{i=1}^{d-1} \to a_d}$ is an good edge. In that case, by definition, we have $Z_{\{a_i\}_{i=1}^d} > 3t$, which also means

$$\sum_{j=1}^{d} Y_{\{a_i\}_{i=1}^j} > 2t.$$

Let N_k be the number of edges in the path $\{E_{\{a_i\}_{i=1}^{j-1} \to a_j} : j \in [d]\}$ with label k.

$$\sum_{k=1}^{\infty} N_k \cdot 2^{-k+1} > 2t,$$

which simplifies to

$$\sum_{k=1}^{\infty} N_k \cdot 2^{-k} > t.$$

Note that since $N_k \le d$, we have

$$\sum_{k=\log_2(2d/t)+1}^{\infty} N_k \cdot 2^{-k} \le d \cdot \frac{t}{2d} \le \frac{t}{2}.$$

Which means

$$\sum_{k=1}^{\log_2(2d/t)} N_k \cdot 2^{-k} > t/2.$$

In particular, this means there exists an $k \in [\log_2(2d/t)]$, such that

$$N_k \cdot 2^{-k} > \frac{t}{2\log n} \Rightarrow N_k \ge \frac{2^k \cdot t}{2\log n}.$$

Let the above be event $\mathcal{E}_{\mathsf{bad}}^{d,k}$, we have

$$\Pr[\mathcal{E}_{\mathsf{bad}}^d] \le \sum_{k=1}^{\log(2d/t)} \Pr[\mathcal{E}_{\mathsf{bad}}^{d,k}].$$

And by Fact 4.6, we have

$$\Pr[\mathcal{E}^{d,k}_{\mathsf{bad}}] \le O\left(\frac{4^k \cdot t}{n}\right)^{\frac{2^k \cdot t}{2\log n}} \cdot \binom{d}{\frac{2^k \cdot t}{2\log n}} \le O\left(\frac{4^k \cdot t}{n} \cdot d\right)^{\frac{2^k \cdot t}{2\log n}}.$$

Note that $4^k \le (2d/t)^2 = O(d^2/t^2)$, $d \le \ell \le n^{1/4}$ and $t \ge 10\log n$, the above simplifies to

$$\Pr[\mathcal{E}^{d,k}_{\mathsf{bad}}] \le O\left(\frac{d^3/t}{n}\right)^{\frac{2^k \cdot t}{2\log n}} \le n^{-1/4\cdot 10} \le n^{-2}.$$

Putting everything together, we have

$$\Pr[\mathcal{E}^{d}_{\mathsf{bad}}] \le \log n \cdot n^{-2} + O\left(\frac{t}{n}\right) = O\left(\frac{t}{n}\right),$$

and

$$\Pr[\mathcal{E}_{\mathsf{good}}] \ge 1 - \ell \cdot \left(\frac{t}{n}\right) \ge 1 - O\left(\frac{t\cdot\ell}{n}\right).$$

The above completes the proof. □

Now we finish the whole proof by proving Fact 4.5 and Fact 4.6.

Reminder of Fact 4.5 *Let $T_{\{a_i\}_{i=1}^{\ell}}$ be a good node. We have*

$$\Pr_{j\in[n]\setminus\{a_i\}_{i=1}^{\ell}}[E_{\{a_i\}_{i=1}^{\ell}\to j} \text{ is good}] \ge 1 - O\left(\frac{t}{n}\right).$$

Reminder of Fact 4.6 *Let $T_{\{a_i\}_{i=1}^{\ell}}$ be a good node and k be an integer. We have*

$$\Pr_{j\in[n]\setminus\{a_i\}_{i=1}^{\ell}}[E_{\{a_i\}_{i=1}^{\ell}\to j} \text{ has label } k] \le O\left(\frac{4^k \cdot t}{n}\right).$$

Proof of Fact 4.5 and Fact 4.6. Let $X \sim D^{\{a_i\}_{i=1}^{\ell}}$, we have $H(X) \ge n - \ell - 2t$. Also, since X_j for $j \in \{a_i\}_{i=1}^{\ell}$ is always 1, and therefore has entropy 0, by the sub-additive of entropy, we have

$$\sum_{j\in[n]\setminus\{a_i\}_{i=1}^{\ell}} H(X_j) \ge n - \ell - 2t.$$

Or equivalently, we have

$$\mathop{\mathbb{E}}_{j\in[n]\setminus\{a_i\}_{i=1}^{\ell}}(1 - H(X_j)) \le \frac{2t}{n-\ell} \le \frac{4t}{n}.$$

By a simple Markov's inequality, we have

$$\Pr_{j\in[n]\setminus\{a_i\}_{i=1}^{\ell}}[1 - H(X_j) \ge 0.1] \le O\left(\frac{t}{n}\right),$$

which proves Fact 4.5.

Now, let B_j be the set of j satisfying $H(X_j) < 0.9$. We have

$$\sum_{j \in [n] \setminus (\{a_i\}_{i=1}^{\ell} \cup B_j)} (1 - H(X_j)) \leq 2t.$$

Now, let $p_j = \Pr[X_j = 1]$ and $z_j = (p_j - 1/2)$. For $j \in [n] \setminus (\{a_i\}_{i=1}^{\ell} \cup B_j)$, we have $H(p_j) \geq 0.9$ and $|z_j| \leq 0.2$ from Fact 2.3, and also

$$\sum_{j \in [n] \setminus (\{a_i\}_{i=1}^{\ell} \cup B_j)} 2 \cdot z_j^2 \leq 2t. \tag{11}$$

Also, by definition, we have

$$Y_{\{a_i\}_{i=1}^{\ell} \cup \{j\}} = \log(1/p_j) - 1 = -\log(2p_j) = -2\log(1 + 2z_j).$$

Consider the function $g(z) := \log(1+z)/z$, we can see it is a decreasing function when $z \in [-0.4, 0.4]$, and we have

$$0.8 \leq g(0.4) \leq \frac{\log(1+z)}{z} \leq g(-0.4) \leq 1.3.$$

Therefore, if the edge $E_{\{a_i\}_{i=1}^{\ell} \to j}$ has label k, we know that

$$|2\log(1 + 2z_j)| \geq 2^{-k} \Rightarrow |6z_j| \geq 2^{-k} \Rightarrow z_j^2 \geq 4^{-k} \cdot \frac{1}{36}.$$

Using (11), we see there are at most $O\left(4^k \cdot t\right)$ j's such that $E_{\{a_i\}_{i=1}^{\ell} \to j}$ has label k. From which Fact 4.6 follows directly. □

5 Proof Overview For the PRG Construction

In this section we provide an overview of the proof for our PRG construction. We first consider a toy example: a very simple PRG which constructs one pseudo-random bit, and fools any one-round BCAST(1) protocol. Its proof already illustrates the key proof strategy which is used to prove our full PRG results. Then in Subsections 5.1 and 5.2 we sketch the key ideas to generalize the proof for the general PRG theorem.

The Toy PRG. Here we describe the PRG, and below we will analyze it to show it is indeed pseudo-random. Suppose there are n processors, and each processor receives k *truly random bits*. Suppose there is also a shared random bit-vector b of length k, which is also sampled uniformly at random. Then each processor's extra pseudo-random bit is the inner product (modulo 2) of the vector formed by its random bits and b (so the complete pseudo-random string is its initial k random bits concatenated with these extra random bits obtained with the inner product). Note that in the typical case $n \gg k$, this PRG generates n pseudo-random bits out of a shared random string b of length k. When analyzing the PRG, we think of b as a "secret" string, since distinguishing the PRG from true randomness corresponds to discovering whether such a b exists.

The goal here is to show that the above PRG construction and the case that all processor get $k + 1$ truly random bits are indistinguishable to a one-round BCAST(1) protocol (see Theorem 5.1 for a formal statement). We begin with some notations.

Notations. Throughout the paper, except when explicitly stated, all matrices and vectors are over $\mathbb{F}_2$. We use $\{0,1\}^n$ ($\{0,1\}^{n\times m}$) and $\mathbb{F}_2^n$ ($\mathbb{F}_2^{n\times m}$) interchangeably. For two vectors u and v, we use (u,v) to denote their concatenation.

Recall that we can assume all processors are deterministic as we are trying to prove a lower bound for distinguishing two input distributions by Yao's principle. Processor i can then be defined by a function $f_i : \{0,1\}^{k+1} \times \{0,1\}^* \to \{0,1\}$, such that $f_i(z,p)$ is the bit that player i outputs when it gets the input z and transcript p. We use $f_i^{|p}$ to denote the function $f_i(\cdot,p)$ for simplicity. If transcript p is incompatible with player i having input z, then we set $f_i(z,p)$ arbitrarily.

We use $\mathcal{U}_{[b]}$ to denote the uniform distribution on the set $\{(x, x\cdot b) : x \in \{0,1\}^k\}$, which is the distribution of inputs a processor receives when the shared random string during the construction of the pseudo-randomness is b. We can now formally state our theorem.

Theorem 5.1. *Let k be an integer and n be the number of processors. Consider the following two cases:*

- *(A) All processors receive random inputs from $\mathcal{U}_{k+1}$.*
- *(B) Let b be a uniform sample from $\mathcal{U}_k$, then all processors receive inputs from $\mathcal{U}_{[b]}$.*

For any one-round BCAST(1) protocol, the statistical distance between the distributions of its transcripts in case (A) and (B) is at most $O\left(\frac{n}{2^{k/2}}\right)$.

We need the following technical lemma, whose proof can be found at the end of this section.

Lemma 5.2. *Given a function $f : \{0,1\}^{k+1} \to \{0,1\}$, we have*

$$\sum_{b\in\{0,1\}^k} \|f(\mathcal{U}_{k+1}) - f(\mathcal{U}_{[b]})\|^2 \le \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)] \le 1.$$

Note that $f(\mathcal{U}_{k+1})$ and $f(\mathcal{U}_{[b]})$ are two distributions on $\{0,1\}$.

Intuitively, the above lemma says that for any function f (think of this as a function describing a processor), it cannot distinguish distributions $\mathcal{U}_{[b]}$ and $\mathcal{U}_{k+1}$ for most strings b. So, fixing a few random entries of x to be 1 doesn't change the probability that $f(x)$ is 1 by much.

Now we are ready to prove Theorem 5.1.

Proof. Instead of viewing the algorithm as a single round algorithm, we will prove a slightly stronger lower bound. Consider the model where we have n turns. On the t^{th} turn, processor t gets to send a single bit. This model is stronger than one round of the BCAST(1) model, since it allows the later processors to condition their outputs on earlier the processors' messages. Hence, our lower bound implies a lower bound for the BCAST(1) model as well.

Notations. Let $\mathcal{P}^{(t)}_{\mathsf{rand}}$ and $\mathcal{P}^{(t)}_{[b]}$ be the distributions of the transcript of the first t turns when all processors get a random input from $\mathcal{U}_{k+1}$ and $\mathcal{U}_{[b]}$ respectively.

Note that to prove the theorem, it suffices to show that the distribution $\mathcal{P}^{(n)}_{\mathsf{rand}}$ is close to $\mathcal{P}^{(n)}_{[b]}$ for most choices of b. For this purpose, we are going to prove the following inequality holds for any $t \le n$:

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\|\right] \le t\cdot 2^{-k/2}. \tag{12}$$

It is easy to see that plugging in $t = n$, (12) implies the theorem. To prove (12) for all t, we induct on t. Clearly, (7) holds when $t = 0$. So it suffices to show that when it holds for $t-1$, it also holds for t.

For $b \in \{0,1\}^k$, we wish to bound the distance $\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\|$. By Lemma 1.9, it follows that

$$\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\| \le \|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{[b]}\| + \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f_t^{|p}(\mathcal{U}_{k+1}) - f_t^{|p}(\mathcal{U}_{[b]})\right\|\right]. \tag{13}$$

Recall that in above $f_t^{|p}$ is the output function of process t when seeing the transcript p.
For each $b \in \{0,1\}^k$ and transcript $p \in \{0,1\}^{t-1}$, we define scores $s_{b,p}$ and s_b as follows:

$$s_{b,p} := \left\|f_t^{|p}(\mathcal{U}_{k+1}) - f_t^{|p}(\mathcal{U}_{[b]})\right\| \quad \text{and} \quad s_b := \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}[s_{b,p}].$$

It suffices to give an upper bound on $\mathbb{E}_{b\sim\mathcal{U}_k}[s_b]$. By Lemma 5.2, for all $p \in \{0,1\}^{t-1}$, we have

$$\sum_{b\in\{0,1\}^k} s_{b,p}^2 \le 1,$$

and therefore

$$\sum_{b\in\{0,1\}^k} s_{b,p} \le 2^{k/2} \quad \text{and} \quad \mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_{b,p}] \le 2^{-k/2}.$$

By the definition of s_b, it follows that

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_b] \le 2^{-k/2}.$$

Therefore, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\|\right] \le& \mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{[b]}\| + \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f_t^{|p}(\mathcal{U}_{k+1}) - f_t^{|p}(\mathcal{U}_{[b]})\right\|\right]\right] \\
\le& (t-1)\cdot 2^{-k/2} + \mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_b] \\
\le& t\cdot 2^{-k/2}.
\end{aligned}$$

The above proves (12) for t, which completes the whole proof.

□

Finally, we prove Lemma 5.2.

Proof of Lemma 5.2. Note that since f is Boolean valued, we have

$$\|f(\mathcal{U}_{k+1}) - f(\mathcal{U}_{[b]})\| = \left|\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[f(x)]\right|.$$

The proof is an application of the analysis of Boolean functions (see Section 2.2). Let $b \in \{0,1\}^k$, and let S_b be the corresponding subset of $[k]$ (if $b_i = 1$ then $i \in S_b$). We use $\overline{\mathcal{U}}_{[b]}$ to denote the uniform distribution on the set $\{(x, 1 - x\cdot b) : x \in \{0,1\}^b\}$, that is, the uniform distribution on the complement of the support of $\mathcal{U}_{[b]}$.

Note that for every x from the support of $\mathcal{U}_{[b]}$, we have $x \cdot (b,1) = 0$, and for every x from the support of $\overline{\mathcal{U}}_{[b]}$, we have $x \cdot (b,1) = 1$. Then we have

$$\begin{aligned}
\widehat{f}(S_b \cup \{k+1\}) :=& \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}\left[f(x)\cdot(-1)^{(b,1)\cdot x}\right].\\
=&\frac{1}{2}\cdot\left(\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\overline{\mathcal{U}}_{[b]}}[f(x)]\right)\\
=&\frac{1}{2}\cdot\left(2\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\overline{\mathcal{U}}_{[b]}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[f(x)]\right)\\
=& \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)].
\end{aligned}$$

By Parseval's identity (see Section 2.2) and the fact that f is Boolean valued, we have

$$\sum_{b\in\{0,1\}^k}\widehat{f}(S_b\cap\{k+1\})^2 \le \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)^2] = \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)],$$

and it follows that

$$\sum_{b\in\{0,1\}^k}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)]\right)^2 \le \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[f(x)] \le 1. \qquad \square$$

5.1 Generalization to Multi-Round Case

Now we outline how to extend the proof to the multi-round case. The PRG is still the same as the toy PRG, we just need to prove it also fools multiple round BCAST(1) protocols (i.e. Theorem 5.3). In the following, we will explain the key difficulty for generalizing the previous proof to the multi-round case, and how we address them.

Theorem 5.3. *Consider the following two cases:*

- *(A) All processors receive random inputs from $\mathcal{U}_{k+1}$.*
- *(B) Let b be a uniform sample from $\mathcal{U}_k$, then all processors receive inputs from $\mathcal{U}_{[b]}$.*

For $j \le k/10$, and any j-round BCAST(1) protocol, the statistical distance between the distributions of its transcripts in case (A) and (B) is at most $O\left(\frac{j\cdot n}{2^{k/9}}\right)$.

The key technical part of the proof of Theorem 5.1, is to bound $\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\|$, i.e., the Inequality (13):

$$\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\| \le \|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{[b]}\| + \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f_t^{|p}(\mathcal{U}_{k+1}) - f_t^{|p}(\mathcal{U}_{[b]})\right\|\right].$$

Let X_{rand} ($X_{[b]}$) denote the random variable for the input to the processor i broadcasting at the t^{th} turn, in the case all processors receive inputs from $\mathcal{U}_{k+1}$ ($\mathcal{U}_{[b]}$). Inequality (13) holds crucially because X_{rand} ($X_{[b]}$) is independent of the previous part of the transcript during the first $(t-1)$ turns (b is fixed).

However, the independence condition no longer holds in the multi-round case, as the transcript contains previous broadcasts of the *same* processor i, which contain information about processor i's input. To deal

with that, we have to consider the conditional random variables $X_{\mathsf{rand}}^{|p}$ and $X_{[b]}^{|p}$ which are X_{rand} and $X_{[b]}$ conditioning on seeing the transcript p.

Let $D_p^{(t-1)}$ denote the set of inputs to f_i which are consistent with the transcript $p \in \{0,1\}^{t-1}$,[12] then $X_{\mathsf{rand}}^{|p}$ and $X_{[b]}^{|p}$ distribute uniformly on $\{0,1\}^{k+1} \cap D_p^{(t-1)}$ and $\{(x, x \cdot b) : x \in \{0,1\}^k\} \cap D_p^{(t-1)}$. We use $\mathcal{U}_{k+1,p}$ and $\mathcal{U}_{[b],p}$ to denote their distributions. Then we can state a bound similar to (13) in the multi-round case:

$$\|\mathcal{P}_{\mathsf{rand}}^{(t)} - \mathcal{P}_{[b]}^{(t)}\| \leq \|\mathcal{P}_{\mathsf{rand}}^{(t-1)} - \mathcal{P}_{[b]}^{(t-1)}\| + \mathop{\mathbb{E}}_{p \sim \mathcal{P}_{\mathsf{rand}}^{(t-1)}} \left[\left\| f_t^{|p}(\mathcal{U}_{k+1,p}) - f_t^{|p}(\mathcal{U}_{[b],p}) \right\| \right].$$

Our one-round proof depends on Lemma 5.2, which cannot be used directly to bound the right side of the above inequality. Luckily, we are able to generalize Lemma 5.2 such that it works as long as $D_p^{(t-1)}$ is sufficiently large (see Lemma 6.1), which happens to be the case with high probability (see Claim 4).

5.2 Generalization to the Complete PRG

Before discussing how to generalize the proof to get a complete PRG. We give a formal description of the PRG here.

The Full PRG. Suppose there are n processors and the PRG wants to create m pseudo-random bits that fool an $\Omega(k)$-round $\mathsf{BCAST}(1)$ protocol. Then the PRG is described as follows: each processor gets k *truly random bits*. There is also a hidden "secret" matrix M of size $k \times (m-k)$, which distributes uniformly random (when constructing the pseudo-randomness, this matrix is created by having each processor broadcast some additional uniformly random bits until there are enough to create the matrix). Then each processor's extra $m-k$ pseudo-random bits are simply the vector matrix product of its random bits and M, i.e., $x^T M$ (see also Theorem 1.3).

The generalization to the complete PRG case (Theorem 5.4) is quite technical. To state the whole technical theorem, we need to introduce some definitions. Let $M \in \{0,1\}^{n \times m}$. We use $\mathcal{U}_M$ to denote the uniform distribution on the following set $\{(x, x^T M) : x \in \{0,1\}^n\}$, which is a subset of $\{0,1\}^{n+m}$. For integers n and m, we use $\mathcal{U}_{n \times m}$ to denote the uniform distribution on $\{0,1\}^{n \times m}$. Formally, we want to show:

Theorem 5.4. *Let n, m, k be three integers. Consider the following two cases:*

- *(A) All processors receive random inputs from $\mathcal{U}_m$.*
- *(B) Let M be a uniform sample from $\mathcal{U}_{k \times (m-k)}$, then all processors receive inputs from $\mathcal{U}_M$.*

For $j \leq k/10$, $m \leq 2^{k/20}$ and any j-round $\mathsf{BCAST}(1)$ protocol, the statistical distance between the distributions of its transcripts in case (A) and (B) is at most $O\left(\frac{j \cdot n}{2^{k/9}}\right)$.

The proof strategy is still similar to that of Theorem 5.3. But now for each turn t, we need to maintain a set $S^{(t)}$ of secret matrices $M \in \{0,1\}^{k \times (m-k)}$ instead of a set of secret strings. Following the same reasoning as in the previous subsection, we can state a similar bound in this case:

$$\|\mathcal{P}_{\mathsf{rand}}^{(t)} - P_M^{(t)}\| \leq \|\mathcal{P}_{\mathsf{rand}}^{(t-1)} - P_M^{(t-1)}\| + \mathop{\mathbb{E}}_{p \sim \mathcal{P}_{\mathsf{rand}}^{(t-1)}} \left[\left\| f_t^{|p}(\mathcal{U}_{m,p}) - f_t^{|p}(\mathcal{U}_{M,p}) \right\| \right].$$

In which we use $P_M^{(t)}$ to denote the distribution of the transcript of the first t rounds when all processors get random input from $\mathcal{U}_M$. And $\mathcal{U}_{m,p}$ and $\mathcal{U}_{M,p}$ are distributions to the current processor i conditioning on seeing transcript p. Using a clever hybrid argument, we are able to prove the sufficient technical lemma (Lemma 7.2) to bound the right side of the above inequality.

[12] That is, simulating f_i with transcript p on that input results in transcript p itself.

6 Creating a Single Extra Pseudo-random Bit And an Average Case Lower Bound

In this section we show our toy PRG (see Section 5) also fools multiple rounds $\mathsf{BCAST}(1)$ protocols by proving Theorem 5.3 (restated below). We also show that our average case lower bound (Theorem 1.4) is a simple corollary of it.

Reminder of Theorem 5.3. *Consider the following two cases:*

- *(A) All processors receive random inputs from $\mathcal{U}_{k+1}$.*
- *(B) Let b be a uniform sample from $\mathcal{U}_k$, then all processors receive inputs from $\mathcal{U}_{[b]}$.*[13]

For $j \leq k/10$, and any j-round $\mathsf{BCAST}(1)$ protocol, the statistical distance between the distributions of its transcripts in case (A) and (B) is at most $O\left(\frac{j \cdot n}{2^{k/9}}\right)$.

6.1 An Average Case Lower Bound for $\mathsf{BCAST}(1)$

First, we show Theorem 5.3 implies the average case lower bound we want.

Reminder of Theorem 1.4 *Let n be a large enough integer and $F_{\mathsf{full\text{-}rank}} : \{0,1\}^{n \times n} \to \{0,1\}$ be the indicator function that whether the given matrix has full rank. Suppose there are n processors, i-th processor is given with the i-th row of the input matrix. For all $n/20$-round $\mathsf{BCAST}(1)$ protocol and all processor i in it, i cannot compute F correctly with probability better than 0.99, over a uniform random matrix from $\{0,1\}^{n \times n}$.*

Proof. Let M be the input matrix from $\mathcal{U}_{n \times n}$, where processor i gets its i-th row. Let $\mathcal{U}_A$ be the uniform distribution $\mathcal{U}_{n \times n}$, and $\mathcal{U}_B$ be the input distribution of case (B) in Theorem 5.3 when setting $k = n - 1$.

We need some results about random $\mathbb{F}_2$ matrix from Section 3.2 of [Kol99]. In particular, let $P_{n,s}$ be the probability that a uniformly random $\mathbb{F}_2$ matrix from $\mathbb{F}_2^{n \times n}$ has rank $n - s$. For all s, we have

$$\lim_{n \to \infty} P_{n,s} = Q_s := 2^{-s^2} \cdot \left(\prod_{i \geq s+1} (1 - 2^{-i}) \right) \cdot \left(\prod_{1 \leq i \leq s} (1 - 2^{-i})^{-1} \right).$$

Numerically, we have $Q_0 \approx 0.2887880950866$. Let i be a processor, and $\mathsf{acc}(M)$ be i's output on input matrix M, it suffices to show that $\mathsf{acc}(M)$ can not be correct w.r.t. $F_{\mathsf{full\text{-}rank}}$ with probability higher than 0.99. Set $\varepsilon = 1 - 0.99 = 0.01$ for convenience.

Suppose for the contradiction that $\mathsf{acc}(M)$ is correct w.r.t. $F_{\mathsf{full\text{-}rank}}(M)$ with probability at least $1 - \varepsilon$ over $M \sim \mathcal{U}_A$, then we have

$$\left| \mathop{\mathbb{E}}_{M \sim \mathcal{U}_A} [\mathsf{acc}(M)] - \mathop{\mathbb{E}}_{M \sim \mathcal{U}_A} [F_{\mathsf{full\text{-}rank}}(M)] \right| \leq \varepsilon,$$

which means

$$\left| \mathop{\mathbb{E}}_{M \sim \mathcal{U}_A} [\mathsf{acc}(M)] - Q_0 \right| \leq \varepsilon + o(1).$$

[13] recall that $\mathcal{U}_{[b]}$ denotes the uniform distribution on the set $\{(x, x \cdot b) : x \in \{0,1\}^k\}$

Also, by Theorem 5.3, we have

$$\left| \mathop{\mathbb{E}}_{M\sim\mathcal{U}_A}[\mathsf{acc}(M)] - \mathop{\mathbb{E}}_{M\sim\mathcal{U}_B}[\mathsf{acc}(M)] \right| = o(1),$$

and therefore

$$\left| \mathop{\mathbb{E}}_{M\sim\mathcal{U}_B}[\mathsf{acc}(M)] - Q_0 \right| \le \varepsilon + o(1).$$

However, for all matrices in the support of $\mathcal{U}_B$, their rank is at most $n-1$, which means $\mathsf{acc}(M)$ must be wrong on most of them. Note that

$$\Pr_{M\sim\mathcal{U}_A}[\mathsf{acc}(M) \neq F_{\mathsf{full\text{-}rank}}(M)] \ge \mathop{\mathbb{E}}_{M\sim\mathcal{U}_B}\left[[\mathsf{acc}(M) \neq F_{\mathsf{full\text{-}rank}}(M)] \cdot \frac{\mathcal{U}_A(M)}{\mathcal{U}_B(M)} \right],$$

where $\mathcal{U}_A(M)$ and $\mathcal{U}_B(M)$ denote the probability of getting M for distributions $\mathcal{U}_A$ and $\mathcal{U}_B$.

For a matrix $M \sim \mathcal{U}_B$, suppose the rank of its first $n-1$ columns is $n-s$, $\mathcal{U}_B(M)$ can be computed as

$$\mathcal{U}_B(M) = 2^{-n(n-1)} \cdot 2^{-(n-s)} = 2^{-n^2} \cdot 2^s.$$

Furthermore, for $M \sim \mathcal{U}_B$, the probability that its first $n-1$ columns have rank at least $n-s$ is at least $\sum_{j=0}^{s-1} P_{n-1,j}$, as the rank of an $n\times(n-1)$ matrix is always no less than the rank of its left-top $(n-1)\times(n-1)$ matrix. Setting $s=3$, we can see that for large enough n, with probability at least $\sum_{j=0}^{2} Q_j \ge 1 - 0.006$, the first $n-1$ columns of M have rank at least $n-3$. In that case, $\frac{\mathcal{U}_A(M)}{\mathcal{U}_B(M)} \ge 2^{-s} = 1/8$.

Putting them together, we have

$$\begin{aligned}
&\Pr_{M\sim\mathcal{U}_A}[\mathsf{acc}(M) \neq F_{\mathsf{full\text{-}rank}}(M)] \\
\ge &\mathop{\mathbb{E}}_{M\sim\mathcal{U}_B}\left[[\mathsf{acc}(M) \neq F_{\mathsf{full\text{-}rank}}(M)] \cdot \frac{\mathcal{U}_A(M)}{\mathcal{U}_B(M)} \right] \\
\ge &\mathop{\mathbb{E}}_{M\sim\mathcal{U}_B}\left[[\mathsf{acc}(M) \neq F_{\mathsf{full\text{-}rank}}(M)] \cdot [\text{the first } n-1 \text{ columns of } M \text{ have rank at least } n-3] \right] \cdot \frac{1}{8} \\
\ge &(1 - Q_0 - \varepsilon - o(1) - 1 - 0.006) \cdot \frac{1}{8} > 0.05,
\end{aligned}$$

contradiction, which completes the proof. □

Considering the problem that checking whether the top $k \times k$ sub-matrix has full-rank, one immediately get the following average case time-hierarchy theorem for $\mathsf{BCAST}(1)$.

Reminder of Theorem 1.5 *For any $\omega(\log n) \le k \le n$, there is a function F such that a k-round $\mathsf{BCAST}(1)$ protocol can compute exactly, while any $k/20$-round $\mathsf{BCAST}(1)$ protocols cannot compute F correctly with probability 0.99 over the uniform distribution.*

6.2 Proof of Theorem 5.3

We need the following technical lemma first, whose proof is deferred to the end of the section.

Lemma 6.1. *Given a function $f : \{0,1\}^{k+1} \to \{0,1\}$ and a set $D \subseteq \{0,1\}^{k+1}$ with $|D| \geq 2^{k/2}$, let $\mathcal{U}_{[b],D}$ and $\mathcal{U}_{k+1,D}$ be the conditional distributions of $\mathcal{U}_{[b]}$ and $\mathcal{U}_{k+1}$ on the set D.*[14] *We have*

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k} \|f(\mathcal{U}_{[b],D}) - f(\mathcal{U}_{k+1,D})\| \leq 2^{-k/9}.$$

Now we are ready to prove Theorem 5.3.

Proof of Theorem 5.3. Similar to the proof of Theorem 5.1, we will consider a slightly stronger model where we have $j \cdot n$ turns, and on the t^{th} turn, processor $(t-1) \bmod n + 1$ gets to send a single bit. Recall that we use $\mathcal{P}^{(t)}_{\mathsf{rand}}$ to denote the distribution of the transcript of the first t rounds when all processors get random input from $\mathcal{U}_{k+1}$, and $\mathcal{P}^{(t)}_{[b]}$ to denote the distribution of the transcript of the first t rounds when all processors get random input from $\mathcal{U}_{[b]}$.

Note that to prove the theorem, it suffices to show that the distribution $\mathcal{P}^{(j\cdot n)}_{\mathsf{rand}}$ is close to $\mathcal{P}^{(j\cdot n)}_{[b]}$ for most choices of b. For this purpose, we are going to prove the following inequality holds for any $t \leq j \cdot n$:

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\|\right] \leq 2 \cdot t \cdot 2^{-k/9}. \tag{14}$$

It is easy to see that plugging in $t = j \cdot n$, (14) implies the theorem. To prove (14) for all t, we induct on t. Clearly, (7) holds when $t = 0$. So it suffices to show that when it holds for $t-1$, it also holds for t. Let $i = (t-1) \pmod n + 1$ be the processor broadcasting at the t-th turn.

Again, for $b \in \{0,1\}^k$, we wish to bound $\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\|$. By Lemma 1.9, we have

$$\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_{[b]}\| \leq \|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_{[b]}\| + \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f_i^{|p}(\mathcal{U}_{k+1,p}) - f_i^{|p}(\mathcal{U}_{[b],p})\right\|\right]. \tag{15}$$

In the above, $\mathcal{U}_{k+1,p}$ and $\mathcal{U}_{[b],p}$ are the distributions $\mathcal{U}_{k+1}$ and $\mathcal{U}_{[b]}$ conditioned on the previous transcript p. Specifically, let $D_p^{(t-1)}$ denote the set of inputs to f_i which are consistent with the transcript p[15], then $\mathcal{U}_{[b],p}$ and $\mathcal{U}_{k+1,p}$ are the conditional distributions of $\mathcal{U}_{[b]}$ and $\mathcal{U}_{k+1}$ on set $D_p^{(t-1)}$.

Here, we wish to use Lemma 6.1 to bound the second term on the right side of (15). To satisfy the requirement of Lemma 6.1, we have to show that $D_p^{(t-1)}$ is a large subset of $\{0,1\}^{k+1}$ with high probability. Hence, we need the following claim, whose proof is deferred until we prove the theorem first.

Claim 4. *For $t \leq j \cdot n \leq \frac{k\cdot n}{10}$, with probability $1 - 2^{-k/4}$ over $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$, we have $|D_p^{(t-1)}| \geq 2^{k/2}$.*

Now, for each b, we define a score s_b as follows:

$$s_b := \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f_i^{|p}(\mathcal{U}_{k+1,p}) - f_i^{|p}(\mathcal{U}_{[b],p})\right\|\right].$$

For $p \in \{0,1\}^{t-1}$, we set $s_{b,p} := \left\|f_i^{|p}(\mathcal{U}_{k+1,p}) - f_i^{|p}(\mathcal{U}_{[b],p})\right\|$. Then by Lemma 6.1 and Claim 4, when $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$, with probability at least $1 - 2^{-k/4}$, we have $|D_p^{(t-1)}| \geq 2^{k/2}$ and

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_{b,p}] \leq 2^{-k/9}.$$

Therefore, it follows

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_b] = \mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_{b,p}] \leq 2^{-k/4} + 2^{-k/9} \leq 2 \cdot 2^{-k/9}.$$

[14] When $\mathcal{U}_{[b]}$ ($\mathcal{U}_{k+1}$) has no mass on D, we set $\mathcal{U}_{[b],D}$ ($\mathcal{U}_{k+1,D}$) to be the uniform distribution on D.

[15] That is, simulating f_i with transcript p on that input results in transcript p itself.

Now we have

$$\begin{aligned}\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\|\mathcal{P}^{(t)}_{\mathsf{rand}}-\mathcal{P}^{(t)}_{[b]}\|\right]\leq&\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\|\mathcal{P}^{(t-1)}_{\mathsf{rand}}-\mathcal{P}^{(t-1)}_{[b]}\|+\mathop{\mathbb{E}}_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[\left\|f_i^{|p}(\mathcal{U}_{k+1,p})-f_i^{|p}(\mathcal{U}_{[b],p})\right\|\right]\right]\\\leq&2\cdot(t-1)\cdot2^{-k/9}+\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}[s_b]\\\leq&2\cdot t\cdot2^{-k/9}.\end{aligned}$$

The above proves (14) for t, which completes the whole proof.

□

Now we prove Claim 4.

Proof of Claim 4. Let $t_1,t_2,\ldots,t_\ell$ be the indices of all previous ℓ turns with processor i broadcasting, before the current t-th turn. We have $\ell\leq j\leq k/10$. Let $x\in\{0,1\}^{k+1}$, note that x is consistent with transcript p, if for all $a\in[\ell]$, we have

$$f_i^{|p^{(t_a-1)}}(x)=p_{t_a},$$

where $p^{(t_a-1)}$ denotes the first t_a-1 bits of p. We set $F_i(x,p)=1$ if x and p are consistent, and 0 otherwise.

Consider the random process of generating $p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}$, suppose inputs to all processors other than i are fixed, let $x^{-i}=(x_1,x_2,\ldots,x_{i-1},x_{i+1},\ldots,x_n)\in\{0,1\}^{(n-1)\times(k+1)}$ be those fixed input. Let $P^{(t)}_{x^{-i}}$ be the distribution of the transcript when $x_i\sim\mathcal{U}_{k+1}$, and all other processors get (fixed) input according to x^{-i}.

For a fixed x^{-i}, note that there are only 2^ℓ possible transcripts p from $P^{(t-1)}_{x^{-i}}$, as the transcript is determined after fixing the output of processor i at all ℓ rounds. Therefore, let $T(x^{-i},x_i)$ be the transcript when all processors get inputs according to x^{-i} and x_i, we can see when $p\sim P^{(t-1)}_{x^{-i}}$, $F_i(x,p)=1$ if and only if $T(x^{-i},x)=p$. That is,

$$P^{(t-1)}_{x^{-i}}(p)=\Pr_{x_i\sim\mathcal{U}_{k+1}}[T(x^{-i},x_i)=p]=D^{(t-1)}_p/2^{k+1}.$$

In above $P^{(t-1)}_{x^{-i}}(p)$ is the probability that getting p from distribution $P^{(t-1)}_{x^{-i}}$. Then we have

$$\begin{aligned}&\Pr_{p\sim P^{(t-1)}_{x^{-i}}}\left[D^{(t-1)}_p<2^{-\ell-k/4}\cdot2^{k+1}\right]\\=&\Pr_{p\sim P^{(t-1)}_{x^{-i}}}\left[P^{(t-1)}_{x^{-i}}(p)<2^{-\ell-k/4}\right]\\\leq&2^{-\ell-k/4}\cdot2^\ell=2^{-k/4}.\end{aligned}$$

The last inequality holds because the support size of $P^{(t-1)}_{x^{-i}}$ is at most 2^ℓ.

Hence, we have

$$\begin{aligned}\Pr_{p\sim\mathcal{P}^{(t-1)}_{\mathsf{rand}}}\left[D^{(t-1)}_p<2^{-\ell-k/4}\cdot2^{k+1}\right]=&\mathop{\mathbb{E}}_{x^{-i}\sim\mathcal{U}_{(n-1)\times(k+1)}}\left[\Pr_{p\sim P^{(t-1)}_{x^{-i}}}\left[D^{(t-1)}_p<2^{-\ell-k/4}\cdot2^{k+1}\right]\right]\\\leq&2^{-k/4}.\end{aligned}$$

The claim follows from that $\ell\leq k/10$. □

6.3 Proof of Lemma 6.1

Here we prove Lemma 6.1.

Proof of Lemma 6.1. Let g be the following function

$$g(x) := \begin{cases} f(x) & x \in D \\ 0 & x \notin D. \end{cases}$$

Let $N_D := |D|$, and for $b \in \{0,1\}^k$, let $D_{[b]}$ be the support set of $\mathcal{U}_{[b]}$ and $N_b := |D \cap D_{[b]}|$. That is, N_D is the size of the support set of $\mathcal{U}_{k+1,D}$, while N_b is the size of the support size of $\mathcal{U}_{b,D}$.

We need the following claim, which shows that for most b's N_b is close of a half of N_D. We defer its proof until we prove the lemma.

Claim 5. *Let $b \sim \mathcal{U}_k$, with probability $1 - 2^{-k/8}$, we have $|N_b/N_D - \frac{1}{2}| < 2^{-k/8}$.*

By Lemma 5.2, we have

$$\sum_{b \in \{0,1\}^k} \|g(\mathcal{U}_{[b]}) - g(\mathcal{U}_{k+1})\|^2 \le \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1}}[g(x)] \le N_D/2^{k+1} \le N_D/2^k.$$

Equivalently,

$$\mathop{\mathbb{E}}_{b \sim \mathcal{U}_k}\left[\|g(\mathcal{U}_{[b]}) - g(\mathcal{U}_{k+1})\|^2\right] = \mathop{\mathbb{E}}_{b \sim \mathcal{U}_k}\left[\left|\mathop{\mathbb{E}}_{x \sim \mathcal{U}_{[b]}}[g(x)] - \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1}}[g(x)]\right|^2\right] \le N_D/2^{2k}. \tag{16}$$

In order to make use of the above bound (16), we now relate $g(\mathcal{U}_{[b]})$ and $g(\mathcal{U}_{k+1})$ to $f(\mathcal{U}_{[b],D})$ and $f(\mathcal{U}_{k+1,D})$. From the definition of g, we have

$$\begin{aligned} \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{[b],D}}[f(x)] &= \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{[b]}}[g(x)] \cdot \frac{2^k}{N_b} \\ &= \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{[b]}}[g(x)] \cdot \frac{2^k}{N_D/2} \cdot \frac{N_D}{2N_b}, \end{aligned} \tag{17}$$

and

$$\begin{aligned} \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1,D}}[f(x)] &= \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1}}[g(x)] \cdot \frac{2^{k+1}}{N_D} \\ &= \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1}}[g(x)] \cdot \frac{2^k}{N_D/2}. \end{aligned}$$

That is, $\mathbb{E}_{x \sim \mathcal{U}_{[b],D}}[f(x)]$ and $\mathbb{E}_{x \sim \mathcal{U}_{k+1,D}}[f(x)]$ are $\mathbb{E}_{x \sim \mathcal{U}_{[b]}}[g(x)]$ and $\mathbb{E}_{x \sim \mathcal{U}_{k+1}}[g(x)]$ scaled by a factor of $\frac{2^k}{N_D/2}$, except for another $\frac{N_D}{2N_b}$ factor in (17), which is very close to 1 for most b's by Claim 5.

We first ignore the $\frac{N_D}{2N_b}$ factor in (17), and define

$$F_{[b]} := \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{[b]}}[g(x)] \cdot \frac{2^k}{N_D/2},$$

and

$$F_{k+1} := \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1,D}}[f(x)] = \mathop{\mathbb{E}}_{x \sim \mathcal{U}_{k+1}}[g(x)] \cdot \frac{2^k}{N_D/2}.$$

Scaling each side of (16) by $\left(\frac{2^k}{N_D/2}\right)^2$, we have

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[|F_{[b]} - F_{k+1}|^2\right] \le N_D/2^{2k} \cdot \left(\frac{2^k}{N_D/2}\right)^2 = 4/N_D \le 4 \cdot 2^{-k/2},$$

the last inequality follows from the assumption that $N_D = |D| \ge 2^{-k/2}$.

That is, by Markov's inequality, when $b \sim \mathcal{U}_k$, with probability $1 - 2^{-k/8}$, we have

$$|F_{[b]} - F_{k+1}|^2 \le 4 \cdot 2^{-k/2} \cdot 2^{k/8} < 2^{-k/4},$$

which means $|F_{[b]} - F_{k+1}| < 2^{-k/8}$.

Now we take care of the additional $\frac{N_D}{2N_b}$ factor in (17). By Claim 5, when $b \sim \mathcal{U}_k$, with probability $1 - 2^{-k/8}$, we have $\left|N_b/N_D - \frac{1}{2}\right| < 2^{-k/8}$, which means $\left|\frac{N_D}{2N_b} - 1\right| < 3 \cdot 2^{-k/8}$.

Putting everything together, when $b \sim \mathcal{U}_k$, with probability $1 - 2 \cdot 2^{-k/8}$, we have

$$\begin{aligned}
\|f(\mathcal{U}_{[b],D}) - f(\mathcal{U}_{k+1,D})\| &= \left|\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b],D}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1,D}}[f(x)]\right| \\
&= \left|F_{[b]} \cdot \frac{N_D}{2N_b} - F_{k+1}\right| \\
&\le \left|F_{[b]} - F_{k+1}\right| + F_{[b]} \cdot \left|\frac{N_D}{2N_b} - 1\right| \\
&\le 4 \cdot 2^{-k/8}.
\end{aligned}$$

The last line follows from that $F_{[b]} = \mathbb{E}_{x\sim\mathcal{U}_{[b]}}[g(x)] \cdot \frac{2^{k+1}}{N_D} \le \frac{N_D}{2^{k+1}} \cdot \frac{2^{k+1}}{N_D} = 1$.

Therefore,

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\|f(\mathcal{U}_{[b],D}) - f(\mathcal{U}_{k+1,D})\| \le (2 \cdot 2^{-k/8}) + 4 \cdot 2^{-k/8} \le 2^{-k/9}.$$

□

Finally, we prove Claim 5.

Proof of Claim 5. Let I be the indicator function for set D:

$$I(x) := \begin{cases} 1 & x \in D \\ 0 & x \notin D. \end{cases}$$

By Lemma 5.2, we have

$$\sum_{b\in\{0,1\}^k}\left|\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{[b]}}[I(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[I(x)]\right|^2 \le \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k+1}}[I(x)] = N_D/2^{k+1} \le N_D/2^k. \tag{18}$$

Note that from the definitions, $\mathbb{E}_{x\sim\mathcal{U}_{[b]}}[I(x)] = N_b/2^k$ and $\mathbb{E}_{x\sim\mathcal{U}_{k+1}}[I(x)] = N_D/2^{k+1}$. Plugging these in (18), we have

$$\sum_{b\in\{0,1\}^k}\left|N_b/2^k - N_D/2^{k+1}\right|^2 \le N_D/2^k.$$

Scaling each side by $\left(2^k/N_D\right)^2$, we have

$$\sum_{b\in\{0,1\}^k}\left|N_b/N_D-\frac{1}{2}\right|^2 \le N_D/2^k\cdot\left(2^k/N_D\right)^2 = 2^k/N_D.$$

Equivalently,

$$\mathop{\mathbb{E}}_{b\sim\mathcal{U}_k}\left[\left|N_b/N_D-\frac{1}{2}\right|^2\right] \le 1/N_D \le 2^{-k/2},$$

where the last inequality follows from the assumption that $N_D = |D| \ge 2^{k/2}$. Finally, by Markov's inequality, when $b\sim\mathcal{U}_k$, with probability $1-2^{-k/8}$, we have $\left|N_b/N_D-\frac{1}{2}\right|^2 \le 2^{-k/2}\cdot 2^{k/8} \le 2^{-k/4}$, and it follows $\left|N_b/N_D-\frac{1}{2}\right| < 2^{-k/8}$, which completes the proof. □

7 The Complete Pseudo-random Generator

In this section we construct the PRG.

Reminder of Theorem 1.3 *For all $m = O(n)$ and $k = \Omega(\log n)$, there exists an $(O(k), m, n, \Omega(k))$ BCAST(1) PRG that can be constructed within $O(k)$ rounds. In particular, the PRG works as follows*

- *Each processor gets $k + k\cdot\frac{(m-k)}{n}$ private random bits.*
- *Then in $O\left(\frac{m-k}{n}\cdot k\right) = O(k)$ rounds, all processors broadcast their last $k\cdot\frac{(m-k)}{n}$ random bits. And they use that to construct a random matrix $M\in\{0,1\}^{k\times(m-k)}$.*
- *Each processor's output is simply the concatenation of its first k random bits x and x^TM.*

The following corollary follows directly from the above theorem.

Corollary 7.1. *Let A be a k-round randomized BCAST(1) algorithm with poly(n) time processors, where each processor uses up to n random bits and $k = \Omega(\log n)$. Then there exists an algorithm A' solving the same problem within $O(k)$-rounds, where each processor uses at most k random bits.*

Note that the correctness of Theorem 1.3 follows directly from Theorem 5.4 (restated below). We spend the remainder of this section proving Theorem 5.4.

Notations. We first recall some notations. Let $M\in\{0,1\}^{n\times m}$. We use $\mathcal{U}_M$ to denote the uniform distribution on the following set $\{(x, x^TM) : x\in\{0,1\}^n\}$, which is a subset of $\{0,1\}^{n+m}$. For integers n and m, we use $\mathcal{U}_{n\times m}$ to denote the uniform distribution on $\{0,1\}^{n\times m}$.

Supposing there are n processors in total, we are going to assume they are deterministic. Processor i can be defined by a function $f_i : \{0,1\}^m\times\{0,1\}^*\to\{0,1\}$, such that $f_i(z,p)$ is the bit player i outputs when it gets the input z and previous history p. We are going to use $f_i^{|p}$ to denote the function $f_i(\cdot,p)$ for simplicity. If transcript p is incompatible with player i having input z, then we set $f_i(z,p)$ arbitrarily.

Reminder of Theorem 5.4 *Let n, m, k be three integers. Consider the following two cases:*

- *(A) All processors receive random inputs from $\mathcal{U}_m$.*
- *(B) Let M be a uniform sample from $\mathcal{U}_{k\times(m-k)}$, then all processors receive inputs from $\mathcal{U}_M$.*

For $j \le k/10$, $m \le 2^{k/20}$ and any j-round BCAST(1) protocol, the statistical distance between the distributions of its transcripts in case (A) and (B) is at most $O\left(\frac{j \cdot n}{2^{k/9}}\right)$.

To prove Theorem 5.4, we need the following technical lemma, whose proof is deferred to the end of this section.

Lemma 7.2. *Assuming $m \le 2^{k/20}$, given a function $f : \{0,1\}^m \to \{0,1\}$ and a set $D \subseteq \{0,1\}^m$ with $|D| \ge 2^{m-k/2}$, let $\mathcal{U}_{M,D}$ and $\mathcal{U}_{m,D}$ be the conditional distributions of $\mathcal{U}_M$ and $\mathcal{U}_m$ on the set D.[16] We have*

$$\mathop{\mathbb{E}}_{M \sim \mathcal{U}_{k\times(m-k)}} \|f(\mathcal{U}_{M,D}) - f(\mathcal{U}_{m,D})\| \le 2^{-k/9}.$$

Proof of Theorem 5.4. Similar to the proof of Theorem 5.1, we will consider a slightly stronger model where we have $j \cdot n$ turns, and on the t^{th} turn, processor $(t-1) \bmod n+1$ gets to send a single bit. We use similar notations as in the proof of Theorem 5.1 and Theorem 5.3. Let $\mathcal{P}^{(t)}_{\mathsf{rand}}$ be the distribution of the transcripts of the first t rounds when all processors get random input from $\mathcal{U}_m$. For a matrix $M \in \{0,1\}^{k\times(m-k)}$, we use $P^{(t)}_M$ to denote the distribution of the transcript of the first t rounds when all processors get random input from $\mathcal{U}_M$. Recall that $\mathcal{U}_M$ is the uniform distribution on the set $\{(x, x^T M) : x \in \{0,1\}^k\}$.

Note that to prove the theorem, it suffices to show that the distribution $\mathcal{P}^{(j\cdot n)}_{\mathsf{rand}}$ is close to $\mathcal{P}^{(j\cdot n)}_M$ for most choices of $M \sim \mathcal{U}_{k\times(m-k)}$. For this purpose, we are going to prove the following inequality holds for any $t \le j \cdot n$:

$$\mathop{\mathbb{E}}_{M \sim \mathcal{U}_{k\times(m-k)}} \left[\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_M\|\right] \le 2 \cdot t \cdot 2^{-k/9}. \tag{19}$$

It is easy to see that plugging in $t = j \cdot n$, (19) implies the theorem. To prove (19) for all t, we induct on t. Clearly, (7) holds when $t = 0$. So it suffices to show that when it holds for $t-1$, it also holds for t. Let $i = (t-1) \pmod n + 1$ be the processor broadcasting at the t-th turn.

For an $M \in \{0,1\}^{k\times(m-k)}$, we wish to bound $\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_M\|$. By Lemma 1.9, we have

$$\|\mathcal{P}^{(t)}_{\mathsf{rand}} - \mathcal{P}^{(t)}_M\| \le \|\mathcal{P}^{(t-1)}_{\mathsf{rand}} - \mathcal{P}^{(t-1)}_M\| + \mathop{\mathbb{E}}_{p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}} \left[\left\| f_i^{|p}(\mathcal{U}_{m,p}) - f_i^{|p}(\mathcal{U}_{M,p})\right\|\right]. \tag{20}$$

In above, $\mathcal{U}_{m,p}$ and $\mathcal{U}_{M,p}$ are distributions $\mathcal{U}_m$ and $\mathcal{U}_M$ conditioned on the previous transcript p. Specifically, let $D^{(t-1)}_p$ denote the set of inputs to f_i which are consistent with the transcript p, then $\mathcal{U}_{m,p}$ and $\mathcal{U}_{M,p}$ are the conditional distributions of $\mathcal{U}_m$ and $\mathcal{U}_M$ on set $D^{(t-1)}_p$.

We wish to use Lemma 7.2 to bound the second term on the right side of (20). To do so, we need to show $D^{(t-1)}_p$ is large with high probability for $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$. The following claim can be proved in exactly the same way as Claim 4 in the proof of Theorem 5.3.

Claim 6. *For $t \le j \cdot n \le \frac{k\cdot n}{10}$, with probability $1 - 2^{-k/4}$ over $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$, we have $|D^{(t-1)}_p| \ge 2^{m-k/4}$.*

Now, for each $M \in \{0,1\}^{k\times(m-k)}$, we again define a score s_M as follows:

$$s_M := \mathop{\mathbb{E}}_{p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}} \left[\left\| f_i^{|p}(\mathcal{U}_{m,p}) - f_i^{|p}(\mathcal{U}_{M,p})\right\|\right].$$

For $p \in \{0,1\}^{t-1}$, we set $s_{M,p} := \left\| f_i^{|p}(\mathcal{U}_{m,p}) - f_i^{|p}(\mathcal{U}_{M,p})\right\|$. Then by Lemma 7.2 and Claim 6, when $p \sim \mathcal{P}^{(t-1)}_{\mathsf{rand}}$, with probability at least $1 - 2^{-k/4}$, we have $|D^{(t-1)}_p| \ge 2^{m-k/4}$, and therefore

[16] When $\mathcal{U}_M$ ($\mathcal{U}_m$) has no mass on D, we set $\mathcal{U}_{M,D}$ ($\mathcal{U}_{m,D}$ to be the uniform distribution on D.)

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}[s_{M,p}] \le 2^{-k/9}.$$

Therefore, it follows

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}[s_M] = \mathop{\mathbb{E}}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}} \mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}[s_{M,p}] \le 2^{-k/4} + 2^{-k/9} \le 2\cdot 2^{-k/9}.$$

Now we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\left[\|\mathcal{P}_{\mathsf{rand}}^{(t)} - \mathcal{P}_{[b]}^{(t)}\|\right] \le & \mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\left[\|\mathcal{P}_{\mathsf{rand}}^{(t-1)} - \mathcal{P}_{[b]}^{(t-1)}\| + \mathop{\mathbb{E}}_{p\sim\mathcal{P}_{\mathsf{rand}}^{(t-1)}}\left[\left\|f_i^{|p}(\mathcal{U}_{m,p}) - f_i^{|p}(\mathcal{U}_{M,p})\right\|\right]\right] \\
\le & 2\cdot(t-1)\cdot 2^{-k/9} + \mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}[s_M] \\
\le & 2\cdot t\cdot 2^{-k/9}.
\end{aligned}$$

The above proves (19) for t, which completes the whole proof. □

7.1 Proof of Lemma 7.2

Before proving Lemma 7.2, we first prove the following technical lemma, which is a generalization of Lemma 5.2.

Lemma 7.3. *Given a function $f:\{0,1\}^m\to\{0,1\}$, we have*

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\left[\|f(\mathcal{U}_m) - f(\mathcal{U}_M)\|^2\right] \le 2^{-k}\cdot(m-k)^2\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[f(x)].$$

Proof. Let $M\sim\mathcal{U}_{k\times(m-k)}$. Let $v_1,v_2,\ldots,v_{m-k}$ be all the $m-k$ columns of M, such that v_1 is the last column and v_{m-k} is the first. Clearly, u_i's are i.i.d. samples from $\mathcal{U}_k$.

We are going to prove this lemma via a hybrid argument, for a fixed M, let $\mathcal{U}_{M,j}$ be the uniform distribution on the following set

$$\{(x, x^{(k)}\cdot v_j, x^{(k)}\cdot v_{j-1},\ldots,x^{(k)}\cdot v_1) : x\in\{0,1\}^{m-j}\},$$

where $x^{(k)}$ denotes the first k bits of string x. That is, for $x\sim\mathcal{U}_{M,j}$, the first $m-j$ bits are completely random, while the last j bits are generated according to M. By definition, it is easy to see that $\mathcal{U}_{M,0}=\mathcal{U}_m$, and $\mathcal{U}_{M,m-k}=\mathcal{U}_M$.

The following claim is the central ingredient of our hybrid argument.

Claim 7. *For $0\le j<m-k$, we have*

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\|f(\mathcal{U}_{M,j}) - f(\mathcal{U}_{M,j+1})\|^2 \le 2^{-k}\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[f(x)].$$

Before proving Claim 7, we show it implies our lemma.

First, for k reals $a_1,a_2,\ldots,a_k$, we have $\|a\|_1\le\sqrt{k}\cdot\|a\|_2$, and consequently

$$\left(\sum_{i=1}^k a_i\right)^2 \le k\cdot\sum_{i=1}^k a_i^2. \tag{21}$$

So we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\|f(\mathcal{U}_m)-f(\mathcal{U}_M)\|^2 &= \mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\|f(\mathcal{U}_{M,0})-f(\mathcal{U}_{M,m-k})\|^2 \\
&= \mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}(m-k)\cdot\sum_{j=0}^{m-k-1}\|f(\mathcal{U}_{M,j})-f(\mathcal{U}_{M,j+1})\|^2 \quad \text{(by (21))} \\
&= (m-k)\cdot\sum_{j=0}^{m-k-1}\cdot\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\|f(\mathcal{U}_{M,j})-f(\mathcal{U}_{M,j+1})\|^2 \\
&\le 2^{-k}\cdot(m-k)^2\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[f(x)]. \quad \text{(by Claim 7)}
\end{aligned}$$

Finally, we prove Claim 7.

Proof of Claim 7. First, note that $v_{j+2},\ldots,v_{m-k}$ are not involved in the inequality in the claim. Suppose we fix $v_1,v_2,\ldots,v_j$ first. We use $v^{(j)}$ to denote this vector sequence.

Now, we define the extension function $E_{v^{(j)}}:\{0,1\}^{m-j}\to\{0,1\}^m$ as follows

$$E_{v^{(j)}}(x) := (x, x^{(k)}\cdot v_j, x^{(k)}\cdot v_{j-1},\ldots,x^{(k)}\cdot v_1).$$

That is, extending the vector x as if $v^{(j)}$ is the last j columns of the matrix M.

We also define $g_{v^{(j)}}:\{0,1\}^{m-j}\to\{0,1\}$ by composing $E_{v^{(j)}}$ and f:

$$g_{v^{(j)}}(x) := f(E_{v^{(j)}}(x)).$$

Let $k'=m-j-1$, and $\mathcal{U}_{[v_{j+1}]}$ be the uniform distribution on the set $\{(x,x^{(k)}\cdot v_{j+1}):x\in\{0,1\}^{k'}\}$. By a similar proof of Lemma 5.2, we have

$$\sum_{v_{j+1}\in\{0,1\}^k}\|g_{v^{(j)}}(\mathcal{U}_{[v_{j+1}]})-g_{v^{(j)}}(\mathcal{U}_{k'+1})\|^2 \le \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k'+1}}[g_{v^{(j)}}(x)],$$

or equivalently,

$$\mathop{\mathbb{E}}_{v_{j+1}\sim\mathcal{U}_k}\|g_{v^{(j)}}(\mathcal{U}_{[v_{j+1}]})-g_{v^{(j)}}(\mathcal{U}_{k'+1})\|^2 \le 2^{-k}\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{k'+1}}[g_{v^{(j)}}(x)]. \tag{22}$$

Averaging over all $v_1,v_2,\ldots,v_j$ from $\mathcal{U}_k$, from the definition of $\mathcal{U}_{M,j}$ and $\mathcal{U}_{M,j+1}$, the left side of (22) becomes

$$\mathop{\mathbb{E}}_{v_1,v_2,\ldots,v_j\sim\mathcal{U}_k}\left[\mathop{\mathbb{E}}_{v_{j+1}\sim\mathcal{U}_k}\|g_{v^{(j)}}(\mathcal{U}_{[v_{j+1}]})-g_{v^{(j)}}(\mathcal{U}_{m-j})\|^2\right] = \mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\|f(\mathcal{U}_{M,j})-f(\mathcal{U}_{M,j+1})\|^2,$$

and the right side becomes

$$\mathop{\mathbb{E}}_{v_1,v_2,\ldots,v_j\sim\mathcal{U}_k}\left[2^{-k}\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{m-j}}[g_{v^{(j)}}(x)]\right] = 2^{-k}\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[f(x)],$$

which completes the proof.

□

Now we move to prove Lemma 7.2.

Proof of Lemma 7.2. The following proof are quite similar to the proof of Lemma 6.1. Let g be the following function

$$g(x) := \begin{cases} f(x) & x \in D \\ 0 & x \notin D. \end{cases}$$

Let $N_D := |D|$, and for $M \in \{0,1\}^{k\times(m-k)}$, let D_M be the support set of $\mathcal{U}_{[M]}$, and $N_M := |D \cap D_M|$. That is, N_D is the support size of distribution $\mathcal{U}_{m,D}$, while N_M is the support size of $\mathcal{U}_{M,D}$.

We need the following claim, whose proof is deferred until we prove the lemma.

Claim 8. *Let $M \sim \mathcal{U}_{k\times(m-k)}$, with probability $1 - 2^{-k/8}$, we have*

$$\left| N_M/N_D - 2^{-(m-k)} \right| \le 2^{-k/8} \cdot 2^{-(m-k)}.$$

By Lemma 5.2, we have

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}} \|g(\mathcal{U}_M) - g(\mathcal{U}_m)\|^2 \le m^2 \cdot 2^{-k} \cdot \mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[g(x)] \le m^2 \cdot 2^{-k} \cdot N_D/2^m.$$

Equivalently,

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}} \left[\left| \mathop{\mathbb{E}}_{x\sim\mathcal{U}_M}[g(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[g(x)] \right|^2 \right] \le m^2 \cdot 2^{-k} \cdot N_D/2^m. \tag{23}$$

To make use of the above bound (23), we now relate $g(\mathcal{U}_M)$ and $g(\mathcal{U}_m)$ to $f(\mathcal{U}_{M,D})$ and $f(\mathcal{U}_{m,D})$. From the definition of g, we have

$$\begin{aligned} \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{M,D}}[f(x)] &= \mathop{\mathbb{E}}_{x\sim\mathcal{U}_M}[g(x)] \cdot \frac{2^k}{N_M} && \text{(the support size of } \mathcal{U}_M \text{ is } 2^k) \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{U}_M}[g(x)] \cdot \frac{2^k}{N_D/2^{m-k}} \cdot \frac{N_D}{2^{m-k}N_M}, && (24) \end{aligned}$$

and

$$\begin{aligned} \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{m,D}}[f(x)] &= \mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[g(x)] \cdot \frac{2^m}{N_D} \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[g(x)] \cdot \frac{2^k}{N_D/2^{m-k}}. \end{aligned}$$

That is, $\mathbb{E}_{x\sim\mathcal{U}_{M,D}}[f(x)]$ and $\mathbb{E}_{x\sim\mathcal{U}_{m,D}}[f(x)]$ are $\mathbb{E}_{x\sim\mathcal{U}_M}[g(x)]$ and $\mathbb{E}_{x\sim\mathcal{U}_m}[g(x)]$ scaled by a factor of $\frac{2^k}{N_D/2^{m-k}}$, except for another $\frac{N_D}{2^{m-k}N_M}$ factor in (24), which is very close to 1 for most M's by Claim 8.

We first ignore the $\frac{2^k}{N_D/2^{m-k}}$ factor in (24), and define

$$F_M := \mathop{\mathbb{E}}_{x\sim\mathcal{U}_M}[g(x)] \cdot \frac{2^k}{N_D/2^{m-k}},$$

and

$$F_m := \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{m,D}}[f(x)] = \mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[g(x)] \cdot \frac{2^k}{N_D/2^{m-k}}.$$

Scaling each side of (23) by $\left(\frac{2^k}{N_D/2^{m-k}}\right)^2$, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\left[|F_M - F_m|^2\right] &\le m^2\cdot 2^{-k}\cdot N_D/2^m\cdot\left(\frac{2^k}{N_D/2^{m-k}}\right)^2\\ &\le 2^{k/10}\cdot 2^{m-k}/N_D && (m\le 2^{k/20})\\ &\le 2^{k/10}\cdot 2^{-k/2}. && (N_D=|D|\ge 2^{m-k/2})\end{aligned}$$

That is, by Markov's inequality, when $M\sim\mathcal{U}_{k\times(m-k)}$, with probability $1-2^{-k/8}$, we have

$$|F_M - F_{k+1}|^2 \le 2^{k/10}\cdot 2^{-k/2}\cdot 2^{k/8}\le 2^{-k/4},$$

which means $|F_M - F_{k+1}| < 2^{-k/8}$.

Now, by Claim 8, when $M\sim\mathcal{U}_{k\times(m-k)}$, with probability $1-2^{-k/8}$, we have

$$\begin{aligned}&|N_M/N_D - 2^{-(m-k)}| < 2^{-k/8}\cdot 2^{-(m-k)}\\ \Rightarrow&\left|\frac{N_M\cdot 2^{m-k}}{N_D}-1\right| < 2^{-k/8}\\ \Rightarrow&\left|\frac{N_D}{N_M\cdot 2^{m-k}}-1\right| < 2\cdot 2^{-k/8}.\end{aligned}$$

Putting everything together, with probability $1-2\cdot 2^{-k/8}$, we have

$$\begin{aligned}\|f(\mathcal{U}_{M,D}) - f(\mathcal{U}_{m,D})\| &= \left|\mathop{\mathbb{E}}_{x\sim\mathcal{U}_{M,D}}[f(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_{m,D}}[f(x)]\right|\\ &= \left|F_M\cdot\frac{N_D}{2^{m-k}N_M} - F_m\right|\\ &\le |F_M - F_m| + F_M\cdot\left|\frac{N_D}{2^{m-k}N_M}-1\right|\\ &\le 3\cdot 2^{-k/8}.\end{aligned}$$

That last line follows from that $F_M = \mathbb{E}_{x\sim\mathcal{U}_M}[g(x)]\cdot\frac{2^k}{N_D/2^{m-k}}\le\frac{N_D}{2^m}\cdot\frac{2^m}{N_D}=1$.

Therefore, we have

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\|f(\mathcal{U}_{M,D}) - f(\mathcal{U}_{m,D})\|\le 2\cdot 2^{-k/8}+3\cdot 2^{-k/8}\le 2^{-k/9}.$$

□

Finally, we prove Claim 8.

Proof of Claim 8. Let I be the indicator function for set D:

$$I(x) := \begin{cases}1 & x\in D\\ 0 & x\notin D.\end{cases}$$

By Lemma 7.3, we have

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}}\left[\left|\mathop{\mathbb{E}}_{x\sim\mathcal{U}_M}[I(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[I(x)]\right|^2\right]\le 2^{-k}\cdot m^2\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{U}_m}[I(x)] \tag{25}$$

Note that $\mathbb{E}_{x\sim\mathcal{U}_M}[I(x)] = N_M/2^k$ and $\mathbb{E}_{x\sim\mathcal{U}_m}[I(x)] = N_D/2^m$. Plugging these in (25), we have

$$\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}} \left|N_M/2^k - N_D/2^m\right|^2 \leq 2^{-k}\cdot m^2 \cdot N_D/2^m.$$

Scaling both sides by $\left(\frac{2^k}{N_D}\right)^2$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{M\sim\mathcal{U}_{k\times(m-k)}} \left[\left|N_M/N_D - 2^{-(m-k)}\right|^2\right] &\leq 2^{-k}\cdot m^2 \cdot N_D/2^m \cdot \left(\frac{2^k}{N_D}\right)^2 \\
&\leq 2^{-(m-k)}\cdot m^2/N_D \\
&\leq 2^{-(m-k)}\cdot m^2 \cdot 2^{-(m-k+k/2)} && (N_D = |D| \geq 2^{m-k/2}) \\
&\leq 2^{-2(m-k)-k/2}\cdot 2^{k/10}. && (m \leq 2^{k/20})
\end{aligned}$$

By Markov's inequality, when $M \sim \mathcal{U}_{k\times(m-k)}$, with probability $1 - 2^{-k/8}$, we have

$$\left|N_M/N_D - 2^{-(m-k)}\right|^2 \leq 2^{-2(m-k)-k/2}\cdot 2^{k/10}\cdot 2^{k/8} \leq 2^{-2(m-k)-k/4},$$

which is equivalent to

$$\left|N_M/N_D - 2^{-(m-k)}\right| \leq 2^{-k/8}\cdot 2^{-(m-k)}.$$

□

8 A Matching Lower bound

In this section, we prove that our pseudo-random generator is optimal up to constant factors. That is, we prove that the seed length is optimal. We show that any pseudo-random generator with a seed length of size s can be broken within $O(s)$ rounds (note that our pseudo-random generator is secure up to $\Omega(s)$ rounds when the output of the PRG is of length n for each processor).

Theorem 8.1 (Seed Length Lower Bound)**.** *Let n, m, and k be three integers. Suppose that there are n processors in the Broadcast Congested Clique model. Furthermore, suppose there is a pseudo-random generator which when each processor starts with a seed of size k, gives each node a pseudo-random string of size m. Then there is an $O(k)$ round Broadcast Congested Clique protocol that can break this PRG.*

Proof. Consider the following $k+1$-round protocol: each processor broadcasts its first $k+1$ pseudo-random bits. In the case that these bits are pseudo-random, we know that since nk random bits were used as a seed to construct these strings, the transcript of the first round must be one of 2^{nk} options. However, in the truly random case, there are $2^{n(k+1)}$ options. So, consider the algorithm that outputs 1 if the transcript is one of the 2^{nk} options consistent with the pseudo-random generator, and otherwise outputs a 0. Then if the pseudo-random generator was used, then the probability of outputting a 1 is 1. In the truly random case, the probability of outputting a 1 is $\frac{2^{nk}}{2^{n(k+1)}} = \frac{1}{2^n}$. Hence, this algorithm distinguished between the truly random and the pseudo-random case with all but an exponentially small probability. □

9 Discussion

Our paper leaves some problems open. A main open problem is whether it is possible to improve the planted clique lower bound to show that if the clique is of size $k = \Theta(n^{1/2-\varepsilon})$, the planted clique problem still requires a number of rounds polynomial in n.

It would be interesting to extend the framework to work for undirected graphs as well. This causes the rows of the input matrix not to be independent (instead, each pair of rows contain one shared bit). Our current proofs rely on the rows of the input being independent, but we believe it may be possible to extend the framework to also work when the rows exhibit a small amount of dependence.

There are many problems in the BCAST model that may be interesting to try to prove lower bounds for using the techniques in this paper. These include counting triangles (or K_4s) in random graphs, constructing an MST on a complete graph with random weights to the edges, finding communities in a graph sampled from the stochastic block model, the "planted Hamiltonian cycle" problem (or, determining whether there is a Hamiltonian cycle in a random graph where the probability of an edge being included is chosen properly so that the probability of such a cycle existing is some constant), graph connectivity, finding the diameter of a random graph (the average degree must be chosen to be low enough so that the diameter is not 2 with high probability), and APSP on a complete graph with random weight assignments. There are many possibilities.

The uniform distribution is not necessarily the most natural distribution to consider in the broadcast congested clique model. Our techniques are more general and could hopefully be used to prove lower bounds for other distributions as well. It would be interesting to consider other input distributions which may be "natural" for the problem at hand.

Acknowledgments

We would like to thank Shafi Goldwasser, Yang Liu, and Merav Parter for helpful discussions. Many thanks to Sidhanth Mohanty for very fruitful discussions about the planted clique problem.

Thanks to anonymous reviewers for many useful comments.

References

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[BARR15] Florent Becker, Antonio Fernández Anta, Ivan Rapaport, and Eric Rémila. Brief announcement: A hierarchy of congested clique models, from broadcast to unicast. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 167–169, 2015.

[BGR96] Mihir Bellare, Juan A. Garay, and Tal Rabin. Distributed pseudo-random bit generators - A new way to speed-up shared coin tossing. In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, Philadelphia, Pennsylvania, USA, May 23-26, 1996*, pages 191–200, 1996.

[BHK+16] Boaz Barak, Samuel B. Hopkins, Jonathan A. Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 428–437, 2016.

[BKRW17] Mark Braverman, Young Kun-Ko, Aviad Rubinstein, and Omri Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1326–1341, 2017.

[BMRT18] Florent Becker, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. The impact of locality on the detection of cycles in the broadcast congested clique model. In *Latin American Symposium on Theoretical Informatics*, pages 134–145. Springer, 2018.

[BT06] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006.

[CK18] Artur Czumaj and Christian Konrad. Detecting cliques in CONGEST networks. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, pages 16:1–16:15, 2018.

[CKK+15] Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. Algebraic methods in the congested clique. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 143–152, 2015.

[CP17] Yi-Jun Chang and Seth Pettie. A time hierarchy theorem for the LOCAL model. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 156–167, 2017.

[CPS17] Keren Censor-Hillel, Merav Parter, and Gregory Schwartzman. Derandomizing local distributed algorithms under bandwidth restrictions. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, pages 11:1–11:16, 2017.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[dERRU16] Pablo Moisset de Espanés, Ivan Rapaport, Daniel Remenik, and Javiera Urrutia. Robust reconstruction of barabási-albert networks in the broadcast congested clique model. *Networks*, 67(1):82–91, 2016.

[DGP14] Yael Dekel, Ori Gurel-Gurevich, and Yuval Peres. Finding hidden cliques in linear time with high probability. *Combinatorics, Probability & Computing*, 23(1):29–49, 2014.

[DKO14] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 367–376, 2014.

[DM15] Yash Deshpande and Andrea Montanari. Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 523–562, 2015.

[DNO14] Shahar Dobzinski, Noam Nisan, and Sigal Oren. Economic efficiency requires interaction. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 233–242, 2014.

[DW08] Ronald De Wolf. A brief introduction to fourier analysis on the boolean cube. *Theory of Computing, Graduate Surveys*, 1(1-20):5, 2008.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994–1005, 1993.

[FK00] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semi-random graph. *Random Struct. Algorithms*, 16(2):195–208, 2000.

[FK03] Uriel Feige and Robert Krauthgamer. The probable value of the lovász–schrijver relaxations for maximum independent set. *SIAM J. Comput.*, 32(2):345–370, 2003.

[FKP13] Pierre Fraigniaud, Amos Korman, and David Peleg. Towards a complexity theory for local distributed computing. *J. ACM*, 60(5):35:1–35:26, 2013.

[Gal16] François Le Gall. Further algebraic algorithms in the congested clique model and applications to graph-theoretic problems. In *Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings*, pages 57–70, 2016.

[GHK18] Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. On derandomizing local distributed algorithms. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 662–673, 2018.

[GHM18] Ofer Grossman, Bernhard Haeupler, and Sidhanth Mohanty. Algorithms for noisy broadcast with erasures. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 107. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[GKM17] Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. On the complexity of local distributed graph problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 784–797, 2017.

[HKP+18] Samuel B Hopkins, Pravesh Kothari, Aaron Henry Potechin, Prasad Raghavendra, and Tselil Schramm. On the integrality gap of degree-4 sum of squares for planted clique. *ACM Transactions on Algorithms (TALG)*, 14(3):28, 2018.

[HP15] Stephan Holzer and Nathan Pinsker. Approximation of distances and shortest paths in the broadcast congest clique. In *19th International Conference on Principles of Distributed Systems, OPODIS 2015, December 14-17, 2015, Rennes, France*, pages 6:1–6:16, 2015.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 356–364. ACM, 1994.

[Jer92] Mark Jerrum. Large cliques elude the metropolis process. *Random Struct. Algorithms*, 3(4):347–360, 1992.

[JN17a] Tomasz Jurdzínski and Krzysztof Nowicki. Brief announcement: On connectivity in the broadcast congested clique. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 91. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[JN17b] Tomasz Jurdzinski and Krzysztof Nowicki. MSF and connectivity in limited variants of the congested clique. *arXiv preprint arXiv:1703.02743*, 2017.

[Kol99] Valentin Fedorovich Kolchin. *Random graphs*, volume 53. Cambridge University Press, 1999.

[KS18] Janne H. Korhonen and Jukka Suomela. Towards a complexity theory for the congested clique. In *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, SPAA 2018, Vienna, Austria, July 16-18, 2018*, pages 163–172, 2018.

[Kuc95] Ludek Kucera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995.

[MPW15] Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 87–96, 2015.

[MT16] Pedro Montealegre and Ioan Todinca. Brief announcement: Deterministic graph connectivity in the broadcast congested clique. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 245–247, 2016.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Information processing letters*, 39(2):67–71, 1991.

[NPR99] Moni Naor, Benny Pinkas, and Omer Reingold. Distributed pseudo-random functions and kdcs. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 327–346, 1999.

[NY19] Jelani Nelson and Huacheng Yu. Optimal lower bounds for distributed and streaming spanning forest computation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1844–1860, 2019.

[O'D14] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[PY18] Merav Parter and Eylon Yogev. Congested clique algorithms for graph spanners. In *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, pages 40:1–40:18, 2018.

[Vad12] Salil P Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Yao77] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 222–227, 1977.

A An Analogue of Newman's Theorem in BCAST(1)

In this appendix we adapt Newman's technique from [New91] to show that in the computationally unbounded setting, every randomized k rounds $\mathsf{BCAST}(1)$ protocols in which there are n processors, each with m input bits and outputs k bits at the end, can be simulated with only $O(k \cdot n + \log m)$ public random bits (this has not been observed before this work in the context of the broadcast congested clique). We note that Newman's approach can be adapted to the Unicast Congested Clique model (where each vertex may send different messages to different nodes, instead of broadcasting the same message to all other nodes),

but it is not clear how to adapt the the approach to the standard CONGEST model, in which nodes may communicate only along edges of the input graph.

Let $\vec{x} = (x_1, x_2, \ldots, x_n) \in (\{0,1\}^m)^n$ be an input, we use $\mathcal{P}(P, \vec{x})$ to denote the joint distribution of the transcript and the concatenation of all processors' output bits of the k-round $\mathsf{BCAST}(1)$ protocol P running on input $\vec{x}$, that is, $\mathcal{P}(P, \vec{x})$ is a distribution on $\{0,1\}^{2kn}$.

We say a protocol P_{new} ε-simulates another protocol P, if for all possible input $\vec{x}$, we have $\|\mathcal{P}(P, \vec{x}) - \mathcal{P}(P_{\mathsf{new}}, \vec{x})\| < \varepsilon$.

Theorem A.1. *Let P be a randomized $\mathsf{BCAST}(1)$ protocol with n processors, each with m input bits and outputs k bits at the end. For all $\varepsilon > 0$, there is an equivalent randomized $\mathsf{BCAST}(1)$ protocol P_{new} ε-simulating P with only $O(k \cdot n + \log(m) + \log \varepsilon^{-1})$ public random bits.*

Proof. In the following we are just going to mimic the proof of Newman's theorem.

Without loss of generality we can assume P is a public coin protocol. Suppose it makes use of at most N public coins, where N can be arbitrary large.

Fix an input $x \in (\{0,1\}^m)^n$. Note that $\|\mathcal{P}(P, \vec{x}) - \mathcal{P}(P_{\mathsf{new}}, \vec{x})\| < \varepsilon$ is equivalent to that for all function $f : \{0,1\}^{2kn} \to \{0,1\}$,

$$\left| \mathop{\mathbb{E}}_{p \sim \mathcal{P}(P, \vec{x})} [f(p)] - \mathop{\mathbb{E}}_{p \sim \mathcal{P}(P_{\mathsf{new}}, \vec{x})} [f(p)] \right| < \varepsilon.$$

We then fix a function f. Suppose we draw a public random string $w \sim \mathcal{U}_N$, and we use P_w to denote protocol P with public random string setting to w and $P_w(\vec{x})$ to denote the concatenation of its transcript and all processor' output bits on input $\vec{x}$.

Now, suppose we pick T $w_1, w_2, \ldots, w_T$ uniform random samples from $\mathcal{U}_N$, by a simple Chernoff bound, we have

$$\Pr\left[\left| \frac{1}{T} \cdot \sum_{i=1}^{T} f(P_{w_i}(\vec{x})) - \mathop{\mathbb{E}}_{p \sim \mathcal{P}(P_{\mathsf{new}}, \vec{x})} [f(p)] \right| > \varepsilon \right] < \exp(-\Omega(\varepsilon^2 \cdot T)).$$

Setting

$$T = \Theta(\varepsilon^{-2} \cdot \left(nm + 2^{2kn} \right)),$$

it follows

$$\Pr\left[\left| \frac{1}{T} \cdot \sum_{i=1}^{T} f(P_{w_i}(\vec{x})) - \mathop{\mathbb{E}}_{p \sim \mathcal{P}(P_{\mathsf{new}}, \vec{x})} [f(p)] \right| > \varepsilon \right] < \frac{1}{10 \cdot 2^{nm} \cdot 2^{2^{2kn}}}.$$

Since there are at most $2^{2^{2kn}}$ functions and 2^{nm} input bits, by a simple union bound, we have with probability at least 0.9 over our T samples, $\left| \frac{1}{T} \cdot \sum_{i=1}^{T} f(P_{w_i}(\vec{x})) - \mathbb{E}_{p \sim \mathcal{P}(P_{\mathsf{new}}, \vec{x})}[f(p)] \right| < \varepsilon$ for all input $\vec{x}$ and function $f : \{0,1\}^{2kn} \to \{0,1\}$.

So we can just pick T samples $w_1, w_2, \ldots, w_T$ satisfying the above condition, and define P_{new} as the protocol that makes use of $\log T = O(kn + \log m + \log \varepsilon^{-1})$ coins to select a random index $i \in [T]$, and act according to P_{w_i}. It ε-simulates P by the above discussions. □

Remark A.2. *We remark that in the worst case, at least $\Omega(k \cdot n)$ bits are required to ε-simulate a k-round $\mathsf{BCAST}(1)$ protocol P where each processor outputs k bits. Since if all processors output k uniform random bits, the total entropy of $\mathcal{P}(P, \vec{x})$ on any input $\vec{x}$ is at least $k \cdot n$.*

B Algorithm for Planted Clique in $\mathsf{BCAST}(1)$

In this section we give an algorithm for finding planted clique in $\mathsf{BCAST}(1)$.

Theorem B.1. *Let n be an integer and $\omega(\log^2 n) \leq k \leq n$. Given an input from $\mathcal{A}_k$, there is an $O(n/k \cdot \mathrm{polylog}(n))$ round $\mathsf{BCAST}(1)$ protocol such that at the end of the protocol, with probability at least $1 - 1/n^2$, all processors know the hidden clique C.*

Proof. Let $p = \frac{1}{k} \cdot \log^2 n$.

Algorithm. The algorithm is very simple.

- At the first round of the protocol, each processor decides to stay active with probability p, and broadcasts whether it is active to everyone else.
- Let N_{active} be the number of active processors, if $N_{\mathsf{active}} > 2 \cdot n \cdot p$, all processors just terminate.
- Each active processors broadcast whether it has an edge to each other active processor, which takes $O(n \cdot p) = O(n/k \cdot \mathrm{polylog}(n))$ rounds (i.e., all information about the subrgraph induced by the active processors is broadcasted).
- Now everyone knows the induced subgraph G_{active} consisting of all active processors. Let the largest clique in G_{active} be C_{active}. If $|C_{\mathsf{active}}| < \frac{1}{2} \cdot \log^2 n$, all processors terminate.
- Every processor (including the non-active ones) checks whether it is connected to at least a $9/10$ fractions of vertices in C_{active}, and if it is, it broadcasts with a message saying it is in the clique C. (if it is already in C_{active}, then it also says that.)

Analysis. Intuitively, the algorithm works because a random graph doesn't contain a clique of size $10 \log n$ with high probability. And in the hidden clique case, if we pick each vertex with probability p, then in expectation we would pick $p \cdot k = \log^2 n$ vertices in C, and therefore $|C_{\mathsf{active}}| \geq \frac{1}{2} \cdot \log^2 n$ with high probability, while in a random graph the largest clique is of size $\Theta(\log n)$ with high probability.

Let X_i be the random variable indicating whether processor i is active. And let Y_i be the random variable indicating whether processor i is both in the clique and active.

Note that X_i's are i.i.d., by the multiplicative Chernoff bound, we have

$$\Pr\left[N_{\mathsf{active}} = \sum_{i=1}^{n} X_i > (1+\delta) \cdot p \cdot n\right] \leq e^{-\frac{\delta \cdot p \cdot n}{3}}.$$

Setting $\delta = 1$, we have with high probability, $N_{\mathsf{active}} \leq 2 \cdot p \cdot n$.

Note that although Y_i's are not independent, they are negatively associated, and we have the following by another multiplicative Chernoff bound,

$$\Pr\left[\sum_{i=1}^{n} Y_i < (1-\delta) \cdot p \cdot k\right] \leq e^{-\frac{\delta^2 \cdot p \cdot k}{2}}.$$

Set $\delta = 0.5$. With high probability, there are more than $\frac{1}{2} \cdot p \cdot k = \frac{1}{2} \cdot \log^2 n$ active vertices in C.

Finally, since with high probability, a random graph doesn't contain a clique of size larger than $10 \log n$. We can conclude that with high probability, at least $\frac{1}{2} \log^2 n - 10 \log n$ vertices in C_{active} are actually in C. And it is easy to see that the last step of the algorithm identifies the clique C correctly with high probability. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il