# Topology Dependent Bounds For FAQs

Michael Langberg
University at Buffalo

Shi Li
University at Buffalo

Sai Vikneshwar Mani Jayaraman
University at Buffalo

Atri Rudra
University at Buffalo

## ABSTRACT

In this paper, we prove topology dependent bounds on the number of rounds needed to compute Functional Aggregate Queries (FAQs) studied by Abo Khamis et al. [PODS 2016] in a synchronous distributed network under the model considered by Chattopadhyay et al. [FOCS 2014, SODA 2017]. Unlike the recent work on computing database queries in the Massively Parallel Computation model, in the model of Chattopadhyay et al., nodes can communicate only via private point-to-point channels and we are interested in bounds that work over an *arbitrary* communication topology. This model, which is closer to the well-studied CONGEST model in distributed computing and generalizes Yao's two party communication complexity model, has so far only been studied for problems that are common in the two-party communication complexity literature.

This is the first work to consider more practically motivated problems in this distributed model. For the sake of exposition, we focus on two special problems in this paper: Boolean Conjunctive Query (BCQ) and computing variable/factor marginals in Probabilistic Graphical Models (PGMs). We obtain tight bounds on the number of rounds needed to compute such queries as long as the underlying hypergraph of the query is $O(1)$-degenerate and has $O(1)$-arity. In particular, the $O(1)$-degeneracy condition covers most well-studied queries that are efficiently computable in the centralized computation model like queries with constant treewidth. These tight bounds depend on a new notion of 'width' (namely internal-node-width) for Generalized Hypertree Decompositions (GHDs) of acyclic hypergraphs, which minimizes the number of internal nodes in a sub-class of GHDs. To the best of our knowledge, this width has not been studied explicitly in the theoretical database literature. Finally, we consider the problem of computing the product of a vector with a chain of matrices and prove tight bounds on its round complexity (over the finite field of two elements) using a novel min-entropy based argument.

**ACM Reference Format:**
Michael Langberg, Shi Li, Sai Vikneshwar Mani Jayaraman, and Atri Rudra. 2019. Topology Dependent Bounds For FAQs. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 38 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

In this paper, we prove topology dependent bounds on the number of rounds needed to compute Functional Aggregate Queries (FAQs) of [39] in a synchronous distributed network under the model considered by Chattopadhyay et al. [18, 19]. For ease of exposition, we consider the FAQ-SS problem [7, 39, 51] i.e., FAQ with a single semiring (also called *Marginalize a Product Function* in [3]), which is a special case of the general FAQ problem (defined in Section 5). In FAQ-SS, we are given a multi-hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ where for each hyperedge $e \in \overline{\mathcal{E}}$ we are given an input function $f_e : \prod_{v \in e} \mathrm{Dom}(v) \to \mathbb{D}$. In addition we are given a set of *free variables*[1] $\mathcal{F} \subseteq \overline{\mathcal{V}}$ and our goal is to compute the function:

$$\phi_{\mathcal{F}}(\mathbf{x}) = \sum_{\mathbf{y} \in \prod_{v \in \overline{\mathcal{V}}} \mathrm{Dom}(v) : \mathbf{y}_{\mathcal{F}} = \mathbf{x}} \prod_{e \in \overline{\mathcal{E}}} f_e(\mathbf{y}_e) \qquad (1.0)$$

for every $\mathbf{x} \in \prod_{v \in \mathcal{F}} \mathrm{Dom}(v)$, where $\mathbf{y}_e$ and $\mathbf{y}_{\mathcal{F}}$ are $\mathbf{y}$ projected down to co-ordinates in $e \subseteq \overline{\mathcal{V}}$ for every $e \in \overline{\mathcal{E}}$ and $\mathcal{F} \subseteq \overline{\mathcal{V}}$ respectively. Further, all the operations are over the *commutative semiring*[2] $(\mathbb{D}, +, \cdot)$ with additive identity $\mathbf{0}$. As with database systems, we assume that the functions are given in *listing* representation i.e., the function $f_e$ is represented as a list of its non-zero values: $R_e = \{(\mathbf{y}, f_e(\mathbf{y})) | \mathbf{y} \in \prod_{v \in e} Dom(v) : f_e(\mathbf{y}) \neq \mathbf{0}\}$[3]. We define $D = \max_{v \in \overline{\mathcal{V}}} |\mathrm{Dom}(v)|$, $N = \max_{e \in \overline{\mathcal{E}}} |R_e|$, $k = |\overline{\mathcal{E}}|$ and $r$ as the maximum arity among all functions.

Though our results are semiring agnostic, we mention two special problems that we consider in this paper. The first problem is when $\mathcal{F} = \emptyset$ and the semiring is the *Boolean semiring* $(\mathbb{D} = \{0, 1\}, \vee, \wedge)$. This corresponds to the *Boolean Conjunctive Query* (which we will call BCQ).[4] The other problem is when $\mathcal{F} = e$ for some $e \in \overline{\mathcal{E}}$ and the semiring is $(\mathbb{R}_{\geq 0}, +, \cdot)$, which corresponds to computing a *factor marginal* in *Probabilistic Graphical Models* (or PGMs) – here we think of $f_e$ as a probability distribution. The FAQ setup (and even FAQ-SS) encompasses a large class of problems in varied domains. We refer the reader to the surveys [3, 40] for an overview of these applications.

[1] We would like to mention here that our results hold only for specific choices of free variables.
[2] A triple $(\mathbb{D}, \oplus, \otimes)$ is a *commutative semiring* if $\oplus$ and $\otimes$ are commutative binary operators over $\mathbb{D}$ satisfying the following: (1) $(\mathbb{D}, \oplus)$ is a commutative monoid with an additive identity, denoted by $\mathbf{0}$. (2) $(\mathbb{D}, \otimes)$ is a commutative monoid with a multiplicative identity, denoted by $\mathbf{1}$. (In the usual semiring definition, we do not need the multiplicative monoid to be commutative.) (3) $\otimes$ distributes over $\oplus$. (4) For any element $d \in \mathbb{D}$, we have $d \otimes \mathbf{0} = \mathbf{0} \otimes d = \mathbf{0}$.
[3] We use function/relation interchangeably for $f_e / R_e$ but both mean the same.
[4] $\mathcal{F} = \overline{\mathcal{V}}$ over the Boolean semiring is the natural join problem.

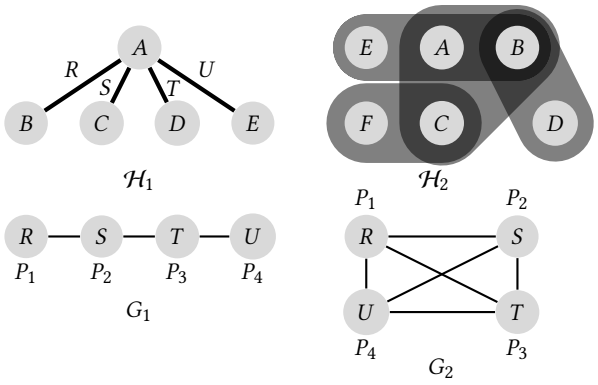


**Figure 1: Two example queries $\mathcal{H}_1$ and $\mathcal{H}_2$ and two topologies – "line" $G_1$ and "clique" $G_2$. $\mathcal{H}_2$ has hyperedges $R(A,B,C)$, $S(B,D)$, $T(C,F)$ and $U(A,B,E)$.**

Given a query $q = \left(\mathcal{H}, \{f_e\}_{e\in\overline{\mathcal{E}}}, \mathcal{F}\right)$, we will consider the number of rounds needed to compute $q$ in a distributed environment. In particular, the underlying *communication topology*[5] $G = (V, E)$ is assumed to be a synchronous network and we would like to compute $q$ on $G$ with the following constraints [18, 19]. Initially, all functions $\{f_e\}_{e\in\overline{\mathcal{E}}}$ are assigned to specific nodes $K \subseteq V : 1 \leq |K| \leq k$ (called *players*). In each *round* of communication, $O(r \cdot \log_2(D))$ bits[6] can be simultaneously communicated on each edge in $E$ (each such edge or *channel* is private to the nodes at its endpoints). At the end of the protocol, a pre-determined player in $K$ knows the answer to $q$. Naturally, we would like to design protocols that minimize the total number of rounds of communication (rounds hereon) needed to compute $q$ on $G$. More generally, we would like to obtain tight bounds depending on $\mathcal{H}$ and $G$ for this problem for *every* query topology $\mathcal{H}$ and *every* network topology $G$. Note that we do not take into account the internal computation done by nodes in $G$ and we assume that all nodes in $V$ co-operatively compute the answer to $q$.

## 1.1 Why this distributed model?

We believe that the strength of our model is its generality. Specifically, it captures query computation in three different paradigms, namely: (1) Computing the natural join query in the Massively Parallel Computation (MPC) model [2, 9, 10, 37, 45, 46], (2) Computing join and aggregation queries for sensor networks [13, 25, 50] and (3) Computation of FAQs on arbitrary topologies using *software defined networks* and optical reconfigurable networks like ProjecToR [27]. Before we discuss these in detail, we would like to mention that the CONGEST model in distributed computing has the same setup as ours [54] with one crucial difference. Unlike our case, where we can compute FAQs on *any* topology in the CONGEST model, the topologies for computing a fixed FAQ typically depend on the query itself.

The sequence of works in the MPC setting focus on computing the natural join $q$ (which is a special case of FAQ-SS as mentioned earlier) on a topology $G$ (with $p$ nodes/servers), which is typically well-connected. Each round of communication has two phases – (1) internal computation among the nodes and (2) communication between the nodes bounded by a node capacity $L$. The goal in MPC is to minimize the number of rounds $h$ needed for computing $q$. There are two different lines of work in this regime – one where $p$ is fixed and the goal is to determine $h, L$ [9, 10, 45, 46] and the other is when $h, L$ are fixed and the goal is to determine $p$ [2]. We compare both these classes of models with ours in the full paper [47] and present an executive summary here.

Roughly speaking, the MPC model defined in [9] is a special case of our model. Moreover, for the case when $\mathcal{H}$ is a star, our generic protocols obtain the same guarantees as the one in [9, 10] up to a constant factor. We consider two different MPC models – one with no replication (which we call MPC(0) [9]) and one with replication (which we dub MPC($\epsilon$) [2, 45]). Both these models have some differences from ours and among themselves. For instance, both these models assume a specific network topology $G'$ (as opposed to any topology $G$ in our case), work on node capacities $L$ (as opposed to edge capacities in our setting) and prove bounds for the natural join problem (in contrast, our bounds apply for the more general FAQ). The input functions are systematically assigned to players in MPC(0) and are uniformly distributed among players in MPC($\epsilon$). The instantation of these models for the setting where $p$ is fixed and $h, L$ is to be determined is the closest to our model. In particular, when $\mathcal{H}$ is a star, our protocols obtain the same guarantees as MPC(0) and are slightly worse in MPC($\epsilon$). Our model does not (yet) handle the scenario when $L$ is fixed and the goal is to determine $p$.

Sensor networks are typically tree-like topologies, where the goal is to efficiently and accurately report aggregate queries on data generated by the sensors. Since the sensors can store only little data, these queries are typically restrictive. We show in the full paper [47] that our results imply bounds for some of these queries. Recently, Internet of Things (IoT) devices [1] show the promise of expanding the data storage/class of queries that can be computed on sensor networks. We believe that our model/results will find more relevance in the IoT setting since the sensors used posess more computation power than those considered in [50]. Finally, our work initiates the study of computation on general topologies to be used in emerging technologies like ProjecToR [27], which has been proposed for use in data centers where topologies can be changed based on the workload.

## 1.2 Summary of Our Contributions

Table 1 lists our results and Section 2 contains a detailed overview of techniques used to obtain the results. We summarize our contributions here. For the sake of brevity, we focus on the BCQ problem. Our main result is the following. For

[5]Note that this is distinct from $\mathcal{H}$ and is just a simple graph: see Figure 1 for an example illustrating this difference.

[6]This is a natural choice since any tuple in any function can be communicated with at most $O(r \cdot \log_2(D))$ bits. Our bounds seamlessly generalize to the cases when each edge – (1) can transmit $B \neq r \cdot \log_2(D)$ bits and (2) has a different capacity, but for ease of exposition, we will not consider that generalization in this paper.

(hyper)graphs $\mathcal{H}$ with constant degeneracy[7] ($d$) and constant arity ($r$), we prove tight bounds (up to constant factors) for computing *any* BCQ on *any* network topology $G$. Constant treewidth implies constant $d$ and, as a result, queries having constant $d$ encompass most well-studied queries that are efficiently computable in the centralized computation model.

*Upper Bound.* Our upper bound needs protocols for solving the following two basic algorithmic tasks: (1) set intersection and (2) sending all inputs to a single node. For (1), our protocol is new in the FAQ literature and for (2), we use a standard protocol from flow networks. Interestingly, our results highlight a notion of width of *acyclic* queries– the number of internal nodes for a subclass of GHDs[8] (defined in Section 2.2.2), which to the best of our knowledge, has not been explicitly studied in the database literature.

*Lower Bound.* Our lower bounds follow from known lower bounds on the (well-studied) TRIBES function in two-party communication complexity literature (defined in Section 2.2.2). At a high level, we start with an arbitrary TRIBES instance and show that it can be reduced to a suitable BCQ problem in our model. We then prove lower bounds for the corresponding BCQ problem using known lower bounds on TRIBES.

We note here that the simplicity of our techniques allows us to extend our results to the general FAQ problem. Further, we would like to mention that extending our bounds to $d$-degenerate graphs with non-constant $d$ has a known bottleneck of solving BCQ of $\mathcal{H}$ on $G$ when $\mathcal{H}$ is a clique and $G$ is an edge. In particular, the gaps dependent on $d$ in Table 1 cannot be resolved without addressing this bottleneck.

Finally, we consider the following FAQ-SS problem of Chain Matrix-Vector Multiplication (MCM): computing $\mathbf{A}_k \cdot \mathbf{A}_{k-1} \cdot \ldots \cdot \mathbf{A}_1 \cdot \mathbf{x}$, where each player gets $\mathbf{x}, \mathbf{A}_1, \ldots, \mathbf{A}_k$ in order and they would like to compute the product over the finite field $\mathbb{F}_2$.[9] Note that this problem is different from the well-known Online Matrix Vector Multiplication problem[10] and is related to $k$ layer neural networks[11]. We prove a tight bound for this problem. The upper bound is simple but the lower bound argument (though conceptually simple), is technically the most involved part of the paper. We use an entropy-based argument using min-entropy instead of the standard Shannon's entropy. This requires more care since we can no longer use the chain rule.

# 2 OUR MODEL AND DETAILED OVERVIEW OF OUR RESULTS

In this section, our goal is to provide a walkthrough of our results and techniques used to prove them. We start with a formal definition of our model. Then, we illustrate, with examples, our results for the case when $\mathcal{H}$ has arity at most two and subsequently, our new notion of width for GHDs. We conclude this section with our results on Chain Matrix-Vector Multiplication (MCM).

| Query | $G$ | $d$, $r$ | Gap | Ref |
|---|---|---|---|---|
| FAQ | L | $O(1)$, $O(1)$ | $\widetilde{O}(1)$ | Thm 5.1 |
| FAQ | A | $O(1)$, $O(1)$ | $\widetilde{O}(1)$ | Thm 5.1 |
| BCQ | A | $d$, 2 | $\widetilde{O}(d)$ | Thm 4.1 |
| FAQ | A | $d$, $r$ | $\widetilde{O}(d^2r^2)$ | Thm 5.2 |
| MCM* | L | 1, 2 | $O(1)$ | Sec 6 |

**Table 1: The first and second columns denote the query that we compute and topology on which the query is computed. In the second column, $L$ denotes a line and $A$ denotes an arbitrary $G$. The third column denotes the degeneracy (Definition 3.3) and arity conditions $(d, r)$. The fourth column denotes the gap between our upper and lower bounds ignoring polylogarithmic factors in $N$ and $G$ (denoted by $\widetilde{O}$). The final column denotes the relevant result in this paper. Note that all our results except MCM (denoted by a '*') assume worst-case assignment of functions in the Query to nodes in $G$.**

## 2.1 Our Model

We first define our model.

Model 2.1. *We are given a query $q$, its underlying hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ with input functions $f_e$ (having at most $N$ non-zero values) for every $e \in \overline{\mathcal{E}}$ and a topology $G = (V, E)$. Further, each function is completely assigned to a unique node in $V$. It follows that there exists a subset $K : K \subseteq V$ that contains the players with functions and $|K| \leq k = |\overline{\mathcal{E}}|$. We assume $N \geq |V(G)|^2$ and consider worst-case inputs for the functions.*
*We would like to compute* BCQ *(and more generally an* FAQ*) of $\mathcal{H}$ on $G$. To design a protocol for this computation, we assume that every node in $G$ has the knowledge of $\mathcal{H}$ and $G$. In each round of the protocol, at most $O(r \cdot \log_2(D))$ bits can be communicated over every edge in $E$. In particular, this implies any subset of edges in $G$ can communicate in the same round. Further, at the end of the protocol, a pre-determined player in $K$ has the answer to $q$.*
*Finally, given the above setup, our goal is to design protocols that minimize the total number of rounds needed to compute $q$ assuming worst-case assignment of the functions to players in $G$. Note that we do not take into account the internal computation done by nodes in $G$ and we assume that all nodes are always available in $V$ (i.e., node failures do not happen) and they cooperatively compute the answer to $q$.*

We prove both upper and lower bounds on the total number of rounds needed to compute $q$ on $G$ for every query hypergraph $\mathcal{H}$ and every topology $G$. While our upper bounds hold for any assignment of input functions to players in $G$, our lower bounds hold for a specific class of worst-case assignments of input functions to players in $G$. In Section **??**, we further discuss the assumptions on $\mathcal{H}$ and $G$ in the above model.
Before we move to our results for the case when $\mathcal{H}$ has arity at most two, we would like to point out that our bounds do not assume that the size of $q$ is negligible compared to $N$, which is a standard assumption for computing database queries. Thus, our results are more general and in particular, for applications

[7] Degeneracy is defined as the smallest $d$ such that every sub(hyper)graph in $\mathcal{H}$ has a vertex of degree at most $d$.
[8] An internal node is a non-leaf nodes in a GHD.
[9] $\mathbb{F}_2$ has two elements: the additive identity **0** and multiplicative identity **1**. Addition, and Multiplication are all modulo 2.
[10] We illustrate this difference in the full paper [47].
[11] In neural networks, a non-linear function is applied after each matrix-vector multiplication and the multiplication is over reals instead of $\mathbb{F}_2$.

in PGMs, this is necessary since the size of $q$ cannot be assumed as negligible w.r.t. $N$.

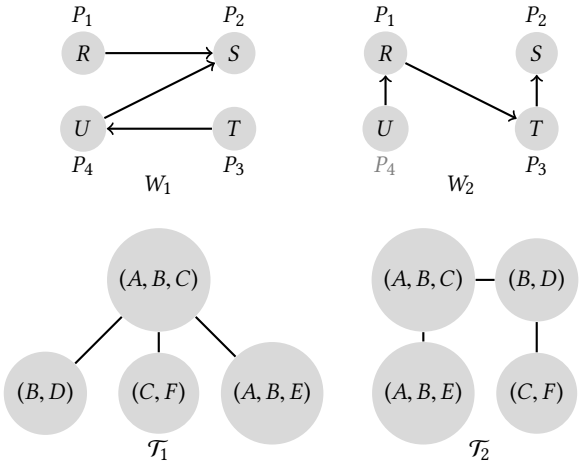


**Figure 2: Two directed paths $W_1$ and $W_2$ for $G_2$ and two GHDs $\mathcal{T}_1$ (with 1 internal node) and $\mathcal{T}_2$ (with 2 internal nodes) both rooted at $(A, B, C)$ for $\mathcal{H}_2$. $G_2$ and $\mathcal{H}_2$ are in Figure 1.**

## 2.2 Arity Two

We consider the case when $\mathcal{H}$ has arity at most two and illustrate our upper and lower bound techniques through examples.

*2.2.1 Upper Bounds.* We start with a trivial protocol to compute any query $\mathcal{H}$ on any $G$. We then show how to improve upon it when $\mathcal{H}$ has a special structure. We use two extremal instances of $G$ for an easy exposition of our results – a line (least connectivity) and a clique (full connectivity). We refer the reader to Figure 1 for all examples (except $\mathcal{H}_0$) considered in this section.

*Trivial Protocol.* There is always a *trivial protocol* to solve any query $\mathcal{H}$ on any $G$ in which all players send their functions to one designated player who then computes the answer.

. We consider the topologies $G_1$ and $G_2$ from Figure 1. We first start by computing a toy query $\mathcal{H}_0$ on $G_1$.

*Example 2.1.* Consider the query hypergraph $\mathcal{H}_0 = (\overline{\mathcal{V}} = \{A\}, \overline{\mathcal{E}} = \{R(A), S(A), T(A), U(A)\})$ i.e., all edges are self-loops on $A$ and the line $G_1$. We would like to solve BCQ of $\mathcal{H}_0$ on $G_1$, which in Datalog format is $q_0() : -R(A), S(A), T(A), U(A)$. In $G_1$, player $P_1$ gets $R$, $P_2$ gets $S$, $P_3$ gets $T$ and $P_4$ gets $U$. Then, solving BCQ of $\mathcal{H}_0$ on $G_1$ is equivalent to checking if the set-intersection $R(A) \cap S(A) \cap T(A) \cap U(A)$ is empty. Let's assume that player $P_4$ needs to know the answer for this query. We can solve this query in $N + 2$ rounds as follows. In the first round, player $P_1$ sends a value $a \in \text{Dom}(A)$ such that there exists $R(a) = 1$ to player $P_2$ who then checks if $S(a) = 1$. More generally, in the $i$-th round, player $P_j$ for $2 \le j \le 4$ receives an $a$ from its left neighbor $(j-1)$ and checks if $a$ is present in its table. If so, it passes $a$ to its right neighbor $(j+1)$ (if $j \le 3$) in the next $(i+1)$-th round. Otherwise, it does not pass anything. Notice that this protocol will terminate once all matching values of $a$ are passed from $P_1$ to $P_4$ which takes $N+2$ rounds in the worst case. In other words, we are computing the semijoin (see Definition 3.5) query $((R(A) \ltimes S(A)) \ltimes T(A)) \ltimes U(A)$, which is equivalent to computing $R(A) \cap S(A) \cap T(A) \cap U(A)$. Note that this is much better than the *trivial protocol* for this case, which takes $3 \cdot N + 2$ rounds.
At the end of this protocol, $P_4$ knows the answer to the query. It is not too hard to see that we can extend the above protocol to the case when any other player say $P_i$ for some $i \in [3]$ is designated to know the answer. In particular, we can orient $G_1$ in such a way that all paths are directed towards $P_i$ and then run the protocol above simultaneously on all paths (there are at most two) towards $P_i$ (recall that we assume knowledge of $G$ for all nodes). Note that $P_i$ would have the answer to the query and the new protocol takes $N + x$ rounds, where $x \le 2$ depends on the choice of $P_i$.

It is not too hard to see that our protocol in the above example can be extended to the case when $\mathcal{H}$ is a star. We illustrate this in the following example.

*Example 2.2.* Consider the star $\mathcal{H}_1$ and the line $G_1$ in Figure 1. We would like to solve BCQ of $\mathcal{H}_1$ on $G_1$, which in Datalog format is $q_1() : -R(A, B), S(A, C), T(A, D), U(A, E)$. In $G_1$, player $P_1$ gets $R$, $P_2$ gets $S$, $P_3$ gets $T$ and $P_4$ gets $U$. Then, BCQ of $\mathcal{H}_1$ is 1 iff $\pi_A(R) \cap \pi_A(S) \cap \pi_A(T) \cap \pi_A(U)$ is non-empty and 0 otherwise. HEre, $\pi_A(\cdot)$ denotes the projection onto attribute $A$. We assume $P_2$ needs to know the answer for this query.
We can solve this query in $N + 2$ rounds using the same protocol as in Example 2.1. In other words, we are computing the semjoin query[12] $((\pi_A(R) \ltimes \pi_A(S)) \ltimes \pi_A(T)) \ltimes \pi_A(U)$. Note that each node needs to compute $\pi_A(\cdot)$ internally for this computation but this doesn't need any communication between the nodes. At the end of this protocol, $P_2$ knows the answer to the query.

We now show how to do the same computation (i.e., BCQ of $\mathcal{H}_1$) on $G_2$.

*Example 2.3.* Consider the star $\mathcal{H}_1$ and the clique $G_2$ in Figure 1. We would like to compute BCQ of $\mathcal{H}_1$ on $G_2$, which in Datalog format is same as $q_1$ from Example 2.2. In $G_2$, player $P_1$ gets $R$, $P_2$ gets $S$, $P_3$ gets $T$ and $P_4$ gets $U$. We assume that Dom$(A)$ is split into two halves and $P_2$ needs to know the answer for this query.
We can solve this query in $\frac{N}{2}+2$ rounds as follows. We consider the two edge-disjoint directed paths $W_1$ and $W_2$ (see Figure 2) on $G_2$ that end with $P_2$. Our protocol from Example 2.2 runs on both these paths simultaneously with one caveat – the values of $a$ in the first half of Dom$(A)$ are sent through $W_1$ and the ones in the second half of Dom$(A)$ are sent through $W_2$. Since both these directed paths involve the same set of nodes, our protocol is valid and takes only $\frac{N}{2}+2$ rounds as claimed above. Note that this is better than our bound in Example 2.2.

The protocols in Examples 2.2 and 2.3 can be generalized to solve any star $\mathcal{H}$ on any $G$. Given the protocol for a star, there

[12]We would like to mention that casting the computation of BCQ on a star query as a semijoin is well-known [42].

is a natural extension to $\mathcal{H}$ being a tree (or more generally a forest): we handle all the stars of the tree in a bottom-up fashion (starting with the stars at the "end" of the tree) and recurse. In particular, we can apply our protocol for the star case as a black-box on each of these stars. To extend this result to general $d$-degenerate graphs $\mathcal{H}$, we first decompose $\mathcal{H}$ into a forest and a *core* that contains the roots of all trees in the forest and all remaining vertices not in the forest. We run the above protocol on the forest and use the *trivial protocol* on the core. For general $G$, note that we need to find optimal ways of applying these protocols – for the forest part, we extend the idea of a paths packing from Example 2.3 to a Steiner tree (Definition 3.8) packing and for the *trivial protocol*, we use standard ideas from network flows (Definition 3.12). We would like to mention here that our upper bounds hold even when more than one function is assigned to a player (i.e., $|K| < k$). We will crucially exploit this fact in our lower bounds. We present more details in Section 4.1.
We are now ready to talk about our lower bounds.

*2.2.2 Lower Bounds.* All our lower bounds follow from known lower bounds on the (well-studied) TRIBES function (see [18] and references therein) in two-party communication complexity literature. To this end, we first consider an arbitrary TRIBES instance of a specific size and show that it can be reduced to a suitable two-party BCQ instance. In particular, solving the two-party BCQ instance (we constructed) indeed solves the TRIBES instance (we started with). Thus, known lower bounds on TRIBES implies lower bounds for BCQ. Finally, we generalize our results from the two-party setting to general $G$ using ideas from graph theory and exploit the fact that our (upper and) lower bounds are for worst-case input functions and worst-case assignments of input functions to players in $G$.
We start by defining the two-party communication complexity model as a special case of Model 2.1.

Model 2.2. *Consider two players Alice (a) and Bob (b) on a graph $\mathcal{G} = (V = \{a, b\}, E = \{(a, b)\})$ with strings $\bar{X} = (X_1, \ldots, X_m)$ and $\bar{Y} = (Y_1, \ldots, Y_m)$, where $X_i, Y_i \in \{0,1\}^N$. Further, Alice gets $\bar{X}$, Bob gets $\bar{Y}$ and both have knowledge of only their inputs. The goal for these two players is to compute the boolean function $f(\bar{X}, \bar{Y}) : \left(\{0,1\}^{m \cdot N}, \{0,1\}^{m \cdot N}\right) \to \{0,1\}$. The randomized two-party communication complexity of computing $f$, denoted by $\mathcal{R}(f(\bar{X}, \bar{Y}), \mathcal{G}, \{a, b\})$, is defined as the minimum worst-case number of rounds[13] needed by a randomized protocol that deterministically computes $f(\bar{X}, \bar{Y})$ with error at most $\frac{1}{3}$.*

We would like to mention that considering the randomized two-party communication complexity over its deterministic counterpart makes our lower bounds only stronger. We define TRIBES and state the lower bound result that we will use in arguments.

Theorem 2.3 (Jayram et. al [36]). *Let* $\text{TRIBES}_{m,N}(\bar{X}, \bar{Y}) \equiv \bigwedge_{i=1}^{m} \text{DISJ}_N(X_i, Y_i)$, *where* $\text{DISJ}_N(X_i, Y_i)$ *is* 1 *if* $X_i \cap Y_i \neq \emptyset$ *and* 0 *otherwise,* $X_i, Y_i \in \{0,1\}^N$ *for every* $i \in [m]$ *and* $\bar{X} = (X_1, \ldots, X_m)$, $\bar{Y} = (Y_1, \ldots, Y_m)$. *Note that in the two-party model, Alice gets* $\bar{X}$ *and Bob gets* $\bar{Y}$. *Given this setup, we have*

$$\mathcal{R}\left(\text{TRIBES}_{m,N}(\bar{X}, \bar{Y}), \mathcal{G}, \{a, b\}\right) \geq \Omega(m \cdot N).$$

We start with an arbitrary TRIBES instance $\text{TRIBES}_{m,N}(\bar{X}, \bar{Y})$ of a suitable size and show that it can be reduced to a suitable two-party BCQ instance $\text{BCQ}_{\mathcal{H},\bar{X},\bar{Y}}$. Note that $m$ is a function of $\mathcal{H}$. In particular, such a reduction would imply

$$\begin{aligned}\mathcal{R}\left(\text{BCQ}_{\mathcal{H},\bar{X},\bar{Y}}, \mathcal{G}, \{a, b\}\right) &\geq \mathcal{R}\left(\text{TRIBES}_{m,N}(\bar{X}, \bar{Y}), \mathcal{G}, \{a, b\}\right) \\ &\geq \Omega(m \cdot N),\end{aligned}$$

where the final inequality follows from Theorem 2.3. The above inequality implies the following since we consider worst-case input functions for a fixed $\mathcal{H}$.

Corollary 2.4.

$$\mathcal{R}\left(\text{BCQ}_{\mathcal{H},N}, \mathcal{G}, \{a, b\}\right) \geq \mathcal{R}\left(\text{BCQ}_{\mathcal{H},\bar{X},\bar{Y}}, \mathcal{G}, \{a, b\}\right),$$

*where* $\text{BCQ}_{\mathcal{H},N}$ *denote the class of problems where all functions in* $\mathcal{H}$ *have size at most* $N$.

We generalize the above result to any $G$ using ideas from graph theory. We consider an appropriate cut $C = (A, B)$ of $G$ that partitions $V$ into two vertex-disjoint subsets $A$ and $B$ and a correponding assignment, where each function $e \in \overline{\mathcal{E}}(\mathcal{H})$ is assigned to a node in either $A$ or $B$. Since this is a valid assignment of functions in $\mathcal{H}$ to players in $G$, the minimum number of rounds needed to compute an instance of $\text{BCQ}_{\mathcal{H},N}$ on $G$ assuming worst-assignments of functions to players in $K$, denoted by $\mathcal{R}\left(\text{BCQ}_{\mathcal{H}}, G, K\right)$, is at least $\mathcal{R}\left(\text{BCQ}_{\mathcal{H},\bar{X},\bar{Y}}, \mathcal{G}, \{a, b\}\right)$. We reconsider $\mathcal{H}_1$ and $G_1$ from Example 2.2 here.

*Example 2.4.* Recall we proved an upper bound of $N + 2$ for computing BCQ of $\mathcal{H}_1$ on $G_1$. We start with an arbitrary $\text{TRIBES}_{m=1,N}(\bar{X} = (X_1), \bar{Y} = (Y_1))$ instance. With a slight abuse of notation, we treat $X_1, Y_1$ as subsets of $[N]$ (instead of elements in $\{0,1\}^N$). We now construct a corresponding $\text{BCQ}_{\mathcal{H}_1,\bar{X},\bar{Y}}$ instance from the TRIBES one as follows – we assign $R(A, B) = X_1 \times \{1\}$, $S(A, C) = T(A, D) = [N] \times \{1\}$ and $U(A, E) = Y_1 \times \{1\}$. It is not too hard to see that $\text{BCQ}_{\mathcal{H}_1,\bar{X},\bar{Y}}$ is 1 iff $\text{TRIBES}_{1,N}(\bar{X}, \bar{Y})$ is 1, implying that solving the BCQ instance would solve the TRIBES instance. Finally, to obtain a lower bound for computing $\text{BCQ}_{\mathcal{H}_1,\bar{X},\bar{Y}}$ on the line $G_1$, we only need a cut where $R$ and $U$ are on different sides. We consider the cut $C = (\{P_1, P_2\}, \{P_3, P_4\})$ of $G_1$ and the assignment where $P_1$ gets $R$, $P_2$ gets $S$, $P_3$ gets $T$ and $P_4$ gets $U$. Then, we can use Lemma 2.4 and Theorem 2.3 to obtain the required lower bound of $\Omega(N)$. It's not too hard to see that the above lower bound holds for any star $\mathcal{H}$. The same TRIBES instance can be used for Examples 2.1 and 2.3 as well. While a similar assignment holds for Example 2.1, Example 2.3 requires a slightly different assignment since we use a different cut.

For general $d$-degenerate graphs $\mathcal{H}$, we start by recalling that $m$ (i.e., size of the TRIBES instance) is a function of $\mathcal{H}$. As mentioned in Section 2.2.1, we can decompose $\mathcal{H}$ into a forest and a *core*. We prove three different lower bounds on $\mathcal{H}$, where the size of the TRIBES instance $m$ used in our reduction is the maximum of three different bounds, each one on a different part of $\mathcal{H}$. The first one is on $\mathcal{H}$'s forest part, the second and

[13] In each round, we assume at most one bit is sent from $a$ to $b$ instead of $O(\log_2(r \cdot D))$ bits to be consistent with the two-party communication complexity literature.

third ones are on $\mathcal{H}$'s *core* part – lower bounded by applying Moore's bound [5] and Turan's theorem [6] respectively. Note that this covers *Step 1* and later we show functional equivalence with a corresponding BCQ instance on $\mathcal{H}$ covering *Step 2*. Finally, for general $G$ in *Step* 3, we use ideas from [18] to obtain an appropriate cut for $G$ and use lower bounds from the induced two-party communication complexity problem across the cut. Note that the assignment of functions depends on the cut. We present the details in Section 4.2.

For constant $d$, our upper and lower bounds match. However, for non-constant $d$, we have a gap of $\widetilde{O}(d)$. We would like to note that there is a fundamental bottleneck in getting rid of this factor as the case of $\mathcal{H}$ being a clique is an outstanding open question[14] (even in Model 2.2) and seems beyond the reach of current communication complexity techniques [16].

## 2.3 Notion of Width

We start by defining the notion of *GHD*s and acyclic (hyper)graphs.

Definition 2.5 (GHD). *A GHD of $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is defined by a triple $\langle \mathcal{T}, \chi, \lambda \rangle$, where $\mathcal{T} = (V(\mathcal{T}), E(\mathcal{T}))$ is a tree, $\chi : V(\mathcal{T}) \to 2^{\mathcal{V}}$ is a function associating a set of vertices $\chi(v) \subseteq \mathcal{V}$ to each node $v$ of $\mathcal{T}$, and $\lambda : V(\mathcal{T}) \to 2^{\mathcal{E}}$ is a function associating a set of hyperedges to each node $v$ of $\mathcal{T}$ such that the following two properties hold. First, for each $e \in \mathcal{E}$, there is at least one node $v \in V(\mathcal{T})$ such that $e \subseteq \chi(v)$ and $e \in \lambda(v)$. Second, for every $V' \subseteq \mathcal{V}$, the set $\{v \in V(\mathcal{T}) | V' \subseteq \chi(v)\}$ is connected in $\mathcal{T}$, called the running intersection property (RIP hereon). We only consider rooted GHDs.*

*A* reduced-GHD *has the additional property that every hyperedge $e \in \overline{\mathcal{E}}$ has a unique node $v \in V(\mathcal{T})$ such that $\chi(v) = e$ (note that this is an equality).*

Definition 2.6 (Acyclicity). *A hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ is acyclic iff there exists a GHD $(\mathcal{T}, \chi, \lambda)$ in which for every node $v \in V(\mathcal{T})$, $\chi(v)$ is a hyperedge in $\overline{\mathcal{E}}$.*

We now define the sub-classes of reduced-GHDs that we consider in this paper. In particular, we construct reduced-GHDs using the GYO-Elimination order [31, 59, 66] and call them GYO-GHDs. We start by defining the *GYO-reduction* of a hypergraph $\mathcal{H}$.

Definition 2.7 (GYO-reduction). *For any hypergraph $\mathcal{H}$, the GYO-reduction is defined as the leftover hypergraph after running the GYO algorithm (GYOA) [31, 59, 66] on $\mathcal{H}$. For acyclic hypergraphs $\mathcal{H}$, the GYO-reduction results in an empty hypergraph.*

We now define $\mathrm{C}(\mathcal{H})$ and $\mathrm{F}(\mathcal{H})$ based on the *GYO-reduction* of $\mathcal{H}$.

Definition 2.8 ($\mathrm{C}(\mathcal{H}), \mathrm{F}(\mathcal{H})$). *$\mathrm{C}(\mathcal{H})$ is the union of the* GYO-reduction *of $\mathcal{H}$ and all roots in the forest of acyclic hypergraphs generated by running GYOA on $\mathcal{H}$. $\mathrm{F}(\mathcal{H})$ is the output generated by running GYOA on $\mathcal{H}$ minus $\mathrm{C}(\mathcal{H})$.*

We are now ready to construct GYO-GHDs.

[14] We state this problem formally in the longer version.

Construction 2.9. *Let $\mathcal{T}$ be the GYO-GHD be obtained from this procedure. We define the root $r'$ of $\mathcal{T}$ with $\chi(r') = V(\mathrm{C}(\mathcal{H}))$. For each edge $e \in \overline{\mathcal{E}}$ with $e \subset V(\mathrm{C}(\mathcal{H}))$, we create a new node $v'_e$ in $\mathcal{T}$ with $\chi(v'_e) = e$ and add the edge $(r', v'_e)$ to $\mathcal{T}$ in order to make it a reduced-GHD. Note that we do not enforce any constraints on the remaining edges in $\mathcal{T}$ as long as it remains a valid GHD.*

We argue that the above procedure produces a reduced-GHD in the full paper [47]. Our new notion of width based on GYO-GHDs, which we call $y$ (Internal Node Width), is defined as follows.

Definition 2.10.

$$y(\mathcal{H}) = \min_{\forall \mathcal{T} : \mathcal{T} \text{ is a GYO-GHD of } \mathcal{H}} y(\mathcal{T})$$

*where $y(\mathcal{T})$ is the number of internal/non-leaf nodes in $\mathcal{T}$.*

**Unless specified otherwise, in the rest of the paper when we refer to GHDs, we are referring to GYO-GHDs.** As an example in Figure 2, we consider two different GHDs $\mathcal{T}_1$ and $\mathcal{T}_2$ for the acyclic hypergraph $\mathcal{H}_2$ from Figure 1. Both are outcomes of Construction 2.9 and while $\mathcal{T}_2$ has two internal nodes, $\mathcal{T}_1$ has only one, implying $y(\mathcal{H}) = 1$. For $\mathcal{H}_1$ in Figure 1, it is easy to construct a GHD with one internal node (i.e., $y(\mathcal{H}) = 1$) by keeping $(A, B)$ as the root and $(A, C)$, $(A, D)$, $(A, E)$ as leaves. We show how this can be achieved for simple graphs $\mathcal{H}$ in Section 4.

## 2.4 Chain Matrix-Vector Multiplication

Finally, in this work, we consider the problem of computing $\mathbf{A}_k \cdots \mathbf{A}_1 \mathbf{x}$ where the computation is over $\mathbb{F}_2$. The player $P_i$ gets $\mathbf{A}_i$ for $i \in [k]$ and $P_0$ gets $\mathbf{x}$. Player $P_{k+1}$ wants to know the answer (and does not have any input). The topology $G$ is a line with $P_i$ connected to $P_{i+1}$ for $0 \leq i \leq k$. We show that when $k \leq N$ the natural algorithm that computes the partial product $\mathbf{A}_i \cdots \mathbf{A}_1 \mathbf{x}$ at $P_i$ taking $\Theta(kN)$ rounds is indeed optimal. By contrast, if the matrices are assigned randomly to the players then the optimal number of rounds is $\Theta(k^2 N)$ (this follows from a trivial protocol). On the other extreme, if all matrices are assigned to one player, then the problem is trivial. So we are proving a tight lower-bound for arguably the simplest assignment of matrices to players that is not trivial.

We note that the existing technique of [18] cannot prove a lower bound better than $\Omega(N)$ for this problem (see the full paper [47] for a more detailed description). To get a better lower bound of $\Omega(kN)$, we use an entropy based inductive argument to show that at end of the $\Omega(iN)$ rounds, in player $P_i$'s view, $\mathbf{A}_{i-1} \cdots \mathbf{A}_1 \mathbf{x}$ has very high entropy. However, Shannon's entropy is too weak for this argument to go through and we use the stronger notion of min-entropy, which is omnipresent in pseudorandomness and cryptography [62]. Unfortunately, this means that we can no longer appeal to the chain rule and the arguments become a bit more delicate. Finally, in the process we prove the following natural result: if $\mathbf{A}$ and $\mathbf{x}$ have high enough min-entropy, then $\mathbf{Ax}$ has higher min-entropy than

$\mathbf{x}$. To the best of our knowledge this result is new, though it follows by combining known results in pseudorandomness.[15]

## 3 PRELIMINARIES AND NOTATION

*Query (Hyper)graph $\mathcal{H}$.*

Definition 3.1 ($n_2(\mathcal{H})$). *Using Construction 2.9, we can decompose any $\mathcal{H}$ into a core $\mathrm{C}(\mathcal{H})$ and a forest $\mathrm{F}(\mathcal{H})$. We define $n_2(\mathcal{H}) = V(\mathrm{C}(\mathcal{H}))$.*

Definition 3.2 (Degree). *The degree of a vertex $v \in \mathcal{H}$ is given by $\left|\{e \ni v : e \in \overline{\mathcal{E}}\}\right|$.*

Definition 3.3 ($d$-degenerate (hyper)graph [43]). *In a $d$-degenerate (hyper)graph every sub(hyper)graph has a vertex of degree at most $d$.*

We now define *natural join* and *semijoin*.

Definition 3.4 (Natural Join). *The* join $J = \bowtie_{e \in \overline{\mathcal{E}}} R_e$ *is a relation $J$ with attribute set $V(\mathcal{H})$ satisfying the following condition (where $\bowtie$ denotes the* join *operator). A tuple $\mathbf{t} \in J$ iff for every $e \in E(\mathcal{H})$, the* projection *of $\mathbf{t}$ onto attributes in $v(e)$ - denoted by $\pi_{v(e)}(\mathbf{t})$ - belongs to $R_e$. Note that $J \subseteq \prod_{v \in V(\mathcal{H})} \mathrm{Dom}(v)$.*

Definition 3.5 (Semijoin). *A semijoin $J' = R_1 \ltimes R_2$ of relations $R_1$ and $R_2$ is defined as $J' = R_1 \bowtie \pi_{\mathrm{attr}(R_1) \cap \mathrm{attr}(R_2)}(R_2)$, where $\mathrm{attr}(\cdot)$ denotes the attribute set of the relations and $\ltimes$ is the semijoin operator.*

We show in the full paper [47] that *natural join* and *semijoin* are special cases of FAQ.

*Network Topology $G$.* We define some standard graph notions that will be used throughout the paper.

Definition 3.6 ($\mathrm{MinCut}(G, K)$). *We denote the size of the minimum cut of $G$ separating vertices in $K$ by* $\mathrm{MinCut}(G, K)$.

Definition 3.7 (Star Graph). *A star is a tree on $n$ vertices with one internal node and $n-1$ leaves (e.g. $\mathcal{H}_1$ in Figure 1).*

Definition 3.8 (Steiner tree). *Given a graph $G = (V, E)$ and a set of nodes $K \subseteq V$, we call a tree $T$ a Steiner tree if it connects all vertices in $K$ only using edges in $E$.*

In particular, we are interested in Steiner trees with diameter at most $\Delta$ (i.e., distance between any two nodes in $K$). Let $\mathcal{T}_{\Delta, K}$ denote the set of all such Steiner trees.

Definition 3.9 ($\mathrm{ST}(G, K, \Delta)$). $\mathrm{ST}(G, K, \Delta)$ *denotes the maximum number of edge disjoint Steiner trees from $\mathcal{T}_{\Delta, K}$ in $G$.*

We will need this result:

Theorem 3.10 ([48]). $\mathrm{ST}(G, K, |V(G)|) = \Omega(\mathrm{MinCut}(G, K))$.

Finally, we state a recent result under Model 2.1 on set-intersection queries over any topology $G$ and any subset of players $K \subseteq V : |K| \leq k$, which we will use frequently in our arguments.

[15] We thank David Zuckerman for showing us the high level proof idea of this result.

Theorem 3.11 ([18]). *Let $\mathbf{x}_u \in \{0,1\}^N$ for every player $u \in K$. The number of rounds taken by a protocol that deterministically computes $\bigwedge_{u \in K} \mathbf{x}_u$ (where the $\wedge$ is bit-wise AND) is given by $\Theta\left(\min_{\Delta \in [|V|]} \left(\frac{N}{\mathrm{ST}(G,K,\Delta)} + \Delta\right)\right)$.*

We will use the following notation for a special case of a multi-commodity flow problem:

Definition 3.12. *For every graph $G$, subset of players $K$ and integer $N' \geq 0$, let $\tau_{\mathrm{MCF}}(G, K, N')$ be the minimum number of rounds needed to route $N' \log_2(N')$ bits from all players in $K$ to any one player in $K$.*[16]

Let the minimum number of rounds taken by a protocol to determinstically compute BCQ of $\mathcal{H}$ on $G$ be denoted by $\mathcal{D}(\mathrm{BCQ}_{\mathcal{H},N}, G, K)$, where each function in $\mathcal{H}$ has size at most $N$ and is assigned to some player in $K \subseteq V, |K| \leq k$. The *trivial protocol* along with Definition 3.12 implies the following.

Lemma 3.13.

$$\mathcal{D}(\mathrm{BCQ}_{\mathcal{H},N}, G, K) = O\left(\tau_{\mathrm{MCF}}(G, K, k \cdot r \cdot N)\right).$$

### 3.1 Asymptotic Notation

For notational clarity, in our lower bounds, we will ignore the factor $\log_2(N) \cdot \log_2(\mathrm{MinCut}(G, K)) \cdot \log_2(n_2(\mathcal{H}))$. Further, we ignore these factors while arguing for the tightness of our bounds, which we denote by $\widetilde{\Omega}(\cdot), \widetilde{O}(\cdot)$ and $\widetilde{\Theta}(\cdot)$.

## 4 $\mathcal{H}$ IS A DEGENERATE SIMPLE GRAPH

In this section, we consider the class of queries $\mathrm{BCQ}_{\mathcal{H},N}$ for a given $d$-degenerate graph $\mathcal{H}$ with arity $r$ at most two and all functions have size at most $N$. We prove upper and lower bounds that are tight within a factor of $\widetilde{O}(d)$ for computing any query in $\mathrm{BCQ}_{\mathcal{H},N}$. The following is our main result.

Theorem 4.1. *For arbitrary topology $G$, subset of players $K$ and $d$-degenerate simple graph $\mathcal{H}$, we have*

$$\mathcal{D}(\mathrm{BCQ}_{\mathcal{H},N}, G, K) = O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N}{\mathrm{ST}(G,K,\Delta)} + \Delta\right)\right) + O\left(\tau_{\mathrm{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot N)\right). \quad (1.1)$$

*Further, for all simple graphs $\mathcal{H}$, we have*

$$\mathcal{R}(\mathrm{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot N}{\mathrm{MinCut}(G,K)}\right) + \widetilde{\Omega}\left(\frac{n_2(\mathcal{H}) \cdot N}{\mathrm{MinCut}(G,K)}\right). \quad (1.2)$$

We would like to point out that our upper bound holds for every assignment of the functions $f_e$ to players in $K$ while our lower bound holds for some assignment of functions to players in $K$. We first prove the upper bound (1.1), followed by the lower bound (1.2). Finally, we argue how our bounds are tight within a gap of $\widetilde{O}(d)$.

### 4.1 Upper Bound

We first consider the case when $\mathcal{H}$ is a star, which will be a basic building block for our algorithms for general $\mathcal{H}$.

[16] Here, we will consider the worst-case over all possible ways the $N' \log_2(N')$ bits are distributed over $K$. While our upper bounds can be smaller than this, we use this worst-case measure to simplify our bounds.

*4.1.1 $\mathcal{H}$ is a star.* Let $P = (v_0, v_1, \ldots, v_k)$ be the vertices of the star with $v_0$ as it's center. In this case, $\mathcal{H}$ includes $k$ relations of the form $R_{v_0,v_i}$ for every $i \in [k]$. Note that computing the corresponding BCQ query $q$ can be solved via a *set-intersection* problem where we compute $R'_P = \bigcap_{i=1}^{k} R'_{v_i}$, where $R'_{v_i} = \{a_0 | (a_0, a_i) \in R_{v_0,v_i} \text{ for some } a_i \in \text{Dom}(v_i)\}$. It is easy to see that the final output of $q$ is 1 if $R'_P \neq \emptyset$ and 0 otherwise. We can solve the resulting set intersection problem using Theorem 3.11 to compute $R'_P$. The procedure to compute $R'_P$ is described in Algorithm 1, which when combined with the fact that at most $O(\log_2(D))$ bits can be communicated in each round, implies the following result.

Corollary 4.2. *When $\mathcal{H}$ is a star, for arbitrary graphs $G$ and subset of players $K$, we have*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O\left(\min_{\Delta \in [|V|]} \left(\frac{N}{\text{ST}(G,K,\Delta)} + \Delta\right)\right).$$

For the case when $G$ is a line with $k$ vertices, note that $\text{ST}(G,K,\Delta) = 0$ for every $\Delta > k-1$ and $\text{ST}(G,K,k-1) = 1$, which in turn implies the following.

Corollary 4.3. *Let $\mathcal{H}$ be a star and $G$ be a line with $k$ vertices. Then*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O(N + k).$$

Note that the above result is a generalization of Example 2.2.

**Algorithm 1** Algorithm for Star

1: **Input:** A star query with attributes $P = (v_0, \ldots, v_k)$ and relations $\{R_{(v_0,v_i)} : i \in [k]\}$. Note that $v_0$ is the center.
2: **Output:** $R'_P$
3: Each player containing a relation $R_{v_0,v_i}$ computes $R'_{v_i} = \{a_0 | (a_0, a_i) \in R_{v_0,v_i} \exists a_i \in \text{Dom}(v_i)\}, \forall i \in [k]$ internally.
4: $R'_P = \bigcap_{i=1}^{k} R'_{v_i}$ is computed using Theorem 3.11.
5: **return** $R'_P$

*4.1.2 $\mathcal{H}$ is a forest.* We now use the above idea to obtain upper bounds for the case when $\mathcal{H}$ is a forest.

Lemma 4.4. *For arbitrary $G$, subset of players $K$ and $\mathcal{H}$ being a forest, we have*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N}{\text{ST}(G,K,\Delta)} + \Delta\right)\right). \quad (1)$$

Proof Sketch. We keep removing stars from trees in $\mathcal{H}$ in a bottom-up fashion and solve the induced query on each removed star using Algorithm 1. Since the number of stars we remove in this process is $y(\mathcal{H})$, the stated bound follows. The details are in the full paper [47]. □

*4.1.3 The general case: $d$-degenerate graphs.* We now state our upper bound when $\mathcal{H}$ is a $d$-degenerate simple graph:

Lemma 4.5. *For arbitrary $G$, subset of players $K$, and any $d$-degenerate simple graph $\mathcal{H}$, we have*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N}{\text{ST}(G,K,\Delta)} + \Delta\right)\right) + O\left(\tau_{\text{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot N)\right). \quad (2.1)$$

Proof Sketch. We decompose $\mathcal{H}$ into two components via Construction 2.9 – forest ($\text{F}(\mathcal{H})$) and core ($\text{C}(\mathcal{H})$). We then use Lemma 4.4 to solve the induced query on $\text{F}(\mathcal{H})$. For the core, we use the *trivial protocol* of sending all the remaining relations to one player. The details are in the full paper [47]. □

## 4.2 Lower Bound

We start with an overview, followed by lower bounds for the case when $\mathcal{H}$ is a forest and conclude with lower bounds for all simple graphs $\mathcal{H}$.

*4.2.1 Overview.* As we showed in Section 2.2.2, we start by considering an arbitrary TRIBES instance of size $m$ where $m$ is a function of $\mathcal{H}$. We then construct a corresponding BCQ instance from it (*Step* 1) and show that solving the BCQ instance (we constructed) indeed solves the TRIBES instance (*Step* 2). We denote *Steps* $1-2$ succinctly by $\text{TRIBES}_{m,N} \leq \text{BCQ}_{\mathcal{H},N}$. Finally, since our lower bounds are for worst-case assignment of functions to players in $G$, we show a specific assignment of functions to players that would help us achieve the required lower bound (*Step* 3).

*4.2.2 $\mathcal{H}$ is a forest.* We prove the following lemma.

Lemma 4.6. *When $\mathcal{H}$ is a forest, we have*

$$\text{TRIBES}_{\frac{y(\mathcal{H})}{2},N} \leq \text{BCQ}_{\mathcal{H},N}.$$

Proof. For notational simplicity, define $y = y(\mathcal{H})$. Given $\mathcal{H}$ and a $\text{TRIBES}_{\frac{y}{2},N}$ instance we design a corresponding $\text{BCQ}_{\mathcal{H},N}$ instance. As $\mathcal{H}$ is bipartite, let $(L,R)$ be the node partition of $\mathcal{H}$ and consider the set $O_L$ ($O_R$ resp.) consisting of all nodes of degree at least two included in $L$ ($R$ resp.). Let $O$ equal the largest of $O_L$ and $O_R$ (i.e., $O$ consists of nodes of odd or even distance from the roots of the forest). Note that $|O| \geq \frac{y}{2}$,[17] and assume w.l.o.g. that the size of $O$ is exactly $\frac{y}{2}$ (otherwise we take a subset of $O$). We associate a pair of sets $(S_o, T_o)$ from $\text{TRIBES}_{\frac{y}{2},N}$ with each node $o \in O$, such that

$$\text{TRIBES}_{\frac{y}{2},N}(\hat{S}, \hat{T}) = \bigwedge_{o \in O} \text{DISJ}_N(S_o, T_o), \quad (2)$$

where $\text{DISJ}_N(S_o, T_o) = 1$ if $S_o \cap T_o \neq \emptyset$ and 0 otherwise.

We now construct a corresponding $\text{BCQ}_{\mathcal{H},N}$ instance in detail. We start by defining a pair of relations corresponding to each pair $(S_o, T_o)$. Let $o \in O$. If $o$ has a parent in $\mathcal{H}$, let $o_p$ be its parent. Let $o_c$ be a child of $o$. We consider the relations $R_{S_o} = S_o \times \{1\}$ and $R_{T_o} = T_o \times \{1\}$, where the attribute set of $R_{S_o}$ is $(o, o_c)$ and that of $R_{T_o}$ is $(o, o_p)$. Here we treat $S_o$ and $T_o$ as subsets of $[N]$ (instead of elements in $\{0,1\}^N$). In the case that $o$ does not have a parent node, it is a root in $\mathcal{H}$

[17] Note that in the arity two case, it's easy to construct a GYO-GHD with $y$ internal nodes using the structure of $\mathcal{H}$. Details in the full paper [47].

with at least two children, and thus we can set $o_p$ to be a child of $o$ that differs from $o_c$. Thus, $\text{TRIBES}_{\frac{y}{2},N}(\hat{S},\hat{T}) = 1$ iff for each $o \in O$, the join $R_{S_o} \bowtie R_{T_o}$ is not empty. To complete the description of the BCQ instance, for each $o \in O$, we associate all additional edges $(o,v)$ adjacent to $o$ in $\mathcal{H}$ with the relation $[N] \times \{1\}$ on attributes $(o,v)$; and remaining edges $(u,v)$ that are not adjacent to any $o \in O$ with the relation $\{1\} \times \{1\}$. Note that no two vertices $o_1, o_2 \in O$ are adjacent in $\mathcal{H}$. Let us denote the BCQ instance constructed above by $q_{\mathcal{H},\hat{S},\hat{T}}$.

To complete the proof, we show that $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\text{TRIBES}_{\frac{y}{2},N}(\hat{S},\hat{T}) = 1$. If $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ then there exists a tuple $\mathbf{t} \in \prod_{v \in V(\mathcal{H})} \text{Dom}(v)$ that satisfies all relations in $q_{\mathcal{H},\hat{S},\hat{T}}$, i.e. $\mathbf{t}_e \in R_e$ for every $e \in \overline{\mathcal{E}}$. Specifically, for each $o \in O$, $R_{S_o} \bowtie R_{T_o}$ is not empty which implies that $\text{TRIBES}_{\frac{y}{2},N}(\hat{S},\hat{T}) = 1$. Alternatively, if $\text{TRIBES}_{\frac{y}{2},N}(\hat{S},\hat{T}) = 1$, we can find a tuple $\mathbf{t} \in \prod_{v \in V(\mathcal{H})} \text{Dom}(v)$ that satisfies all relations in $q_{\mathcal{H},\hat{S},\hat{T}}$. For each $o \in O$ we set $\pi_o(\mathbf{t})$ to be any element in the intersection of $S_o$ and $T_o$, and for all remaining nodes $v$ we set $\pi_v(\mathbf{t}) = 1$. It holds that the relations corresponding to edges of the form $(o,o_p)$, $(o,o_c)$, $(o,v)$, and $(u,v)$ described above are all satisfied. This concludes our proof. □

Note that the above argument was independent of $G$. We now use the structure of $G$ to obtain a lower bound on $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K)$ using known results for $\text{TRIBES}_{\frac{y}{2},N}$.

*Lower bounds dependent on G.* We show the following lower bound for arbitrary $G$, assuming worst-case assignment of relations to players in $K$.

Lemma 4.7. *For any topology $G$ and $\mathcal{H}$ being a forest,*

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot N}{\text{MinCut}(G,K)}\right).$$

Proof. We first consider a min-cut $(A,B)$ of $G$ that separates $K$, where $A$ and $B$ denote the set of vertices in each partition ($A \cup B = V$). Using the notation given in the proof of Lemma 4.6, let $q_{\mathcal{H},\hat{S},\hat{T}}$ be the query corresponding to a given instance $\text{TRIBES}_{\frac{y}{2},N}(\hat{S},\hat{T})$. We assign relations $\{R_{S_o}\}_{o \in O}$ to vertices in $A$ and relations $\{R_{T_o}\}_{o \in O}$ to vertices in $B$. The other relations in $q_{\mathcal{H},\hat{S},\hat{T}}$ can be assigned arbitrarily. Note that any protocol to compute $q_{\mathcal{H},\hat{S},\hat{T}}$ on $G$ gives a two-party protocol (Alice, Bob) for $\text{TRIBES}_{\frac{y}{2},N}$. In particular, Alice gets the sets $\{S_o\}_{o \in O}$ (corresponding to $R_{S_o}$) assigned to vertices in $A$ and Bob gets the sets $\{T_o\}_{o \in O}$ (corresponding to $R_{T_o}$) assigned to vertices in $B$ (ignoring the additional relations). It follows that if there exists a $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K)$ round protocol on $G$, then we have a two-party protocol (i.e., on a graph $\mathcal{G} = (\{a,b\},(a,b))$) with at most $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \cdot \text{MinCut}(G,K) \cdot \lceil \log_2(\text{MinCut}(G,K)) \rceil$ rounds. Indeed, we can simulate the two-party protocol on $G$ across the cut $(A,B)$, where Alice is responsible for $A$ and Bob for $B$. In particular, if Alice needs to send a message to Bob (or vice-versa), it will be sent across edges crossing the cut. Note that in each round, at most $\text{MinCut}(G,K)\lceil \log_2(\text{MinCut}(G,K)) \rceil$ bits will be exchanged between Alice and Bob. We need $\lceil \log_2(\text{MinCut}(G,K)) \rceil$ bits in order to know the edge on which the message was sent. We can now invoke Corollary 2.4 to have

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \cdot \text{MinCut}(G,K) \cdot \lceil \log_2(\text{MinCut}(G,K)) \rceil \geq \Omega(y(\mathcal{H}) \cdot N).$$

Thus, we have a lower bound of

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot N}{\text{MinCut}(G,K)}\right).$$

□

*4.2.3 General $\mathcal{H}$.* We are now ready to prove our general lower bound for all simple graphs $\mathcal{H}$.

Theorem 4.8. *For arbitrary $G$, $K \subseteq V$, and graph $\mathcal{H}$, we have*

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{(y(\mathcal{H}) + n_2(\mathcal{H})) \cdot N}{\text{MinCut}(G,K)}\right).$$

Proof Sketch. We present a proof sketch here. For notational convenience, define $y = y(\mathcal{H})$ and $n_2 = n_2(\mathcal{H})$. Let $m = \max\left(\frac{y}{2}, \frac{n_2}{2\log(n_2)}\right)$. In general, as in the proof of Lemma 4.6, given $\mathcal{H}$ and a TRIBES instance $\text{TRIBES}_{m,N}(\hat{S},\hat{T})$ we construct a BCQ instance $q_{\mathcal{H},\hat{S},\hat{T}}$ such that $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\text{TRIBES}_{m,N}(\hat{S},\hat{T}) = 1$. To this end we need to "embed" the $m$ pairs of sets $(S_i,T_i)$ from $\text{TRIBES}_{m,N}(\hat{S},\hat{T})$ as relations in $q_{\mathcal{H},\hat{S},\hat{T}}$. For $m = \frac{y}{2}$, we embed the pairs $(S_i,T_i)$ in the forest $\text{F}(\mathcal{H})$ as done in Lemma 4.6. For $m = \frac{n_2}{2\cdot\log(n_2)}$, we consider $\text{C}(\mathcal{H})$. We then show that it must be the case that $\text{C}(\mathcal{H})$ either includes $\left(\frac{n_2}{2\log(n_2)}\right)$ vertex-disjoint cycles (referred to as *Case 1*), or that it has an independent set of size $\Omega(n_2)$ (referred to as *Case 2*). In both cases, we show how one can embed $\frac{n_2}{2\log(n_2)}$ pairs $(S_i,T_i)$ of $\text{TRIBES}_{m,N}(\hat{S},\hat{T})$ in $C(\mathcal{H})$. We defer the proof to the full paper [47].

Assuming the above embeddings, we conclude that $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\text{TRIBES}_{m,N}(\hat{S},\hat{T}) = 1$, where $m = \max\left(\frac{y}{2}, \frac{n_2}{2\log(n_2)}\right)$. Since sum and max are within a factor 2 of each other, we can write $m \geq \frac{y}{4} + \frac{n_2}{4\log(n_2)}$. We can now apply ideas from the proof of Lemma 4.7 to obtain the required lower bound $\widetilde{\Omega}\left(\frac{(y+n_2)\cdot N}{\text{MinCut}(G,K)}\right)$. □

Note that in Theorem 4.1, the upper bound follows from Lemma 4.5 and the lower bound from Theorem 4.8. We conclude this section by noting that when $N \geq |V|^2$, our upper and lower bounds differ by $\widetilde{O}(d)$ factor (for worst-case assignments of relations to players). In particular, Theorem 3.10 implies that the first two terms in the upper and lower bounds match up to an $\widetilde{O}(1)$ factor. In the full paper [47], we show that for worst-case assignment of relations, the second terms in the upper and lower bounds differ by a $\widetilde{O}(d)$ factor, as desired.

## 5 HYPERGRAPHS $\mathcal{H}$ AND GENERAL FAQ

Our results generalize fairly seamlessly to hypergraphs $\mathcal{H}$. For constant $d, r$, our upper and lower bounds match. However, for non-constant $d$, we have a gap of $\widetilde{O}(d^2 \cdot r^2)$, which is worse

than our gap of $\widetilde{O}(d)$ for the arity two case. Details are deferred to the full version [47].

We extend our results from BCQ to the general FAQ problem. We define the general FAQ problem here, which is a generalization of FAQ-SS. We are given a multi-hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ where for each hyperedge $e \in \overline{\mathcal{E}}$, we also have an input function $f_e : \prod_{v \in e} \mathrm{Dom}(v) \to \mathbb{D}$. In addition, we are given a set of *free variables* $\mathcal{F} \subseteq \overline{\mathcal{V}} : |\mathcal{F}| = \ell$ and[18] we would like to compute the function:

$$\phi\left(\mathbf{x}_{[\ell]}\right) = \bigoplus_{x_{\ell+1} \in \mathrm{Dom}(x_{\ell+1})}^{(\ell+1)} \cdots \bigoplus_{x_n \in \mathrm{Dom}(x_n)}^{(n)} \otimes f_S(\mathbf{x}_S), \quad (3)$$

where $\mathbf{x} = (x_u)_{u \in \overline{\mathcal{V}}}$ and $\mathbf{x}_S$ is $\mathbf{x}$ projected down to co-ordinates in $S \subseteq \overline{\mathcal{V}}$. The variables in $\overline{\mathcal{V}} \setminus \mathcal{F}$ are called *bound variables*. For every bound variable $i > \ell$, $\oplus^{(i)}$ is a binary (aggregate) operator on the domain $\mathbb{D}$. Different bound variables may have different aggregates. Finally, for each bound variable $i > \ell$ either $\oplus^{(i)} = \otimes$ (*product aggregate*) or $(\mathbb{D}, \oplus^{(i)}, \otimes)$ forms a commutative semiring (*semiring aggregate*) with the same additive identity $\mathbf{0}$ and multiplicative identity $\mathbf{1}$. As with FAQ-SS, we assume that the functions are input in the *listing* representation, i.e. the function $f_e$ is represented as a list of its non-zero values: $R_e = \{(\mathbf{y}, f_e(\mathbf{y})) | \mathbf{y} \in \prod_{v \in e} Dom(v) : f_e(\mathbf{y}) \neq \mathbf{0}\}$. Note that when $\oplus^{(i)} = \oplus$ is the same semiring aggregate for every $\ell < i \leq n$, we have the FAQ-SS problem.

For any $\mathbb{D}$, let $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ denote the class of FAQ problems, where each function in $\mathcal{H}$ has at most $N$ non-zero entries. (Note that we are not explicitly stating the operators for the bound variables $(\oplus^{(\ell+1)}, \ldots, \oplus^{(n)})$ since our upper and lower bounds hold for all such operators.) Let $\mathcal{R}(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K)$ denote the minimum worst-case number of rounds needed by a randomized protocol with error at most $\frac{1}{3}$ that computes any query in $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ on $G$ with functions assigned to nodes in $K$. For $O(1)$-degenerate hypergraphs $\mathcal{H}$ with $O(1)$-arity, we have

Theorem 5.1.

$$\mathcal{R}(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K) = \widetilde{\Theta}\left(\frac{(y(\mathcal{H}) + n_2(\mathcal{H})) \cdot N}{\mathrm{MinCut}(G, K)}\right)$$

*for any $\mathbb{D}$, specific choices of $\mathcal{F}$, arbitrary $G$ and $K$. When $G$ is a line,* $\mathrm{MinCut}(G, K) = 1$.

For general degenerate hypergraphs $\mathcal{H}$ with arity at most $r$, we have

Theorem 5.2.

$$\mathcal{R}\left(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K\right) \geq \widetilde{\Omega}\left(\frac{(d \cdot y(\mathcal{H}) + n_2(\mathcal{H})) \cdot N}{d \cdot r \cdot \mathrm{MinCut}(G, K)}\right).$$

*The lower bound differs from the upper bound by a factor of $O(r^2 d^2)$ in the worst case.*

We would like to mention here that our upper bound is a deterministic protocol and the lower bound is for randomized protocols. Details are in the full version [47].

[18] For a fixed $\mathcal{F}$, the vertices in $\overline{\mathcal{V}}$ can be renumbered so that $\mathcal{F} = [\ell]$ w.l.o.g.

# 6 MATRIX CHAIN MULTIPLICATION

We consider the following FAQ-SS problem. The network topology has $k+2$ players $P_0, \ldots, P_{k+1}$ such that $(P_i, P_{i+1})$ is an edge (i.e. $G$ is a line) where $P_0$ receives $\mathbf{x} \in \mathbb{F}_2^N$ and $P_i$ for $i \in [k]$ receives $\mathbf{A}_i \in \mathbb{F}_2^{N \times N}$. Player $P_{k+1}$ wants to compute $\mathbf{A}_k \cdot \mathbf{A}_{k-1} \cdots \mathbf{A}_1 \cdot \mathbf{x}$. Alternatively, for every $i \in [k]$, define $\mathbf{y}_i = \mathbf{A}_i \cdot \mathbf{y}_{i-1}$, with $\mathbf{y}_0 = \mathbf{x}$. Note that we want to compute $\mathbf{y}_k$. Note that this is an FAQ-SS problem since we can re-write the above as

$$\phi(z_k) = \sum_{(z_i)_{i=0}^{k-1} \in [N]^k} \left(\prod_{j=1}^{k} A_j(z_j, z_{j-1})\right) X(z_0), \quad (4)$$

where the functions satisfy $A_j(x, y) = \mathbf{A}_j[x, y]$ and $X(z) = \mathbf{x}[z]$ for every triple of indices $x, y, z \in [N]$.

We note that this problem can be solved in $O(kN)$ rounds.[19]

Proposition 6.1. *The* FAQ-SS *problem from* (4) *can be computed in $O(kN)$ rounds.*

We prove this proposition in the full paper [47]. We remark that when $k$ is large, a bottom-to-top fashion merge algorithm can achieve $O(N^2 \log k + k)$ rounds. (See the full version [47] for details.) In the next section, we prove a tight lower bound of $\Omega(kN)$ for the case $k \leq N$.

## 6.1 The Lower Bound

We will argue that the upper bound of $O(kN)$ rounds in Proposition 6.1 is tight if $k \leq N$. Before we do that we collect some definitions and results related to the min-entropy of a random variable.

*6.1.1 Background.* The *min-entropy* of a random variable $X$ is defined as $H_\infty(X) := -\log \max_{x \in \mathrm{supp}(X)} \Pr[X = x]$. For a random variable $X$ and an event $\mathcal{E}$ that is possibly correlated with $X$, define $H_\infty(X\mathcal{E}) = -\log \max_{x \in \mathrm{supp}(X)} \Pr[X = x, \mathcal{E}]$. Notice that in the above definition, we do not "normalize" $\Pr[X = x, \mathcal{E}]$ by a factor of $\Pr[\mathcal{E}]$.

For random variables $X$ and $Y$ , the conditional smooth min-entropy $H_\infty^\epsilon(X|Y)$ is defined as

$$\begin{aligned} H_\infty^\epsilon(X|Y) &= \sup_{\mathcal{E}} \min_{y \in \mathrm{supp}(Y)} H_\infty(X\mathcal{E}|Y = y) \\ &= \sup_{\mathcal{E}} \left(-\log \max_{(x,y) \in \mathrm{supp}(X,Y)} \Pr[\mathcal{E}, X = x | Y = y]\right) \end{aligned}$$

where the quantification over $\mathcal{E}$ is over all events $\mathcal{E}$ (which can be correlated with $X$ and $Y$) with $\Pr(\mathcal{E}) \geq 1 - \epsilon$. When $Y$ is a deterministic variable (in other words, we are not conditioning on any randomized variable), then we simply use $H_\infty^\epsilon(X)$:

$$H_\infty^\epsilon(X) = \sup_{\mathcal{E}} H_\infty(X\mathcal{E}), \quad (5)$$

where again the quantification over $\mathcal{E}$ is over all events $\mathcal{E}$ with $\Pr(\mathcal{E}) \geq 1 - \epsilon$.

The following lemma will be useful in our analysis:

Lemma 6.2 (Lemma 4 and Lemma 7 of [56]). *Let $Y$ be a random variable with support size at most $2^\ell$. Then we have for any $\epsilon \geq 0, \epsilon' > 0$ and random variable $X$, that $H_\infty^{\epsilon+\epsilon'}(X|Y) \geq H_\infty^\epsilon(X) - \ell - \log(1/\epsilon')$.*

[19] Note that the trivial algorithm takes $\Omega(kN^2)$ rounds.

Finally, we will use the following result where $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$:

Theorem 6.3. *Let the constant $\gamma > 0$ be small enough. Let $\mathbf{x} \in \mathbb{F}_2^N$, $\mathbf{A} \in \mathbb{F}_2^{N\times N}$ and $\mathbf{Y}$ be random variables such that for every $y \in \text{supp}(\mathbf{Y})$, $\mathbf{x}$ and $\mathbf{A}$ are independent conditioned on $\mathbf{Y} = y$. Moreover for some reals $\epsilon_1, \epsilon_2 \geq 0$, we have $H_\infty^{\epsilon_1}(\mathbf{A}|\mathbf{Y}) \geq (1-\gamma)N^2$, and $H_\infty^{\epsilon_2}(\mathbf{x}|\mathbf{Y}) \geq \alpha \cdot N$, where $\alpha \stackrel{def}{=} 3\gamma + \sqrt{2\gamma} + h(\sqrt{2\gamma})$. Then $H_\infty^{\epsilon_1+\epsilon_2+2^{-\Omega(\gamma N)}}(\mathbf{Ax}|\mathbf{Y}) \geq \left(1-\sqrt{2\gamma}\right)\cdot N$*

The proof of Theorem 6.3 follows from known results in pseudorandomness and appears in the full paper [47].

*6.1.2 Showing Proposition 6.1 is tight for $k \leq N$.* At a high level, we will prove by induction that for player $P_i$ at time about $\gamma iN$, the min-entropy of $\mathbf{y}_{i-1}$ is at least $\alpha \cdot N$ (and the situation at $P_{i+1}$ should be similar). Since by this time $P_{i+1}$ would have received at most $O(\gamma iN) \leq O(\gamma N^2)$ bits, this means $\mathbf{A}_i$ has min-entropy at least $(1-\gamma)N^2$. Thus, we can apply Theorem 6.3 to argue that at $P_{i+1}$ the min-entropy of $\mathbf{y}_i = \mathbf{A}_i \cdot \mathbf{y}_{i-1}$ is large. To finish the inductive argument we have to wait for $\gamma N$ more steps but by Lemma 6.2, even then $\mathbf{y}_i$ will still have high enough min-entropy. It is natural to wonder if we can make the same argument using Shannon entropy instead of min-entropy. In the longer version, we show that this is not possible.

We define some useful notations before we prove the lower bound. At any given time $t$, let $\mathbf{m}^i(t)$ denote the transcript of messages exchanged on the link between $P_{i-1}$ and $P_i$ till time $t$. For $i \in [k+1]$, define $t_i = \frac{\gamma}{4} \cdot iN$, and $\widetilde{\mathbf{m}}^i = \mathbf{m}^i(t_i)$. For a random variable $\mathbf{m}$, we will use $m$ to denote a specific value of the random variable $\mathbf{m}$. In addition, we use $\widetilde{\mathbf{m}}^{[i]}$ and $\widetilde{m}^{[i]}$ to denote the tuples $(\widetilde{\mathbf{m}}^1, \widetilde{\mathbf{m}}^2, \cdots, \widetilde{\mathbf{m}}^i)$ and $(\widetilde{m}^1, \widetilde{m}^2, \cdots, \widetilde{m}^i)$ respectively.

Let $\epsilon^* = 2^{-\Omega(\gamma N)}$ be at least thrice the maximum of $2^{-\gamma N/4}$ and the $2^{-\Omega(\gamma N)}$ term in Theorem 6.3. We will argue:

Lemma 6.4. *Let $\mathbf{A}_i$ for every $i \in [k]$ and $\mathbf{x}$ be all uniformly and independently distributed. Let $\gamma > 0$ be such that*[20]

$$4\gamma + \sqrt{2\gamma} + h(\sqrt{2\gamma}) \leq 1, \tag{6}$$

*and $\gamma N/4$ is an integer. Then we have the following for every $i \in [k+1]$:*

$$H_\infty^{i\epsilon^*}\left(\mathbf{y}_{i-1}|\widetilde{\mathbf{m}}^{[i]}\right) \geq N(1-\gamma-\sqrt{2\gamma}). \tag{7}$$

The proof appears in the full paper [47]. The above immediately gives us our lower bound (details are in the full paper [47]):

Theorem 6.5. *Any protocol that solves the* FAQ-SS *problem from* (4) *with $k \leq N$ and large enough $N$, with success probability at least $1/2$, takes $\Omega(\gamma kN)$ rounds.*

## 7 RELATED WORK

We now survey the most closely related work. Due to lack of space, a detailed discussion is deferred to the full paper [47].

[20] There exists a value $\gamma \geq 0.01$ (for large enough $N$) that satisfies the required conditions.

*Parallel Database Query Computation.* The MPC model has seen a lot of research activity in the last few years [2, 9, 10, 37, 45, 46]. We compare these models with ours in Section 1.1.

*Widths of GHDs.* The Internal Node Width $y(\mathcal{H})$ of a GHD focuses on minimizing the number of internal (non-leaf) nodes in GHDs of acyclic hypergraphs. There is a related notion for Tree Decompositions called *Lean* Tree Decompositions (LTDs) [11, 24, 60]. For GHDs, the problem of computing GHDs that minimize certain cost functions of the HDs are studied in the framework of Weighted GHDs [33, 57]. We refer the reader to [30] for a recent survey on widths for GHD.

*Distributed Computing and Communication Complexity.* As stated earlier, our model is similar to (and different from) the CONGEST model in distributed computing [54]. Recently, there has been work on the same model as ours but instead of minimizing the number of rounds, they focus on minimizing the total communication of the protocols [14, 19, 20, 55, 61, 63]. Finally, [18] obtained results on minimizing the number of rounds of protocols in our setup for some well-studied functions in two-party communication complexity literature.

## 8 FUTURE WORK

We leave the following questions as future work: handling node failures, finding optimal assignments of functions to players in $G$, identifying the optimal topology for a given query and finally, closing the gap between our upper and lower bounds for $d$-degenerate graphs for super-constant $d$. We address the assumption that the functions are completely assigned to players in $G$ in the longer version. We cannot (yet) handle node failures and the condition that $N$ has to be larger than the size of $G$. We address assumptions on the knowledge of $q$ and $G$ in the full paper [47].

## ACKNOWLEDGMENTS

We thank the anonymous reviewers of PODS'19 for their helpful comments. We are greatly indebted to Arkadev Chattopadhyay and David Zuckerman for their insights that led to the results in Section 6. We thank Dan Suciu, Hung Ngo, Martin Grohe and Oliver Kennedy for helpful discussions. This work was supported by NSF grant CCF-1717134.

## REFERENCES

[1] M. Abu-Elkheir, M. Hayajneh, and N. A. Ali. Data management for the internet of things: Design primitives and solution. *Sensors*, 13(11):15582–15612, 2013.
[2] F. N. Afrati, M. R. Joglekar, C. Ré, S. Salihoglu, and J. D. Ullman. GYM: A multiround distributed join algorithm. In *ICDT*, pages 4:1–4:18, 2017.
[3] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, Mar 2000.
[4] D. Akatov. *Exploiting parallelism in decomposition methods for constraint satisfaction.* PhD thesis, University of Oxford, UK, 2010.
[5] N. Alon, S. Hoory, and N. Linial. The moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, Mar 2002.
[6] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley, 1992.
[7] N. Bakibayev, T. Kociský, D. Olteanu, and J. Zavodny. Aggregation and ordering in factorised databases. *PVLDB*, 6(14):1990–2001, 2013.
[8] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.
[9] P. Beame, P. Koutris, and D. Suciu. Communication steps for parallel query processing. In *PODS*, pages 273–284, 2013.

[10] P. Beame, P. Koutris, and D. Suciu. Skew in parallel query processing. In *PODS*, pages 212–223, 2014.

[11] P. Bellenbaum and R. Diestel. Two short proofs concerning tree-decompositions. *Combinatorics, Probability & Computing*, 11(6):541–547, 2002.

[12] H. L. Bodlaender. Nc-algorithms for graphs with small treewidth. In *Graph-Theoretic Concepts in Computer Science, 14th International Workshop, WG '88, Amsterdam, The Netherlands, June 15-17, 1988, Proceedings*, pages 1–10, 1988.

[13] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. In *Mobile Data Management, Second International Conference, MDM 2001, Hong Kong, China, January 8-10, 2001, Proceedings*, pages 3–14, 2001.

[14] M. Braverman, F. Ellen, R. Oshman, T. Pitassi, and V. Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *FOCS*, pages 668–677, 2013.

[15] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

[16] A. Chattopadhyay. Personal communication, 2018.

[17] A. Chattopadhyay, M. Koucký, B. Loff, and S. Mukhopadhyay. Simulation beats richness: New data-structure lower bounds. In *STOC*, 2018.

[18] A. Chattopadhyay, M. Langberg, S. Li, and A. Rudra. Tight network topology dependent bounds on rounds of communication. In *SODA*, pages 2524–2539, 2017.

[19] A. Chattopadhyay, J. Radhakrishnan, and A. Rudra. Topology matters in communication. In *FOCS*, pages 631–640, 2014.

[20] A. Chattopadhyay and A. Rudra. The range of topological effects on communication. In *ICALP*, pages 540–551, 2015.

[21] S. Cohen, Y. Kanza, and Y. Sagiv. Generating relations from XML documents. In *ICDT*, pages 282–296, 2003.

[22] R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1-2):41–85, 1999.

[23] Y. Dodis and R. Oliveira. On extracting private randomness over a public channel. In *RANDOM*, pages 252–263, 2003.

[24] J. Erde. A unified treatment of linked and lean tree-decompositions. *J. Comb. Theory, Ser. B*, 130:114–143, 2018.

[25] J. Gehrke and S. Madden. Query processing in sensor networks. *IEEE Pervasive Computing*, 3(1):46–55, 2004.

[26] M. Ghaffari. *Improved Distributed Algorithms for Fundamental Graph Problems*. PhD thesis, EECS department of MIT, 2016.

[27] M. Ghobadi, R. Mahajan, A. Phanishayee, N. R. Devanur, J. Kulkarni, G. Ranade, P. Blanche, H. Rastegarfar, M. Glick, and D. C. Kilper. ProjecToR: Agile reconfigurable data center interconnect. In *SIGCOMM*, pages 216–229, 2016.

[28] M. Göös, S. Lovett, R. Meka, T. Watson, and D. Zuckerman. Rectangles are nonnegative juntas. In *STOC*, pages 257–266, 2015.

[29] M. Göös, T. Pitassi, and T. Watson. Query-to-communication lifting for BPP. In *FOCS*, pages 132–143, 2017.

[30] G. Gottlob, G. Greco, N. Leone, and F. Scarcello. Hypertree decompositions: Questions and answers. In *PODS*, pages 57–74, 2016.

[31] M. H. Graham. On the universal relation. In *Tech Report*, 1979.

[32] A. Grama, V. Kumar, A. Gupta, and G. Karypis. *Introduction to Parallel Computing*. Pearson Education. Addison-Wesley, 2003.

[33] G. Greco, N. Leone, and F. Scarcello. On weighted hypertree decompositions. In *Proceedings of the Twelfth Italian Symposium on Advanced Database Systems, SEBD 2004, S. Margherita di Pula, Cagliari, Italy, June 21-23, 2004*, pages 54–61, 2004.

[34] M. M. Halldórsson and E. Losievskaja. Independent sets in bounded-degree hypergraphs. *Discrete Applied Mathematics*, 157(8):1773–1786, 2009.

[35] M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 21–30, 2015.

[36] T. S. Jayram, R. Kumar, and D. Sivakumar. Two applications of information complexity. In *STOC*, pages 673–682, 2003.

[37] M. Joglekar and C. Ré. It's all a matter of degree: Using degree information to optimize multiway joins. In *ICDT*, pages 11:1–11:17, 2016.

[38] M. R. Joglekar, R. Puttagunta, and C. Ré. AJAR: aggregations and joins over annotated relations. In *PODS*, pages 91–106, 2016.

[39] M. A. Khamis, H. Q. Ngo, and A. Rudra. FAQ: questions asked frequently. In *PODS*, pages 13–28, 2016.

[40] M. A. Khamis, H. Q. Ngo, and A. Rudra. Juggling functions inside a database. *SIGMOD Record*, 46(1):6–13, 2017.

[41] J. Kohlas and N. Wilson. Semiring induced valuation algebras: Exact and approximate local computation algorithms. *Artif. Intell.*, 172(11):1360–1399, 2008.

[42] D. Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, 2000.

[43] A. V. Kostochka. On almost (k-1)-degenerate (k+1)-chromatic graphs and hypergraphs. *Discrete Mathematics*, 313(4):366–374, 2013.

[44] P. Koutris. Lecture notes on acyclic joins, lecture 4. 2016.

[45] P. Koutris, P. Beame, and D. Suciu. Worst-case optimal algorithms for parallel query processing. In *ICDT*, pages 8:1–8:18, 2016.

[46] P. Koutris and D. Suciu. A guide to formal analysis of join processing in massively parallel systems. *SIGMOD Record*, 45(4):18–27, 2016.

[47] M. Langberg, S. Li, S. V. M. Jayaraman, and A. Rudra. Topology dependent bounds for computing faqs. In *full paper*, 2019.

[48] L. C. Lau. An approximate max-steiner-tree-packing min-steiner-cut theorem*. *Combinatorica*, 27(1):71–90, 2007.

[49] F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[50] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.

[51] D. Olteanu and J. Závodný. Size bounds for factorised representations of query results. *ACM Trans. Database Syst.*, 40(1):2:1–2:44, 2015.

[52] E. J. O'Neil, P. E. O'Neil, and K. Wu. Bitmap index design choices and their performance implications. In *Eleventh International Database Engineering and Applications Symposium (IDEAS 2007), September 6-8, 2007, Banff, Alberta, Canada*, pages 72–84, 2007.

[53] N. Parzanchevski and A. Ta-Shma. Personal communication, 2018.

[54] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*.

[55] J. M. Phillips, E. Verbin, and Q. Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *SODA*, pages 486–501, 2012.

[56] R. Renner and S. Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *ASIACRYPT*, pages 199–216, Berlin, Heidelberg, 2005. Springer-Verlag.

[57] F. Scarcello, G. Greco, and N. Leone. Weighted hypertree decompositions and optimal query plans. In *PODS*, pages 210–221, 2004.

[58] A. Ta-Shma. Almost optimal dispersers. *Combinatorica*, 22(1):123–145, 2002.

[59] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13(3):566–579, 1984.

[60] R. Thomas. A menger-like property of tree-width: The finite case. *J. Comb. Theory, Ser. B*, 48(1):67–76, 1990.

[61] P. Tiwari. Lower bounds on communication complexity in distributed computer networks. *J. ACM*, 34(4):921–938, 1987.

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

[63] D. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *STOC*, pages 941–960, 2012.

[64] D. P. Woodruff and Q. Zhang. Distributed statistical estimation of matrix products with applications. In *PODS*, pages 383–394, 2018.

[65] A. C. Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213, 1979.

[66] C. T. Yu and M. Z. Ozsoyoglu. An algorithm for tree-query membership of a distributed query. In *The IEEE Computer Society's Third International Computer Software and Applications Conference, COMPSAC 1979, 6-8 November, 1979, Chicago, Illinois, USA*, pages 306–312, 1979.

[67] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996.

# A COMPARISON WITH RELEVANT MODELS

## A.1 Basic MPC model

We formally define the MPC model used in [9] and the model adopted by [2] here, both in the language of our model (Model 2.1). We consider the MPC with no replication, which is known as the basic MPC model in the literature.

Model A.1 (MPC(0)). *We are given a query $q$ and its underlying hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ with input functions $f_e$ having at most $N$ non-zero values for every $e \in \overline{\mathcal{E}}$. We consider the network topology $G'$ with $p + k$ nodes, defined as follows. There are $k$ nodes, each assigned a function $f_e$ for every $e \in \overline{\mathcal{E}}$. We call this set $K$. There are no edges between any pair of nodes in $K$. All nodes in $K$ are directly connected by an edge to every node in a clique with $p$ nodes that are disjoint from $K$ (also a part of $G'$). Each vertex in $K$ has capacity $N$ and all the remaining nodes have capacity $L$. The capacity of a vertex bounds the number of bits it can receive in each round. Given this setup, we would like to compute* BCQ *(and more generally an* FAQ*) of $\mathcal{H}$ on $G'$.*
*To design a protocol for this computation, we can assume that every node in $G'$ has the knowledge of $q$ and $G'$. At the end of the protocol, a pre-determined player in $K$ knows the answer to $q$.*
*Finally, given the above setup, our goal is to design protocols that minimize the number of rounds to compute $q$ on $G$. This model does not take into account the internal computation done by the $p + k$ nodes and assumes the nodes co-operatively compute the answer to $q$.*

We now summarize the differences of this model from ours.

### A.1.1 *Differences from our Model.*

- MPC assumes a specific choice of a network topology $G'$ as opposed to general topology $G$ in our model.
- MPC assumes a specific assignment of functions in $\overline{\mathcal{E}}$ to players in $G'$. Our upper bound techniques can handle any assignment but our lower bounds are for a specific class of assignments. We would like to mention here that this is true for the models in [2, 9] as well and we consider one such assignment in Model A.1.
- MPC assumes node capacities whereas ours assumes edge capacities.
- The models in [2, 9] design protocols wherein the number of rounds is either constant or a function of $k$. The number of rounds in our model are a function of $N$.
- The models in [2, 9] generally prove results for computing natural join whereas we look at BCQ (and more generally FAQs). We note that the results of [2, 9] for natural join apply for BCQ as well. This is true for both upper and lower bounds in [9].

We consider two instantations of this model – one by [9] and the other by [2].

*A.1.2 Fixing $p$ and Determining $L$ [9].* This model assumes $N$ is larger than the size of $G'$ and all the functions $f_e$ are matchings (i.e., skew-free). In other words, for each variable $v \in e$, each of the values $x_v \in \mathrm{Dom}(v)$ can occur in at most one tuple in $f_e$. Using Proposition 3.2 and Theorem 3.3 in [9], it can be shown that there exists an optimal one round protocol to solve BCQ of any star $\mathcal{H}$ on $G'$ with $L = \Omega\left(\frac{k \cdot N}{p}\right)$. Further, when $\mathcal{H}$ is a forest, BCQ of $\mathcal{H}$ on $G'$ takes $\Theta(\log(D'))$ rounds for the same $L$ (where $D'$ is the diameter of $\mathcal{H}$). We would like to mention here that a follow up work [10] handled input functions with specific types of skew and proved upper and lower bounds for the queries considered above. Since each node in the $(p)$-clique can have different capacities in this scenario, we do not discuss it further here.

*A.1.3 Fixing $L$ and Determining $p$ [2].* This model assumes the size of $G'$ is much larger than $N$. Assuming $L = (k \cdot N)^{\frac{1}{\delta}}$ for a fixed small constant $\delta > 1$, we can use the Main Results 1 and 2 from [2] and show that there exists a protocol to solve BCQ of any star $\mathcal{H}$ in: (1) $O(k)$ rounds with $p = (k \cdot N)^{2-\frac{2}{\delta}}$ and (2) $O(\log_2(k))$ rounds with $p = (k \cdot N)^{6-\frac{2}{\delta}}$.

Before we instantiate our model for a comparison with the above models, we would like to state that while our model can handle the constraint where the size of $G'$ can be larger than $N$, our techniques cannot. Hence, we restrict our comparison to the model in Section A.1.2. We now instantiate our model (Model 2.1) with $G'$ and assume that each edge in $G'$ has capacity

$$L' = \frac{L}{k} = \frac{N}{p}. \tag{8}$$

Note that this is a weaker version of Model A.1 since node capacities don't necessarily translate to equal edge capacities when the goal is to compute $q$ on $G'$. We take this route as it helps us make a fair comparison with Section A.1.2.

*A.1.4 Our Results in Model A.1.* We show how our upper bound techniques apply for solving BCQ of any star $\mathcal{H}$ on $G'$. We can instantiate Corollary 4.2 with capacity $\widetilde{\Theta}(1)$ to get $O\left(\min_{\Delta \in [|V(G')|]}\left(\frac{N}{\mathrm{ST}(G', K, \Delta)} + \Delta\right)\right)$ rounds. We claim that $\min_{\Delta \in [|V(G')|]}\left(\frac{N}{\mathrm{ST}(G', K, \Delta)} + \Delta\right) = O\left(\frac{N}{p}\right)$. To see this, we show such a Steiner tree packing containing $p$ trees with diameter 2 – each node in the $p$-clique in $G'$ along with all its $k$ edges incident on $K$ forms a Steiner tree. Since there are $p$ such nodes, we can obtain such a packing. Recall that when each edge in $G'$ has capacity $L'$ (instead of the $O(\log_2(D))$ capacity in Model 2.1), our upper bound gets divided by $L'$. Thus, we have an upper bound of $O\left(\frac{N}{L' \cdot p}\right) = O(1)$ (using (8)) i.e., a constant number of rounds.

Note that a lower bound of one round on the number of rounds is trivial. Hence, we can obtain a tight bound of $\widetilde{\Theta}(1)$ for any star $\mathcal{H}$, resulting in an one round protocol matching results in Section A.1.2.

Given the tight results for the star case, there is a natural generalization for our protocol and bounds when $\mathcal{H}$ is a forest using ideas from the proof of Lemma 4.4. We start by noting that all stars at the same level in $\mathcal{H}$ can be computed simultaneously since each node in $K$ is directly connected to each node in the $(p)$-clique. In particular, we can run the star protocol used above on all these stars simultaneously but we still need to be able to uniquely identify the stars computed. It's not too hard to see that this can be done with $O(\log(y(\mathcal{H})))$ additional information for each internal node $v$. This results in an upper bound of $O\left(D' \cdot \log(y(\mathcal{H})) \cdot \min_{\Delta \in [|V|]} \left(\frac{N}{\mathrm{ST}(G',K,\Delta)} + \Delta\right)\right) = O\left(\frac{D' \cdot \log(y(\mathcal{H})) \cdot N}{p}\right)$, where $\mathrm{ST}(G',K,2) = p$ and $D'$ is the diameter of $\mathcal{H}$. If we divide our upper bound by $L'$ and substitute its value from (8), we can use ideas similar to those used in the star case to obtain a protocol with $O(D')$ rounds. However, our lower bound techniques do not work for the assignment of functions to $K$ in Model A.1. We would like to mention that the model in Section A.1.2 takes $\Theta(\log(D'))$ rounds for this case (though the upper bound only holds for the special case of matching databases).

Finally, for general simple graphs $\mathcal{H}$, we decompose $\mathcal{H}$ into a core and a forest using Construction 2.9. We use the *trivial protocol* on the core, which is basically sending all functions to one player in $K$ and is independent of the induced query in the core. We would like to mention that this is worse than existing protocols [2, 9] for $\mathcal{H}$ with non-constant degeneracy $d$ since we do not exploit any information about the query.

Before we move to the next model, we would like to mention here that the results of Section A.1.2 and our results match up to a constant factor for the case when $\mathcal{H}$ is a star. The upper bounds match since the protocols in both cases split the input functions the same way – the model in Section A.1.2 uses hashes to achieve this and we use Steiner tree packings for the same. The results however start diverging even when $\mathcal{H}$ is a tree of small depth.

## A.2 General MPC model

We now perform our second and final comparison. We formally define the model from [45], which is a followup of [9, 10] and performs a *worst-case analysis* of the communication cost for join queries. All the three models are described in [46]. We define it in the language of our model like we did for Model A.1.

Model A.2 (MPC($\epsilon$)). *Let $\epsilon$ be a fixed value s.t. $0 \leq \epsilon < 1$. We are given a query $q$ and its underlying hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ with input functions $f_e$ having at most $N$ non-zero values for every $e \in \overline{\mathcal{E}}$. We consider the network topology $G''$, which is a clique on $p$. The input of size $k \cdot N$ is uniformly partitioned across the $p$ nodes. Let $K = V(G'')$. It follows that $|K| = p$. All nodes in $G$ have capacity $L(\epsilon)$. The capacity of a vertex bounds the number of bits it can receive in each round. Given this setup, we would like to compute* BCQ *(and more generally an* FAQ*) of $\mathcal{H}$ on $G''$.*

*To design a protocol for this computation, we can assume that every node in $G''$ has the knowledge of $q$ and $G''$. At the end of the protocol, a pre-determined player in $K$ knows the answer to $q$.*

*Finally, given the above setup, our goal is to design protocols that minimize the number of rounds to compute $q$ on $G''$. This model does not take into account the internal computation done by the $p$ nodes and assumes the nodes co-operatively compute the answer to $q$.*

We now summarize the differences of this model from ours.

### A.2.1 *Differences from our Model.*

- MPC($\epsilon$) assumes a specific choice of a network topology $G''$ as opposed to general topology in our model.
- MPC($\epsilon$) assumes a uniform distribution of the input across the $p$ nodes instead of one function being completely assigned to a specific node in $G''$ in our model.
- MPC($\epsilon$) works with node capacities like Model A.1, whereas ours works on edge capacities.
- The model in [45] designs protocols wherein the number of rounds either constant or a function of $k$. The number of rounds in our model are a function of $N$.
- The model in [45] proves results for computing natural join whereas we look at BCQ (and more generally FAQs). Note that their upper results for natural join apply for BCQ as well (but lower bound results do not transfer).

We consider the instantiation of this model by [45]. We would like to mention here that the models studied in [2, 9, 10] can all be instantiated in this setting only for proving upper bounds.

### A.2.2 *Fixing $p$ and Determining $L$ [45].*

This model assumes $N$ is larger than the size of $G'$ and there are no restrictions in the input functions. Using Theorems 3.1 and 3.3 of [45], it can be shown that there exists an optimal one round protocol to solve BCQ of any star $\mathcal{H}$ on $G''$ with $L\left(\epsilon = 1 - \frac{1}{k}\right) = \Omega\left(\frac{N}{p^{1-\epsilon}}\right) = \Omega\left(\frac{N}{p^{\frac{1}{k}}}\right)$. Further, when $\mathcal{H}$ is a forest, BCQ of $\mathcal{H}$ on $G''$ takes $O(k)$ rounds[21] with $L\left(\epsilon = 1 - \frac{1}{\rho^*(\mathcal{H})}\right) = \Omega\left(\frac{N}{p^{1-\epsilon}}\right) = \Omega\left(\frac{N}{p^{\frac{1}{\rho^*(\mathcal{H})}}}\right)$ using ideas in Section 4 of [45]. Here, $\rho^*(\mathcal{H})$ denotes the edge cover number of $\mathcal{H}$ (i.e., size of the minimum edge cover of $\mathcal{H}$).

[21] For the case when we are interested in computing the join query of $\mathcal{H}$, then there is also a matching $\Omega(k)$ lower bound.

We now instantiate our model (Model 2.1) with $G''$ and assume that each edge in $G''$ has capacity

$$L'' = \frac{L(\epsilon)}{p}. \tag{9}$$

Further, we assume that the input functions are not distributed uniformly but rather based on some pre-determined hash functions. Note that this certainly makes our model (Model A.1) more restrictive since node capacities don't necessarily translate to equal edge capacities when the goal is to compute $q$ on $G''$ and the hash-based split (see Appendix H.6.1) restricts the way in which input functions can be distributed across nodes in $G''$. We opt for this since it helps us make a fair comparison with Section A.2.2.

*A.2.3 Our Results in Model A.2.* We now show how our upper bound techniques apply in this model for solving BCQ of any star $\mathcal{H}$ on $G'$. We do not compare lower bounds here since (1) [45] lower bounds do not hold for BCQ (or at least it does not follow immediately from their lower bounds for the join quries) and (2) Our lower bounds for the case when the functions are uniformly distributed over the players are quantitatively very weak. For the upper bound, we can instantiate Corollary H.6 with capacity $\widetilde{\Theta}(1)$ to get $O\left(\min_{\Delta\in[p]}\left(\frac{N}{\mathrm{ST}(G'',K,\Delta)}+p\cdot\Delta\right)\right)$ rounds. We claim that $\min_{\Delta\in[p]}\left(\frac{N}{\mathrm{ST}(G'',K,\Delta)}+p\cdot\Delta\right)=O\left(\frac{N}{p}+p\right)$. To see this, we show a Steiner tree packing containing $\frac{p-1}{2}$ trees with diameter 1 – we can greedily keep picking and throwing out paths of length $p-1$ from $G''$ that contain all the $p$ vertices. Each such path forms a Steiner tree. Since we can identify $\frac{p-1}{2}$ such paths, we can obtain such a packing. Recall that when each edge in $G''$ has capacity $L(\epsilon)$ instead of the standard $O(\log_2(N))$, our upper bound gets divided by $L''$. Thus, we have an upper bound of $O\left(\frac{\frac{N}{p}+p}{L''}\right)$. Using (9) and the fact $N\geq p^2$ (from Model 2.1), we get a $O\left(p^{\frac{1}{k}}\right)$ round protocol. Note that this is worse than the one round protocol by Section A.2.2.

For the case when $\mathcal{H}$ is a forest, we can instantiate Corollary H.7 with capacity $\widetilde{\Theta}(1)$ to get a bound of

$$O\left(y(\mathcal{H})\cdot\min_{\Delta\in[|V|]}\left(\frac{N\cdot r}{\mathrm{ST}(G'',K,\Delta)}+p\cdot\Delta\right)\right)=O\left(\frac{D'\cdot\log(y(\mathcal{H}))\cdot N}{p}\right),$$

where $\mathrm{ST}(G'',K,1)=p$ and $D'$ is the diameter of $\mathcal{H}$. We can use ideas from Section A.1.4 and from those used in the star case to obtain a protocol with $O\left(D'\cdot p^{\frac{1}{\rho^*(\mathcal{H})}}\right)$ rounds. In particular, to get this bound, we divide our upper bound by $L''$ and substitute its value form (9). Note that this is worse than the $O(k)$ round protocol by Section A.2.2.

Finally, for general simple graphs $\mathcal{H}$, we decompose $\mathcal{H}$ into a core and a forest using Construction 2.9. We use the *trivial protocol* on the core, which is basically sending all functions to one player in $K$ and is independent of the induced query in the core. As stated in Section A.1.4, this is worse than existing protocols [45] for $\mathcal{H}$ with non-constant degeneracy $d$ since we do not exploit any information about the query.

## A.3 Scope for Future Work

Many open questions arise out of this comparison. We summarize them here and leave them for future work.

- Can we modify our model to handle node failures like Models A.1 and A.2 do, using replication?
- Can we improve over our *trivial protocol* for cyclic queries using ideas from [2, 9, 10, 37, 45, 46]?
- Can our algorithmic ideas for set intersection be plugged into the Models A.1 and A.2?
- Can we extend our techniques to handle arbitrary distributions of input functions to nodes in the topology?

## A.4 Connection to Sensor Networks

Sensor networks are typically tree-like topologies, where the goal is to efficiently and accurately report aggregate queries on data generated by the sensors. Since the sensors can traditionally store only little data, they stream their data (as they generate them) to designated points in the topology called storage points. There is a server that has more computational power and initiates these queries, collects the query answers, reports them and so on. Join/Aggregate queries are computed either between the storage points or between the server and a storage point [50].

We now restate this setting in our language. The server and the storage points are the nodes in $G$ and the edges are defined based on the sensor network. The query to be computed on $G$ is a FAQ $q$ (Joins/Aggregates are a special cases of FAQ), whose underlying query hypergraph is $\mathcal{H}$. The input functions in $\mathcal{H}$ are assigned to a subset of nodes $K$ in $G$. The upper and lower bounds that we obtain for computing $q$ on $G$ assuming all input functions have size at most $N$ apply for this setting in sensor networks. Further, in our setup, we can make any pre-determined node in $K$ (say the server) know the answer to $q$. In particular, this implies our model captures query computation in Sensor Networks for a specific class of queries.

Due to the theoretical nature of our results, the potential applications of our model/results in the IoT setting are somewhat speculative. We hope that our work motivates more study of our general model in these applications areas.

## A.5 Which Distributed Computing Model to Use?

We believe that different models could be used for different settings. For instance, if all the nodes are interconnected to each other in a compute farm (i.e., $G$ is a clique) and each node can receive only a certain amount of data in a communication round and we

are interested in computing the join query corresponding to $\mathcal{H}$, then the MPC-based models A.1 and A.2 are more suitable. On the other hand, if we are looking at more general topologies $G$, the capacities are on the edges and we are interested in computing BCQ of $\mathcal{H}$, then using our model might make more sense.

# B THE CLIQUE OPEN PROBLEM

Consider the case where $\mathcal{H}$ is a $k$-clique with all input functions having size at most $N$ and $G$ is an edge $e = (a, b)$. The goal is to compute BCQ of $\mathcal{H}$ on $G$ assuming worst-case assignment of functions in $\mathcal{H}$ to players in $G$.

We can prove an upper bound of $O(k^2 \cdot N)$ as follows. Consider an assignment where half of the functions (i.e., $\frac{k\cdot(k-1)}{4}$ of them) are assigned to $a$ and the other half of them is assigned to $b$. In particular, $a$ can send all its functions to $b$ to compute the BCQ of $\mathcal{H}$ on $G$, the upper bound of $O(k^2 \cdot N)$ follows. Since we consider worst-case assignments, we can't prove a better upper bound. The best lower bound known so far for this query is $\Omega(k \cdot N)$, which is worse than the upper bound by a factor of $O(k)$. Going beyond this bound seems beyond the reach of current two-party communication complexity techniques [16]. We believe that our work will provide more motivation to solve this outstanding open question in two-party computational complexity.

# C MORE RELATED WORK

We present a detailed discussion of related work here.

*Algorithms for* FAQ. The authors of [39] defined and presented algorithms to solve the FAQ problem that encompasses many frequently asked questions in databases, PGMs, matrix operations and logic. A quick followup work re-stated the algorithms of [39] in the GHD framework [38]. We would like to note here that special instances of FAQ problems (i.e. FAQ-SS from [39]) have been studied before in [3, 7, 22, 41, 51].

*Weighted GHDs.* The problem of computing GHDs that minimize certain cost functions of the HDs are studied in the framework of Weighted GHDS [33, 57]. For a given hypergraph $\mathcal{H}$, one way to map our notion of width to their setting is to consider a vertex aggregating function on every candidate HD $\mathcal{T}$ for $\mathcal{H}$. In particular, we can write

$$\Lambda_H^{f'}(\mathcal{T}) = \sum_{v' \in V(\mathcal{T})} f'_{\mathcal{H}}(v'), \tag{10}$$

where $f'_{\mathcal{H}} = 1$ if $v'$ is a internal node and 0 otherwise. It's not too hard to see that $f'_{\mathcal{H}}$ can be computed in linear time in size of $\mathcal{T}$. Given this setup, Theorem 3.4 in [33] proves that computing Minimal GHDs over HDs for arbitrary vertex aggregation functions is NP-Hard.

However, this does not hold in our case since there is always a GHD with one internal node (containing all the variables in $\mathcal{H}$). As a result, considering the minimization over all GHDs for our case is trivial and doesn't give use tight results. In particular, our setup is a bit different and we minimize over GYO-GHDs (Construction 2.9). For the tightness of our bounds for $\mathcal{H}$ with constant degeneracy and constant arity, we only need an $O(1)$-factor approximation of Internal-Node-Width, which we achieve.

*Lean Tree Decompositions.* We discuss LTDs in detail here. In particular, the LTDs minimize the internal nodes in the following way – they try to retain only pairs of connected internal nodes whose intersection forms a bridge in the original graph $\mathcal{H}$. The other nodes are forced to become leaves of one of the internal nodes. While our construction procedure of MD-GHDs tries to convert existing internal nodes to become leaves of some internal nodes, we do not (yet) see an exact one-to-one mapping of MD-GHDs to LTDs. We would like to mention here that both the goals of MD-GHDs and LTDs are the same i.e., to minimize the number of internal nodes. We would like to note here that $y(\mathcal{H})$ can potentially reduce the depth of the GHD as well. Reducing depth of GHDs (sometimes by increasing the treewidth) has been considered before [2, 4, 12].

*Two-party communication complexity.* Both the strands of work on round complexity and total communication from [18, 19] coincide for the special case when $G$ is just an edge. Note that in this case we have two players and the model coincides with the very well studied model of two-party communication complexity introduced by Yao [65], which has proved to be an extremely worthwhile model to study with applications in diverse areas of computer science.

*Entropy in communication complexity.* Information complexity by now a well established sub-field of communication complexity that uses Shannon's entropy to measure the amount of information exchanges in a two party communication protocol and was essentially introduced in the work of Chakrabarti et al. [15] and was used in a systematic way to tackle multiple problems in [8]. To the best of our knowledge, min-entropy has only been used very recently in communication complexity [28, 29] though it has found numerous applications in pseudorandomness and cryptography for at least two decades [62]. Our work add to the recently growing body of work that uses min-entropy to prove communication complexity results [17].

*Matrix products in communication complexity.* Parallel algorithms for computing matrix/vector products have been studied extensively [32]. In particular, the communication complexity in these models have been studied: however, these models are different from ours but are not very relevant. Closer to our work is the work of Woodruff and Zhang [64] in the two-party communication complexity model where the two parties are given two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ and they are interested in computing

some statistical property of $\mathbf{A}_1 \cdot \mathbf{A}_2$ (e.g. computing some norms on the product). Also they results are for matrices over integer/reals. By contrast our work is on computing matrix-vector product (over $\mathbb{F}_2$), where the matrix itself is a matrix product and there are multiple players on the line topology.

*Distributed Computing and Communication Complexity.* As was mentioned earlier, our model is similar to (and is different from) the CONGEST model in distributed computing [54] and is still an active area of research. Recently there has been work that deals with the graph communication model as ours but instead of minimizing the round complexity, these results are for the case of minimizing the *total communication* of the protocols. (We note that the total communication corresponds to the *message complexity* of distributed protocols.) Most of the work in this area has been for specific classes of $G$. For example, the early work of Tiwari [61] considered deterministic total communication complexity on cases of $G$ being a path, grid or ring graph. There has been a recent surge of interest for proving lower bounds on total communication for the case when $G$ is a star [14, 55, 63]. This work was generalized to arbitrary topology by Chattopadhyay et al. [19] who proved tight bounds for certain functions for *all* network topologies. A followup work extended the results to some more functions [20]. Results on round complexity in this setup were recently obtained [18].

# D MISSING DETAILS IN SECTION 2

## D.1 GYO-GHD is a reduced GHD

The correctness of Construction 2.9 follows from the facts that the *GYO-reduction* of any $\mathcal{H}$ is unique [21] and the hyperedges removed while running GYOA form an acyclic forest (Lemma 4.8 in [44]). We define $\mathrm{F}(\mathcal{H})$ as the union of all vertices in all hyperedges in the acyclic forest excluding the roots (as they are included in $\mathrm{C}(\mathcal{H})$). To complete our construction, we need to argue that $\mathcal{T}$ is a reduced-GHD. This follows from our construction i.e., edge $e \in \overline{\mathcal{E}}$ satisfies either $e \subseteq V(\mathrm{C}(\mathcal{H}))$ (or) $e \subseteq V(\mathrm{F}(\mathcal{H}))$ and in both these cases, there always exists a node $v'_e$ in $\mathcal{T}$ such that $\chi(v'_e) =$. We argued this already for $\mathrm{C}(\mathcal{H})$ and for $\mathrm{F}(\mathcal{H})$, this follows from the definition of acyclicity.

## D.2 Example for Construction 2.9

Consider a hypergraph $\mathcal{H}_3$ with nodes $\overline{\mathcal{V}}(\mathcal{H}_3) = \{A, B, C, D, E, F, G, H\}$ and hyperedges

$$\overline{\mathcal{E}}(\mathcal{H}_3) = \{e_1 = (A, B, C), e_2 = (B, C, D), e_3 = (A, C, D), e_4 = (A, B, E), e_5 = (A, F), e_6 = (B, G), e_7 = (G, H)\}.$$

We now apply the GYO algorithm (GYOA) [31, 59, 66] on $\mathcal{H}$, which basically keeps performing the following two steps until it cannot. First, it checks if there is a node that is present in one hyperedge and if so, eliminates it. Second, it deletes a hyperedge that is contained in another. We document the execution of GYOA on $\mathcal{H}$ here. Let $\overline{\mathcal{E}}'(\mathcal{H}_3) = \overline{\mathcal{E}}(\mathcal{H}_3)$.

- Choose $H$ as it is present in only one hyperedge $(G, H)$. Remove it and the reduced hypergraph now is $\overline{\mathcal{E}}'(\mathcal{H}3) = \{e_1, e_2, e_3, e_4, e_5, e_6, (G)\}$. Since the edge $(G)$ is subsumed by more than one hyperedge we can remove it from $\overline{\mathcal{E}}'(\mathcal{H}_3)$.
- Choose $G$ as it is present in only one hyperedge $(B, G)$. Remove it and the reduced hypergraph now is $\overline{\mathcal{E}}'(\mathcal{H}3) = \{e_1, e_2, e_3, e_4, e_5, (B)\}$. Since the edge $(B)$ is subsumed by more than one hyperedge we can remove it from $\overline{\mathcal{E}}'(\mathcal{H}_3)$.
- Choose $F$ as it is present in only one hyperedge $(A, F)$. Remove it and the reduced hypergraph now is $\overline{\mathcal{E}}'(\mathcal{H}3) = \{e_1, e_2, e_3, e_4, (A)\}$. Since the edge $(A)$ is subsumed by more than one hyperedge we can remove it from $\overline{\mathcal{E}}'(\mathcal{H}_3)$.
- Choose $E$ as it is present in only one hyperedge $(A, F)$. Remove it and the reduced hypergraph now is $\overline{\mathcal{E}}'(\mathcal{H}3) = \{e_1, e_2, e_3, (A, B)\}$. Since the edge $(A, B)$ is subsumed by more than one hyperedge we can remove it from $\overline{\mathcal{E}}'(\mathcal{H}_3)$.

The GYOA terminates after the final step since it cannot find any more variable that is contained in only one hyperedge. Let $\mathcal{T}$ be the GYO-GHD obtained from this procedure. The final edge set $\overline{\mathcal{E}}'$ returned by GYOA is $e_c = \{e_1, e_2, e_3\}$ and the acyclic forest removed in this process contains the edges $e_f = \{e_4, e_5, e_6, e_7\}$ and is rooted at $e_4$. The forest $\mathrm{F}(\mathcal{H}_3)$ is the union of all vertices in the set $e_f \setminus e_4$. We build the core $\mathrm{C}(\mathcal{H}_3)$ now with vertices that are union of edges in $e_c$ and $e_4$ (i.e., the root of acyclic forest). $\mathcal{T}$ is rooted at $r'$ with $\chi(r') = \cup_{i \in [3]} v(e_i) \cup v(e_4)$. We create new nodes $v'_{e_i}$ for every $i \in [4]$ and all of them are directly connected to $r'$ (i.e., edge $(r', v'_{e_i})$ is added to $E(\mathcal{T})$). Thus, $\mathcal{T}$ contains the nodes $r', v'_{e_1}, v'_{e_2}, v'_{e_3}, v'_{e_4}, e_5, e_6, e_7$. Since we do not enforce constraints on the remaining edges in $\mathcal{T}$ as long as it is a valid GHD, we show two sample GYO-GHDs that can be constructed out of this. The first has edge set $E(\mathcal{T}) \cup \{(r', e_5), (r', e_6), (e_6, e_7)\}$ (having two internal nodes) and the second has edge set $E(\mathcal{T}) \cup \{(e_4, e_5), (e_5, e_6), (e_6, e_7)\}$ (having three internal nodes). It's not too hard to see that both these are reduced-GHDs by Definition 2.5.

# E MISSING DETAILS IN SECTION 3

## E.1 Connecting $\tau_{\text{MCF}}$ and $\text{MinCut}(G, K)$

We want to show that under worst-case assignment of relations to players, the bounds $\tau_{\text{MCF}}(G, K, N')$ and $\frac{N'}{\text{MinCut}(G,K)}$ are within an $\widetilde{O}(1)$ factor of each other. We first show that this is the case and then argue that our upper and lower bounds are tight within a $\widetilde{O}(d)$ factor for a larger class of assignments.

Let $(A, B)$ be a cut that separates $K$ of size $\text{MinCut}(G, K)$. First, consider the assignment where half of the relations are assigned to one player $a$ in $A$ and the rest to another player $b$ in $B$. Note that in this case $\tau_{\text{MCF}}(G, K, N')$ is upper bounded by number of rounds needed to send $N'$ bits from (say) $a$ to $b$. By the max-flow-min-cut theorem, we know we can send $N'$ bits from $a$ to $b$ in $\frac{N'}{\text{MinCut}(G,K)} + d(a, b)$ rounds, where $d(a, b)$ is the distance between $a$ and $b$.

We now somewhat extend the class of assignments so that our upper and lower bounds are still within a factor $\widetilde{O}(d)$ of each other. Let $(A, B)$ be the cut as above. Now, let us assume we distribute the relations that embed the TRIBES instance so that the $m = \frac{n_2}{2 \cdot \log(n_2)}$ pairs $(S_i, T_i)$ of $\text{TRIBES}_{m,N}(\hat{S}, \hat{T})$, the $S_i$ are assigned to some players in $A$ and the $T_i$'s to players in $B$. The rest of the relations are divided equally among $A$ and $B$. Note that our lower bound still holds.

For the upper bound, we have to look at the multicommodity flow that needs to send $O(n_2 \cdot d \cdot N)$ bits of flow from all but one player to a designated player (who is assigned at least one of the $(S_i, T_i)$ in the hard instance for the lower bound). Each of the at most $n_2 \cdot d$ relations denote one "demand" of size $N$. The sparsity $S$ of the cut $(A, B)$ is defined as the ratio of the number of cut edges and the size of the maximum demand separated by cut. We have $S \geq \frac{\text{MinCut}(G,K)}{n_2 \cdot d \cdot N}$ since $\text{MinCut}(G, K)$ is the smallest cut and the maximum demand separated by any cut is at most $n_2 \cdot d \cdot N$. Using the celebrated result of Leighton and Rao [49], one can schedule this multi-commodity flow in $\widetilde{O}\left(\frac{n_2 \cdot d \cdot N}{\text{MinCut}(G,K)} + \Delta(G, K)\right)$ rounds, where $\Delta(G, K)$ is the largest distance between any two players in $K$.

| Notation | Meaning |
|---|---|
| $q$ | Join query $\{R_i, A_i\}_{i \in [k]}$ |
| $R_i / R_e / R_{\upsilon(e)}$ | Function/Relation |
| $r$ | Upper bound on arity of $\{R_i\}_{i \in [k]}$ |
| $A_i$ | Attribute set of relation $R_i$ |
| $A(q)$ | All attributes of $q$ |
| $n$ | Size of $A(q)$ |
| $k$ | Size of $q$ (number of relations) |
| $N$ | Upper bound on size of $R_i$ (number of tuples in relation) |
| $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ | Underlying (multi)-hypergraph of $q$ |
| $\text{F}(\mathcal{H})$, $\text{C}(\mathcal{H})$ | decomposition of $\mathcal{H}$ into forest $\text{F}(\mathcal{H})$ and core $\text{C}(\mathcal{H})$ using Construction 2.9. |
| $y(\mathcal{H})$ | $y(\mathcal{H}) = min_{\forall \mathcal{T}: \mathcal{T} \text{ is a GYO-GHD of } \mathcal{H}} y(\mathcal{T})$. |
| $n_2(\mathcal{H})$ | $\lvert V(\text{C}(\mathcal{H})) \rvert$ |
| $\langle \mathcal{T}, \chi, \lambda \rangle$ | Generalized hypertree decompositions (GHD) of $\overline{\mathcal{V}}$ |
| $\chi(v) \subseteq \overline{\mathcal{V}}$ | Subset of vertices of $\overline{\mathcal{V}}$ associated to each node $v \in V(\mathcal{T})$ |
| $\lambda(v) \subseteq \overline{\mathcal{V}}$ | Subset of hyperedges of $\overline{\mathcal{E}}$ associated to each node $v \in V(\mathcal{T})$ |
| $G = (V, E)$ | Communication graph |
| $K$ | At most $k$ terminal nodes in $G$ |
| $\tau_{\text{MCF}}(G, K, N')$ | Round complexity of routing $N' \log_2(N')$ bits from all players in $K$ to any one player in $K$. |
| $\text{ST}(G, K, \Delta)$ | $\Delta$ diameter Steiner tree packing |

**Table 2: Notations used in our work.**

# F MISSING DETAILS IN SECTION 4

## F.1 Proof of Lemma 4.4

Proof. For simplicity, we assume that $\mathcal{H}$ has only one tree with $y = y(\mathcal{H})$ internal nodes. Next we show that we can solve the BCQ problem on $\mathcal{H}$ by solving another BCQ problem on $\mathcal{H}'$ with $y-1$ internal nodes defined as follows. We remove the bottom most star $P = (v_1, \ldots, v_{|P|})$ (where $v_1$ is the center and $(v_2, \ldots, v_{|P|})$ are the leaves) from $\mathcal{H}$. We define $V(\mathcal{H}') = V(\mathcal{H}) \setminus (v_2, \ldots, v_{|P|})$ and $E(\mathcal{H}') = E(\mathcal{H}) \cup \{(v_1)\} \setminus \{(v_1, v_i) : i \in [2, |P|]\}$. Using arguments in Section 4.1.1, we can process $P$ in $O\left(\min_{\Delta \in [|V|]} \left(\frac{N}{\mathrm{ST}(G,K,\Delta)} + \Delta\right)\right)$ rounds and the result computed is $R'_P = \bigcap_{i=2}^{k} R'_{v_i}$, where $R'_{v_i} = \pi_{v_1}(R_{v_1,v_i})$. Finally, we set $R'_{v_1} = R'_P$ (while the remaining surviving relations remains the same). It is easy to see that BCQ on $\mathcal{H}$ is 1 iff BCQ on $\mathcal{H}'$ is 1. Note that $\mathcal{H}'$ is also a tree, which implies we can continue this process recursively until $\mathcal{H}'$ has only one node left. Thus, the final answer is given by $(R'_P \stackrel{?}{\neq} \emptyset)$ and the number of recursive calls is bounded by the number of internal nodes $y$. Further, if $\mathcal{H}$ is a forest, our argument can be applied individually on each tree, resulting in the upper bound (1). This completes the proof. □

## F.2 Proof of Lemma 4.5

Proof. We starty by considering $\mathrm{F}(\mathcal{H})$ (via Construction 2.9). Using the protocol in the Proof of Lemma 4.4 (stated above), we know that $O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N}{\mathrm{ST}(G,K,\Delta)} + \Delta\right)\right)$ rounds suffice to reduce $\mathcal{H}$ to an updated hypergraph $\mathcal{H}' = (V(\mathrm{C}(\mathcal{H})), E(\mathcal{H}) \cup \{(r')|r' \text{ is a root in } \mathrm{F}(\mathcal{H})\})$. Further, for each root $r'$ in $\mathrm{F}(\mathcal{H})$, the corresponding relation $R_{r'}$ is the set computed by the algorithm in the proof of Lemma 4.4. It is easy to check that BCQ on $\mathcal{H}$ has the same answer as BCQ on $\mathcal{H}'$.
We can now use the *trivial protocol* to solve $\mathrm{BCQ}_{\mathcal{H}',N}$ on $G$, which by Lemma 3.13 gives the upper bound of (2.1), completing the proof. □

## F.3 Proof of Theorem 4.8

We start by stating some standard results that we use in our proof and then prove our general lower bound.

*Existing Results:* We state two standard graph theory results that we will be use in our lower bound arguments.

Lemma F.1 (Moore's Bound [5]). *Every graph with $p > 2|V|$ edges has a cycle of length at most $\frac{2\cdot\log(|V|)}{\log\left(\frac{p}{|V|}-1\right)}$.*

Theorem F.2 (Turan's Theorem [6]). *If a graph $\mathcal{H}$ has $n'$ vertices and at most $n' \cdot d$ edges, then there always exists an independent set of size at least $\frac{n'}{d+1}$ in $\mathcal{H}$.*

*Our Results.* We are now ready to prove our general lower bound for all simple graphs $\mathcal{H}$.

Proof. For notational convenience, define $y = y(\mathcal{H})$ and $n_2 = n_2(\mathcal{H})$. Let $m = \max\left(\frac{y}{2}, \frac{n_2}{2\log(n_2)}\right)$. In general, as in Lemma 4.6, given $\mathcal{H}$ and a TRIBES instance $\mathrm{TRIBES}_{m,N}(\hat{S}, \hat{T})$ we construct a BCQ instance $q_{\mathcal{H},\hat{S},\hat{T}}$ such that $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\mathrm{TRIBES}_{m,N}(\hat{S}, \hat{T}) = 1$. To this end we need to "embed" the $m$ pairs of sets $(S_i, T_i)$ from $\mathrm{TRIBES}_{m,N}(\hat{S}, \hat{T})$ as relations in $q_{\mathcal{H},\hat{S},\hat{T}}$. For $m = \frac{y}{2}$, we embed the pairs $(S_i, T_i)$ in the forest $\mathrm{F}(\mathcal{H})$ as done in Lemma 4.6. For $m = \frac{n_2}{2\cdot\log(n_2)}$, we consider $\mathrm{C}(\mathcal{H})$. We then show that it must be the case that $\mathrm{C}(\mathcal{H})$ either includes $\left(\frac{n_2}{2\log(n_2)}\right)$ vertex-disjoint cycles (referred to as *Case 1*), or that it has an independent set of size $\Omega(n_2)$ (referred to as *Case 2*). In both cases, we show how one can embed $\frac{n_2}{2\log(n_2)}$ pairs $(S_i, T_i)$ of $\mathrm{TRIBES}_{m,N}(\hat{S}, \hat{T})$ in $C(\mathcal{H})$. In particular, we prove the following lemma.

Lemma F.3. *$\mathrm{C}(\mathcal{H})$ either includes $\left(\frac{n_2}{2\log(n_2)}\right)$ vertex-disjoint cycles (Case 1) or it has an independent set of size $\Omega(n_2)$ (Case 2).*

Proof. By definition, the average degree of $C(\mathcal{H})$ is at least two (because if there is a vertex in $\mathrm{C}(H)$ of degree at most one, then it should be part of $\mathrm{F}(\mathcal{H})$, which would contradict Construction 2.9). As long as the average degree is greater than 10, we can use Lemma F.1 to prove that there exists a cycle in $\mathrm{C}(\mathcal{H})$ of length at most $\log(n_2)$. We can remove this cycle from $\mathrm{C}(\mathcal{H})$ and recurse until the average degree is below 10. Let $w$ be the number of vertex-disjoint cycles we have collected. If $w \geq \left(\frac{n_2}{2\log(n_2)}\right)$ we are in Case 1. Otherwise, at some point we are left with an induced subgraph of $\mathrm{C}(\mathcal{H})$ of size at least $\frac{n_2}{2}$ and average degree at most 10. In this case, by Theorem F.2, we can find an independent set in the induced subgraph (and thus in $\mathrm{C}(\mathcal{H})$) of size at least $\Omega(n_2)$, which is Case 2. □

We now show separately for each case how to embed $\frac{n_2}{2\log(n_2)}$ pairs $(S_i, T_i)$ of $\mathrm{TRIBES}_{m,N}$ in $\mathrm{C}(\mathcal{H})$. We start with Case 2. In this case, for large enough $n_2$, $\mathrm{C}(\mathcal{H})$ has an independent set of size at least $\frac{n_2}{2\log(n_2)}$ consisting of nodes of degree at least two. We can thus use a proof identical to that given in Lemma 4.6 to construct the remaining relations of $\mathrm{BCQ}_{\mathcal{H},N}$ corresponding to $\mathrm{C}(\mathcal{H})$. Namely, the independent set of $\mathrm{C}(\mathcal{H})$ will play the role of the set $O$ in Lemma 4.6.

We now address Case 1. Consider a cycle $C$ in $C(\mathcal{H})$ and a pair of sets $(S_i, T_i)$ from $\text{TRIBES}_{m,N}$. Let $c_1, c_2, \ldots, c_\ell$ be the nodes in $C$. To embed $(S_i, T_i)$ in $C$ we first present $S_i$ not as a subset of $[N]$ but rather as a subset of $[\sqrt{N}] \times [\sqrt{N}]$ or alternatively as a relation $R_{S_i}$ over two attributes with domain $[\sqrt{N}]$. Similarly, we associate $T_i$ with a relation $R_{T_i}$ over two attributes with domain $[\sqrt{N}]$. We define the relation corresponding to edge $(c_1, c_2)$ in the cycle as $R_{S_i}$, the relation corresponding to edge $(c_3, c_2)$ as $R_{T_i}$ (note that we reverse the order of attributes for $R_{T_i}$), and the relations corresponding to the remaining cycle edges as $\{(i, i) : i \in [\sqrt{N}]\}$. Notice, with this assignment of relations to the edges it holds that $\text{DISJ}(S_i, T_i) = 1$ iff there is an assignment that satisfies all relations in the cycle. Indeed, if tuple $\mathbf{t}$ satisfies all relations on the cycle, then the pair $(\pi_{c_1}(\mathbf{t}), \pi_{c_2}(\mathbf{t}))$ is in $S_i$, the pair $(\pi_{c_3}(\mathbf{t}), \pi_{c_2}(\mathbf{t}))$ is in $T_i$, and it holds that $\pi_{c_3}(\mathbf{t}) = \pi_{c_4}(\mathbf{t}) = \cdots = \pi_{c_\ell}(\mathbf{t}) = \pi_{c_1}(\mathbf{t})$. Thus, we conclude that the pair $(\pi_{c_1}(\mathbf{t}), \pi_{c_2}(\mathbf{t}))$ is in $S_i$ and the pair $(\pi_{c_2}(\mathbf{t}), \pi_{c_1}(\mathbf{t}))$ is in $T_i$, which in turn implies that $\text{DISJ}(S_i, T_i) = 1$. Alternatively, if $\text{DISJ}(S_i, T_i) = 1$ then there exists a pair $(\alpha, \beta)$ such that $(\alpha, \beta) \in S_i \cap T_i$. We can now set $\mathbf{t}$ with $\pi_{c_3}(\mathbf{t}) = \pi_{c_4}(\mathbf{t}) = \cdots = \pi_{c_\ell}(\mathbf{t}) = \pi_{c_1}(\mathbf{t}) = \alpha$ and $\pi_{c_2}(\mathbf{t}) = \beta$ to satisfy all relations corresponding to the cycle.

We continue in a similar manner for each cycle $C$ in our collection of cycles. Namely, for each cycle, we define relations corresponding to a pair of sets from $\text{TRIBES}_{m,N}(\hat{S}, \hat{T})$ such that the sets intersect iff there is an assignment that satisfies the relations in $C$. Notice that the collections of pairs of sets $(S, T)$ corresponding to the cycle collection have pair-wise intersection iff there is an assignment that satisfies all the relations in the cycle collection. To complete the definition of $q_{\mathcal{H},\hat{S},\hat{T}}$ we still need to assign a relation to all edges in $\text{C}(\mathcal{H})$ that do not appear in any of the cycles in the collection. All such edges are assigned the complete relation $[\sqrt{N}] \times [\sqrt{N}]$ over 2 attributes of domain $[\sqrt{N}]$. Note that the complete relations assigned do not impose any restrictions on the possible tuples $\mathbf{t}$ that satisfy the relations corresponding to the collection of cycles, and thus we have successfully embedded $\frac{n_2}{2\log(n_2)}$ pairs $(S_i, T_i)$ of $\text{TRIBES}_{m,N}(\hat{S}, \hat{T})$ in $\text{C}(\mathcal{H})$.

Thus, we can conclude that $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\text{TRIBES}_{m,N}(\hat{S}, \hat{T}) = 1$, where $m = \max\left(\frac{y}{2}, \frac{n_2}{2\log(n_2)}\right)$. Since sum and max are within a factor 2 of each other, we can write $m \geq \frac{y}{4} + \frac{n_2}{4\log(n_2)}$. We can now apply ideas from the proof of Lemma 4.7 to obtain the required lower bound $\widetilde{\Omega}\left(\frac{(y+n_2)\cdot N}{\text{MinCut}(G,K)}\right)$, as desired. □

# G QUERIES WHEN $\mathcal{H}$ IS A $d$-DEGENERATE HYPERGRAPH OF ARITY AT MOST $r$

In this section, we consider BCQs whose underlying hypergraph $\mathcal{H}$ is $d$-degenerate with arity at most $r$. We prove upper and lower bounds that are tight within a factor of $\widetilde{O}(d^2 \cdot r^2)$ for computing $\text{BCQ}_{\mathcal{H},N}$ (for $N$ large enough compared to size of $G$ and for worst-case assignment of relations to players).

## G.1 Main Theorem

We state our main theorem here.

Theorem G.1. *For arbitrary $G$, subset of players $K$ and $d$-degenerate hypergraphs with arity at most $r$, we have*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N \cdot r}{\text{ST}(G, K, \Delta)} + \Delta\right) + \tau_{\text{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N)\right). \tag{11}$$

*Further, for $d$-degenerate hypergraphs $\mathcal{H}$, we have*

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{\frac{y(\mathcal{H})\cdot N}{r} + \frac{n_2(\mathcal{H})\cdot N}{dr}}{\text{MinCut}(G, K)}\right). \tag{12}$$

We prove this theorem in two steps. We first prove the upper bound (11), followed by the lower bound (12). Finally, the arguments in Appendix E.1 imply that our bounds are tight within a gap of $\widetilde{O}(d^2 \cdot r^2)$ (under worst-case assignment of relations to players and $N$ being large enough).

## G.2 Upper Bound

Our proof is similar in nature to that used in the arity two case. For the rest of the section, unless specified otherwise, let $\mathcal{T}$ be a GHD of $\mathcal{H}$ with the root bag being $\text{C}(\mathcal{H})$. Analogous to $y(\mathcal{H})$, let $y(\mathcal{T})$ denote the number of internal nodes in the tree of $\mathcal{T}$. In this section, we will prove our upper bounds in terms of $y(\mathcal{T})$. Since we do not assume anything about $\mathcal{T}$ beyond the fact that it has $\text{C}(\mathcal{H})$ as its root, our bound holds for the smallest $y(\mathcal{T})$ over all such GHDs $\mathcal{T}$. This by Definition 3.1 is *exactly* $y(\mathcal{H})$. We prove our bounds in terms of $y(\mathcal{T})$ since it makes the exposition simpler.

We compute $\text{BCQ}_{\mathcal{H},N}$ on GHD $\mathcal{T}$. We first consider the case when $\mathcal{T}$ is a star, which will be a basic building block for our algorithms for more general $\mathcal{H}$.

*G.2.1 $\mathcal{T}$ is a star.* Let $\mathcal{H}$ be an $\alpha$-acyclic hypergraph whose GHD $\mathcal{T}$ is a star of the form $P = (v_1, \ldots, v_k)$ with $v_1$ as the center. By Definition 2.6, $\mathcal{H}$ includes $k$ relations of the form $R_{\chi(v_i)}$ for every $i \in [k]$. Note that computing the corresponding BCQ query $\text{BCQ}_{\mathcal{H},N}$ can be solved via a *set-intersection problem* of computing $R'_P = \bigwedge_{i=2}^{k} R'_{v_i}$, where $R'_{v_i} = \{\mathbf{t} \in R_{\chi(v_1)} : \exists \mathbf{t}' \in$

$R_{\chi(v_i)}$ s.t. $\pi_{\chi(v_1)\cap\chi(v_i)}(\mathbf{t}) = \pi_{\chi(v_i)\cap\chi(v_1)}(\mathbf{t}')\}$. Note that the intersection is computed on the attribute set $\chi(v_1)$ (each entry in the sets is a $r$-dimensional vector) as opposed to a single attribute (as was the case for arity two) It is easy to see that the final output of the $\text{BCQ}_{\mathcal{H},N}$ instance is 1 if $R'_P \neq \emptyset$ and 0 otherwise.

We describe our algorithm (Algorithm 2) here. We first broadcast the relation $R_{\chi(v_1)}$ to all the remaining players containing relations $R_{\chi(v_i)}$ for every $i \in [2,k]$ in $G$. Then, each player containing $R_{\chi(v_i)}$ for every $i \in [2,k]$ computes $R'_{v_i}$. Finally, $R'_P = \bigcap_{i=2}^{k} R'_{v_i}$ is computed using known upper bounds on set intersection using Theorem 3.11. Using the fact that at most $O(\log_2(r \cdot D))$ bits are communicated in each round, we have the following result.

Corollary G.2. *For arbitrary graphs $G$ and subset of players $K$, we have*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O\left(\min_{\Delta\in[|V|]}\left(\frac{N}{\text{ST}(G,K,\Delta)} + \Delta\right)\right).$$

For the case when $G$ is a line with $k$ vertices, note that $\text{ST}(G,K,\Delta) = 0$ for every $\Delta > k-1$ and $\text{ST}(G,K,k-1) = 1$. In particular, this implies

Corollary G.3. *Let $\mathcal{H}$ be $\alpha$-acyclic, $\mathcal{T}$ be a star and $G$ be a line with $k$ vertices. Then,*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O(r \cdot N + k).$$

**Algorithm 2** Algorithm for $\mathcal{T}$ is a Star

1: **Input:** A star $P = (v_1, \ldots, v_k) \in \mathcal{T}$ and relations $\{R_{\chi(v_i)} : i \in [k]\}$. Note that $v_1$ is the center and the others are leaves.
2: **Output:** $R'_P$
3: The player containing $R_{\chi(v_1)}$ broadcasts it to all players in $G$.
4: For every $i \in [2,k]$, the player containing $R_{\chi(v_i)}$ computes $R'_{v_i} = \{\mathbf{t} \in R_{\chi(v_1)} : \exists \mathbf{t}' \in R_{\chi(v_i)} \text{ s.t. } \pi_{\chi(v_1)\cap\chi(v_i)}(\mathbf{t}) = \pi_{\chi(v_i)\cap\chi(v_1)}(\mathbf{t}')\}$ internally.
5: $R'_P = \bigcap_{i=2}^{k} R'_{v_i}$ is computed using Theorem 3.11.
6: **return** $R'_P$

*G.2.2 $\mathcal{H}$ is an $\alpha$-acyclic forest.* Similar to the proof of Lemma 4.4, we use the analysis on the star to obtain better upper bounds for $\mathcal{H} = \text{F}(\mathcal{H})$.

Lemma G.4. *For arbitrary graphs $G$, subset of players $K$ and any GHD $\mathcal{T}$, we have*

$$\mathcal{D}(\text{BCQ}_{\mathcal{H},N}, G, K) = O\left(y(\mathcal{T}) \cdot \min_{\Delta\in[|V|]}\left(\frac{N}{\text{ST}(G,K,\Delta)} + \Delta\right)\right). \tag{13}$$

*Here, $y(\mathcal{T})$ denotes the number of internal nodes in $\mathcal{T}$.*

Proof. We start with a proof sketch. We keep removing stars from $\mathcal{T}$ in a bottom-up fashion. We solve the induced query on each removed star $P$ using the analysis of Section G.2.1. Since the number of stars we remove in this process is $y(\mathcal{T})$, the stated bound follows.

We now formalize our idea using a recursive algorithm. For simplicity, we assume that $\mathcal{H}$ has only one $\alpha$-acyclic hypergraph with its corresponding $\mathcal{T}$ having $y(\mathcal{T})$ internal nodes. Next, we show that we can solve BCQ problem on $\mathcal{H}$ by solving another BCQ problem on $\mathcal{H}'$ with $y(\mathcal{T}) - 1$ internal nodes defined as follows. We remove the bottom-most star $P = (v_1, \ldots, v_{|P|})$ (where $v_1$ is the center and $(v_2, \ldots, v_{|P|})$ are the leaves) from $\mathcal{T}$. We define $V(\mathcal{T}') = V(\mathcal{T}) \setminus (v_2, \ldots, v_{|P|})$ and $E(\mathcal{T}') = E(\mathcal{T}) \setminus \{(v_1, v_i) : i \in [2, |P|]\}$. This implies that $\mathcal{H}'$ is updated as follows – $V(\mathcal{H}') = V(\mathcal{H}) \setminus \{\chi(v_i) \cap \chi(v_1) : i \in [2,|P|]\}$ and $E(\mathcal{H}') = E(\mathcal{H}) \setminus \{\chi(v_i) : i \in [2,|P|]\}$. Using the arguments of Section G.2.1, we process $P$ in $O\left(\min_{\Delta\in[|V|]}\left(\frac{N}{\text{ST}(G,K,\Delta)} + \Delta\right)\right)$ rounds and compute is $R'_P = \cap_{i=2}^{k} R'_{v_i}$, where $R'_{v_i} = \{\mathbf{t} \in R_{\chi(v_1)} : \exists \mathbf{t}' \in R_{\chi(v_i)} \text{ s.t. } \pi_{\chi(v_1)\cap\chi(v_i)}(\mathbf{t}) = \pi_{\chi(v_1)\cap\chi(v_i)}(\mathbf{t}')\}$. Finally, we set $R'_{\chi(v_1)} = R'_P$ (while the remaining surviving relations remain the same). It is easy to see that BCQ on $\mathcal{H}$ is 1 iff BCQ on $\mathcal{H}'$ is 1.

Note that $\mathcal{H}'$ is also $\alpha$-acyclic, which implies that we can continue this process recursively until $\mathcal{T}'$ has only one node left (in which case we just check if its relation is empty or not). Thus, the final answer is given by $(R'_P \stackrel{?}{\neq} \emptyset)$ and the number of recursive calls is bounded by the number of internal nodes $y(\mathcal{T})$. Further, if $\mathcal{H}$ is a forest of $\alpha$-acyclic hypergraphs, our argument can be applied individually on every hypergraph, resulting in the upper bound (13). This completes the proof. □

*G.2.3 $d$-degenerate Hypergraphs $\mathcal{H}$ with arity at most $r$.* In this section, we prove our general upper bound result when $\mathcal{H}$ is a $d$-degenerate graph of arity at most $r$.

Lemma G.5. *For arbitrary $G$, subset of players $K$, and any GHD $\mathcal{T}$, we have*

$$\mathcal{D}(\mathrm{BCQ}_{\mathcal{H},N}, G, K) = O\left(y(\mathcal{T}) \cdot \min_{\Delta \in [|V|]} \left(\frac{r \cdot N}{\mathrm{ST}(G,K,\Delta)} + \Delta\right) + \tau_{\mathrm{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N)\right). \tag{14}$$

*Here, $y(\mathcal{T})$ denotes the number of internal nodes in $\mathcal{T}$.*

Proof. We start with a proof sketch. We decompose $\mathcal{H}$ into two components – an $\alpha$-acyclic forest ($\mathrm{F}(\mathcal{H})$) and a core ($\mathrm{C}(\mathcal{H})$). We then use Lemma G.4 to solve the induced query on $\mathrm{F}(\mathcal{H})$. For $\mathrm{C}(\mathcal{H})$, we use the *trivial protocol* that sends all the remaining relations to one player.

More formally, consider $\mathrm{F}(\mathcal{H})$ (via Construction 2.9). Using the protocol in Lemma G.4, we know that $O\left(y(\mathcal{T}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N \cdot r}{\mathrm{ST}(G,K,\Delta)} + \Delta\right)\right)$ rounds suffice to reduce $\mathcal{H}$ (with corresponding GHD $\mathcal{T}$) to $\mathrm{C}(\mathcal{H})$. Further, the protocol returns relations of the form $R'_r$ for every root $r$ in $\mathrm{F}(\mathcal{H})$. In particular, since $\chi(r)$ resides with $\mathrm{C}(\mathcal{H})$ for each such root, it is easy to check that BCQ on $\mathcal{H}$ has the same answer as BCQ on $\mathrm{C}(\mathcal{H})$.

We can now use the *trivial protocol* to solve $\mathrm{BCQ}_{\mathrm{C}(\mathcal{H}),N}$ on $G$ with $\tau_{\mathrm{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N)$ via Lemma 3.13. Note that our choice of $\mathcal{T}$ was arbitrary, which implies $y(\mathcal{T}) \geq y(\mathcal{H})$ (by Definition 3.1). Thus, we have an upper bound of (2.1), completing the proof. □

## G.3 Lower Bounds

We start with an overview of our lower bound. Then, we prove lower bounds for the case when $\mathcal{H}$ is a forest of $\alpha$-acyclic hypergraphs (i.e $\mathcal{H} = \mathrm{F}(\mathcal{H})$). Finally, we use the argument for $\mathrm{F}(\mathcal{H})$ to obtain our lower bounds for general $\mathcal{H}$. As in Section 4.2, our lower bounds follow from a reduction from the well-studied TRIBES function.

*G.3.1 Preliminaries and Notation.* We define the concept of Strong Independent Sets (including a lower bound on their size) and introduce a specific construction of GHDs, which we will use in our lower bound arguments.

Definition G.6 (Strong Independent Set). *Given a hypergraph $\mathcal{H}$, a strong independent set $I \subseteq V(\mathcal{H})$ satisfies the following property. For any pair of vertices $u, w \in I, u \neq w$, there exists no hyperedge $e \in E(\mathcal{H})$ with $\{u, w\} \subseteq e$.*

Theorem G.7 (Size of Strong Independent Set [34]). *Any $d$-degenerate hypergraph $\mathcal{H}$ with arity at most $r$ has a strong independent set of size at least $\frac{|V(\mathcal{H})|}{d \cdot (r-1)}$.*

Construction G.8 (MD-GHD). *Let $\mathcal{T}'$ be a GHD of $\mathcal{H}$ (recall that we mean GYO-GHDs obtained by Construction 2.9 when we say GHD). We now construct our GHD $\mathcal{T}$ from $\mathcal{T}'$ as follows. We first set $V(\mathcal{T}) = V(\mathcal{T}')$, $E(\mathcal{T}) = E(\mathcal{T}')$ and modify $\mathcal{T}'$ as follows. Consider any parent-child pair $(u, v) \in E(\mathcal{T})$, where $u$ is the parent and $v$ is the child. If there exists a node $w \in V(\mathcal{T})$ that occurs above $u$ in $\mathcal{T}$ such that $\chi(v) \cap \chi(u) \subseteq \chi(w)$,[22] we perform a modification as follows. We delete the edge $e_1 = (u, v)$ from $E(\mathcal{T})$ and add the edge $e_2 = (w, v)$ to it. Note that the subtree rooted at $v$ is still preserved. We continue this process until this operation cannot be performed.*

It is easy to see that $\mathcal{T}$ is a valid GHD according to definition 2.5.

*G.3.2 Lower Bounds for $\mathrm{F}(\mathcal{H})$.*

Theorem G.9. *When $\mathcal{T}$ is a MD-GHD for $\mathrm{F}(\mathcal{H})$, we have*

$$\mathrm{TRIBES}_{\frac{y(\mathcal{T})}{r},N} \leq \mathrm{BCQ}_{\mathcal{H},N}.$$

*Here, $y(\mathcal{T})$ is the number of internal nodes in $\mathcal{T}$.*

Proof. Given $\mathcal{H}, \mathcal{T}$ and a $\mathrm{TRIBES}_{\frac{y(\mathcal{T})}{r},N}$ instance we design a corresponding $\mathrm{BCQ}_{\mathcal{H},N}$ instance. Let $U = \{u_1, \ldots, u_{y(\mathcal{T})}\}$ be the set of internal nodes in $\mathcal{T}$ indexed in a bottom up fashion. Namely, if $u_i$ is a descendant of $u_j$ then $j > i$. In increasing index order $i$, for each internal node $u_i \in U$, we consider the star $P_i = (v_1 = u_i, v_2, \ldots, v_{|P_i|})$ with $v_1$ as the center and $(v_2, \ldots, v_{|P_i|})$ as leaves.

We first claim for $i = 1$ that there exists at least one attribute $p_1$ in the set $\cup_{j=2}^{|P_1|} \left(\chi(v_1) \cap \chi(v_j)\right)$ that does not occur anywhere in $\mathcal{T} \setminus P_1$. This implies, for this fixed $p_1$, that there exist at least two relations incident on it (one of them being $R_{\chi(v_1)}$). Note that if $p_1$ occurs in $\mathcal{T} \setminus P_1$, then we have a contradiction to the fact that $\mathcal{T}$ is a MD-GHD. In particular, this implies $\{\chi(v_1) \cap \chi(v_i) : i \in [2, |P_1|]\} \subseteq \chi(w)$, where $w = \mathrm{parent}(v_1)$ denotes the parent of $v_1$ (if it exists) in $\mathcal{T}$, which follows from RIP of GHDs. Thus, all nodes $(v_2, \ldots, v_{|P_1|})$ could have been made children of $w$ along with $v_1$, which means Construction G.8 would not have terminated. For $i = 2$, we apply the above argument on $\mathcal{T} = (V(\mathcal{T}) \setminus \{v_2, \ldots, v_{|P_1|}\}, E(\mathcal{T}) \setminus \{(v_1, v_i) : i \in [2, |P_1|]\})$ for $p_2$, and similarly we continue recursively for $i > 2$ and $p_i$.

We have shown above that for each $u_i \in U$, there exists an attribute $p_i$ such that there are at least two relations $R_{p_{i,1}}, R_{p_{i,2}}$ on hyperedges $p_{i,1} \neq p_{i,2} \in E(\mathcal{H})$ incident on $p_i$, and in addition, $p_i$ does not occur in (any bag of) $\mathcal{T} = V(\mathcal{T}) \setminus (\cup_{\ell < i} P_\ell)$. We now

[22] If there are multiple choices, we pick the topmost $w$ among them.

consider the set $P = \{p_1, \ldots, p_{|y(\mathcal{T})|}\}$ of attributes and claim that $P$ includes a strong independent set $I$ of size at least $\frac{|y(\mathcal{T})|}{r}$. We construct such a set greedily. We use the following observation in our analysis: for any $p_i$,

$$|\{p_j \mid j > i \text{ and } \exists e \in E(\mathcal{H}) \text{ s.t. } p_i, p_j \in e\}| \leq r - 1.$$

Assume otherwise, then there exists $p_i$ that shares edges with $r$ attributes $p_j$ for $j > i$. By the discussion above, as such edges include $p_i$ they must be associated with $\cup_{\ell<i} P_\ell$. This implies, via the RIP, that these $r$ attributes together with $p_i$ are in $\chi(u_i)$ in contradiction to $|\chi(u_i)| \leq r$. We now start the greedy construction with $I = \{p_1\}$, and remove $p_1$ from $P$ together will all $p_j$ that share an edge with $p_1$. We have removed at most $r$ attributes from $P$. We continue recursively. At step $\ell$ we consider the smallest index $i$ for which $p_i$ has not been removed from $P$. We add $p_i$ to $I$ and remove $p_i$ and any $p_j$ that shares an edge with $p_i$ from $P$. As all $p_\ell$ for $\ell < i$ have been removed from $P$, we only remove $r$ additional nodes from $P$. By the greedy process, the final $I$ is an independent set of size at least $\frac{|y(\mathcal{T})|}{r}$.

Assume w.l.o.g. that the strong independent set $I$ satisfies $|I| = \frac{y(\mathcal{T})}{r}$ (otherwise we take a subset of $I$). Associating a pair of sets $(S_i, T_i)$ from $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T})$ with each node $p_i \in I$, we have

$$\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T}) = \bigwedge_{p_i \in I} \text{DISJ}_N(S_i, T_i). \tag{15}$$

We now construct a corresponding $\text{BCQ}_{\mathcal{H},N}$ instance in detail. We start by defining a pair of relations corresponding to each pair $(S_i, T_i)$. Recall that each $p_i \in I$ corresponds to a $u_i \in U$, such that $p_i \in \chi(u_i)$ and $\exists u_i' \in \text{children}(u_i) : p_i \in \chi(u_i')$. We set the relations $R_{S_i} = S_i \times_{i=2}^{|\chi(u_i)|} \{1\}$ and $R_{T_i} = T_i \times_{i=2}^{|\chi(u_i')|} \{1\}$ (where, for both, the first attribute is $p_i$). In particular, we have $\text{attr}(R_{S_i}) = \chi(u_i)$ and $\text{attr}(R_{T_i}) = \chi(u_i')$. Further, we treat $S_i$ and $T_i$ as subsets of $[N]$ (instead of elements in $\{0,1\}^N$). Thus, $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T}) = 1$ iff for each $p_i \in I$, the join $R_{S_i} \bowtie R_{T_i}$ is not empty. To complete the description of $\text{BCQ}_{\mathcal{H},N}$, we need to define the other relations in $\mathcal{H}$ as well. Note that all the remaining relations $R' = \{\{R_e : e \in E(\mathcal{H})\} \setminus \{R_{S_i} \cup R_{T_i} : p_i \in I\}\}$ can be incident on only at most one $p_i \in I$ (as $I$ is a strong independent set). If $R_e \in R'$ is incident on $p_i \in I$, we define $R_e = \{(\ell, 1, \ldots, 1) : \ell \in [N]\}$ (where $p_i$ is the first attribute in $e$). Otherwise, we set $R_e = (1, \ldots, 1)$ (note that the order of attributes does not matter here). Let us denote the BCQ instance constructed above by $q_{\mathcal{H},\hat{S},\hat{T}}$.

To complete the proof, we show that $q_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T}) = 1$. In particular, if $q_{\mathcal{H},\hat{S},\hat{T}} = 1$, there exists a tuple $\mathbf{t} \in \prod_{v \in V(\mathcal{H})} \text{Dom}(v)$ that satisfies all relations in $q_{\mathcal{H},\hat{S},\hat{T}}$ i.e., $\mathbf{t}_e \in R_e$ for every $e \in E(\mathcal{H})$. Specifically, for each $p_i \in I$, $R_{S_i} \bowtie R_{T_i}$ is not empty, which implies $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T}) = 1$. Alternatively, if $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T}) = 1$, we can find tuple $\mathbf{t} \in \prod_{v \in V(\mathcal{H})} \text{Dom}(v)$ that satisfies all relations in $q_{\mathcal{H},\hat{S},\hat{T}}$. For each $p_i \in I$, we set $\pi_{p_i}(\mathbf{t})$ to be in the intersection of $S_i$ and $T_i$, and for all remaining nodes $v \in V(\mathcal{H}) \setminus I$ we set $\pi_v(\mathbf{t}) = 1$. Note that this implies all the relations in $q_{\mathcal{H},\hat{S},\hat{T}}$ are satisfied. This concludes our proof. □

Note that the above argument was independent of $G$. We now use the structure of $G$ to obtain a lower bound on $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K)$ using known results for $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}$.

*G.3.3 Lower Bounds dependent on G.* We show the following lower bound for arbitrary $G$ assuming worst-case assignment of relations to players in $K$.

Lemma G.10 (Arbitrary $G$). *If $\mathcal{H} = \text{F}(\mathcal{H})$, then*

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot N}{r \cdot \text{MinCut}(G)}\right).$$

Proof. We first consider a min-cut $(A, B)$ of $G$ that separates $K$, where $A$ and $B$ denote the set of vertices in each partition ($A \cup B = V(G)$). Using the notation used in the proof of Theorem G.9, let $q_{\mathcal{H},\hat{S},\hat{T}}$ be the query computed on a MD-GHD $\mathcal{T}$ corresponding to a given instance $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T})$. We assign relations $\{R_{S_i}\}_{p_i \in I}$ to vertices in $A$ and relations $\{R_{T_i}\}_{p_i \in I}$ to vertices in $B$. The other relations in $q_{\mathcal{H},\hat{S},\hat{T}}$ can be assigned arbitrarily. Note that any protocol to compute $q_{\mathcal{H},\hat{S},\hat{T}}$ on $G$ gives a two-party protocol (Alice, Bob) for $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S}, \hat{T})$. In particular, Alice gets the sets $\{S_i\}_{p_i \in I}$ (corresponding to $R_{S_i}$) assigned to vertices in $A$ and Bob gets the sets $\{T_i\}_{p_i \in I}$ (corresponding to $R_{T_i}$) assigned to vertices in $B$ (ignoring the additional relations). It is not too hard to see that if there exists a $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K)$ round protocol for $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}$ on $G$, then we have a two-party protocol (i.e., on a graph $\mathcal{G} = (\{a, b\}, (a, b))$) with at most $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \cdot \text{MinCut}(G, K) \cdot \lceil \log_2(\text{MinCut}(G, K)) \rceil$ rounds (see Proof of Lemma 4.7 for a detailed discussion). Since $\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \cdot \text{MinCut}(G, K) \cdot \lceil \log_2(\text{MinCut}(G, K)) \rceil$ is lower bounded from Theorem 2.3 by $\widetilde{\Omega}\left(\frac{y(\mathcal{T}) \cdot N}{r}\right)$ and (since by definition of $y(\mathcal{H})$) $y(\mathcal{T}) \geq y(\mathcal{H})$, we conclude our assertion. □

Remark G.11. *In the proof above, we invoked an existing lower bound on Theorem 2.3 for* $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S},\hat{T})$. *We would like to remark here that the lower bound on* TRIBES *is indeed obtained on a product distribution* $\hat{\mathbb{D}}$ *on* $\frac{y(\mathcal{T})}{r}$ *variables. All our arguments use* $\hat{\mathbb{D}}$ *as a black-box but such a* $\hat{\mathbb{D}}$ *always exists (e.g., one of them is defined in Section* 2.1 *in [19]).*

Inspecting the hard distribution defined in Section 2.1 in [19] (which is also used in [36] to prove Theorem 2.3) we observe the following:

Remark G.12. *For every pair of sets* $(S_j, T_j)$ *in the* $\text{TRIBES}_{\frac{y(\mathcal{T})}{r},N}(\hat{S},\hat{T})$ *instance, we have*

$$|S_j \cap T_j| \leq 1,$$

*where* $j \in [\frac{y(\mathcal{T})}{r}]$.

Note that this implies the following based on our lower bound arguments.

Remark G.13. *For all our* $\text{BCQ}_{\mathcal{H},N}$ *instances, we have* $|\bowtie_{e\in E(\mathcal{H})} R_e| \leq 1$.

*G.3.4 Lower Bounds for d-degenerate hypergraphs* $\mathcal{H}$. We are now ready to prove our general lower bound for all $d$-degenerate hypergraphs $\mathcal{H}$.

Theorem G.14. *For arbitrary* $G$, *subset of players* $K$ *and* $d$*-degenerate hypergraphs* $\mathcal{H}$ *with a MD-GHD* $\mathcal{T}$, *we have*

$$\mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K) \geq \widetilde{\Omega}\left(\frac{\left(\frac{y(\mathcal{T})}{r} + \frac{n_2(\mathcal{H})}{d\cdot r}\right)\cdot N}{\text{MinCut}(G,K)}\right). \tag{16}$$

*Here,* $y(\mathcal{T})$ *denotes the number of internal nodes in* $\mathcal{T}$.

Proof. Let $m_1 = \frac{y(\mathcal{T})}{r}$ and $m_2 = \frac{n_2(\mathcal{H})}{d\cdot r}$. We obtain two independent lower bounds on $\mathcal{H}$ and our final bound is the maximum between them (which is at least half their sum). In general, as in Theorem G.9, given $\mathcal{H}$ and a TRIBES instance $\text{TRIBES}_{m_i,N}(\hat{S},\hat{T})$ for every $j \in [2]$, we construct a BCQ query $q^{(j)}_{\mathcal{H},\hat{S},\hat{T}}$ on $\mathcal{H}$ such that $q^{(j)}_{\mathcal{H},\hat{S},\hat{T}} = 1$ iff $\text{TRIBES}_{m_j,N} = 1$. To this end, we need to "embed" the $m_j$ pairs of sets $(S_i, T_i)$ from $\text{TRIBES}_{m_j,N}(\hat{S},\hat{T})$ as relations in $q^{(j)}_{\mathcal{H},\hat{S},\hat{T}}$. Recall that $\text{C}(\mathcal{H})$ is present at the root of $T$. It is easy to check that one can apply the reduction on Theorem G.9 to construct $q^{(1)}_{\mathcal{H},\hat{S},\hat{T}}$ with the required properties.

Finally for $m_2$, we apply Theorem G.7 on the root of $T$ (i.e., $\text{C}(\mathcal{H})$) to obtain a strong independent set of size at least $\frac{n_2(\mathcal{H})}{d\cdot(r-1)} \geq \frac{n_2(\mathcal{H})}{d\cdot r}$ (since $r \geq 2$). We then use a proof identical to that given in Theorem G.9 to embed the TRIBES instance $\text{TRIBES}_{m_2,N}$ onto $\text{C}(\mathcal{H})$.

Let $m = \max(m_1, m_2) = \max\left(\frac{y(\mathcal{T})}{r}, \frac{n_2(\mathcal{H})}{d\cdot r}\right)$. Since sum and max are within a factor 2 of each other, we can write $m \geq \frac{y(\mathcal{T})}{2\cdot r} + \frac{n_2(\mathcal{H})}{2\cdot d\cdot r}$. We can now apply ideas from the proof of Lemma G.10 to obtain the required lower bound $\widetilde{\Omega}\left(\frac{\left(\frac{y(\mathcal{T})}{r}+\frac{n_2(\mathcal{H})}{d\cdot r}\right)\cdot N}{\text{MinCut}(G,K)}\right)$. This concludes our proof. □

We now prove Theorem G.1. The upper bound follows from Lemma G.5. For the lower bound, note that our bounds depend on an arbitrary MD-GHD $\mathcal{T}$ for $\mathcal{H}$. By definition 3.1, we have that $y(\mathcal{T}) \geq y(\mathcal{H})$ and the lower bound (12) follows. Using Definition 2.10, we have that upper and lower bounds match for the GHD that achieves the internal-node-width $y(\mathcal{H})$ (i.e., $y(\mathcal{H}) = y(\mathcal{H})$).

We conclude this section by noting that when $N \geq |V|^2$ our upper and lower bounds differ by $\widetilde{O}(d^2 \cdot r^2)$ factor (for worst-case assignments of relations to players). In particular, Theorem 3.10, implies that the first two terms in the upper and lower bounds match up to an $\widetilde{O}(r^2)$ factor. Using the same arguments as in Appendix E.1, we can show that for worst-case assignment of relations, we have the second terms in the upper and lower bounds differ by a $\widetilde{O}(d^2 \cdot r^2)$ factor, as desired.

# H BOUNDS FOR GENERAL FAQS AND ASSUMPTIONS IN MODEL 2.1

In this section, we prove Theorem 5.1 and address assumptions in Model 2.1. We start with the redefinition of the FAQ problem and state some known results.

## H.1 Preliminaries and Existing Results

We define the general FAQ problem here. We are given a multi-hypergraph $\mathcal{H} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$ where for each hyperedge $e \in \overline{\mathcal{E}}$, we are given an input function $f_e : \prod_{v\in e} \text{Dom}(v) \to \mathbb{D}$. In addition, we are given a set of *free variables* $\mathcal{F} \subseteq \overline{\mathcal{V}} : |\mathcal{F}| = \ell$. We would like

to note that our results hold only for specific choices of $\mathcal{F}$. For a fixed $\mathcal{F}$, the vertices in $\overline{\mathcal{V}}$ can be renumbered so that $\mathcal{F} = [\ell]$ WLOG. We would like to compute the function:

$$\phi\left(\mathbf{x}_{[\ell]}\right) = \bigoplus_{x_{\ell+1}\in \mathrm{Dom}(x_{\ell+1})}^{(\ell+1)} \cdots \bigoplus_{x_n \in \mathrm{Dom}(x_n)}^{(n)} \bigotimes_{S\in\overline{\mathcal{E}}} f_S(\mathbf{x}_S), \tag{17}$$

where we use $\mathbf{x} = (x_u)_{u\in\overline{\mathcal{V}}}$ and $\mathbf{x}_S$ is $\mathbf{x}$ projected down to co-ordinates in $S \subseteq \overline{\mathcal{V}}$. The variables in $\overline{\mathcal{V}} \setminus \mathcal{F}$ are called *bound variables*. For every bound variable $i > \ell$, $\oplus^{(i)}$ is a binary (aggregate) operator on the domain $\mathbb{D}$. Different bound variables may have different aggregates. Finally, for each bound variable $i > \ell$ either $\oplus^{(i)} = \otimes$ (*product aggregate*) or $(\mathbb{D}, \oplus^{(i)}, \otimes)$ forms a commutative semiring (*semiring aggregate*) with the same additive identity $\mathbf{0}$ and multiplicative identity $\mathbf{1}$. As with database systems, we assume that the functions are input in the *listing* representation, i.e. the function $f_e$ is represented as a list of its non-zero values: $R_e = \{(\mathbf{y}, f_e(\mathbf{y}))|\mathbf{y} \in \prod_{v\in e} Dom(v) : f_e(\mathbf{y}) \neq \mathbf{0}\}$. Let $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}=[\ell]}$ denote the class of FAQ problems, where each function $f_e$ for $e \in \overline{\mathcal{E}}$ has at most $N$ non-zero entries. Note that we are not explicitly $(\oplus^{(\ell+1)}, \ldots, \oplus^{(n)})$ since the developments here hold for all such choice of operators for the *bound* variables.

When $\oplus^{(i)} = \oplus$ for every $i \in [\ell+1, n]$ and $(\mathbb{D}, \oplus^{(i)}, \otimes)$ forms a commutative semiring, we have the FAQ-SS problem. We have already seen that BCQ and computing some Factor Marginals in PGMs are special cases of FAQ-SS. We restate them in the language of FAQ for completeness. When $\mathcal{F} = \emptyset$ and $\mathbb{D} = \{0, 1\}$ (i.e., the *Boolean semi-ring*), $\mathsf{FAQ}_{\{0,1\},\mathcal{H},N,\emptyset}$ corresponds to the *Boolean Conjunctive Query* which we denote by $\mathrm{BCQ}_{\mathcal{H},N}$. Further, if $\mathcal{F} = \overline{\mathcal{V}}$ and $\mathbb{D} = \{0, 1\}$, we have the *natural join* problem in Definition 3.4 and if $\mathcal{F} = e$ for any $e \in \overline{\mathcal{E}}$ with $\mathbb{D} = \{0, 1\}$, we have the *semijoin* problem in Definition 3.5. We would like to mention that $R_e$ can be equivalently represented as $R_{v(e)}$ and $R_i$ (denoting the $i$th edge/function in $|\overline{\mathcal{E}}| = k$).

We can use Theorem 9 from [38] to obtain the following result.

Corollary H.1. *If there exists a function $R_{e'}$ for $e' \in E(\mathcal{H})$ and other function $R_e$ for every $e \in E(\mathcal{H}) \setminus \{e'\}$ such that the set of attributes $v(e') \supseteq (z_1, \ldots z_w)$ satisfies $v(e)$ does not contain the attributes $(z_1, \ldots, z_w)$ for every $e \in E(\mathcal{H})$. Then, we have*

$$\left(\bigoplus_{g\in\mathrm{Dom}(g)}^{(g)}\right)_{g\in\cup_{e\in E(\mathcal{H})} v(e)} \left(\bigotimes_{e\in E(\mathcal{H})} R_e\right) = \left(\left(\bigoplus_{g\in\mathrm{Dom}(g)}^{(g)}\right)_{g\in\cup_{v\in V(\mathcal{H})\setminus\{z_1,\ldots,z_w\}} v} \bigotimes_{e\in E(\mathcal{H})\setminus\{e'\}} R_e\right) \otimes \left(\bigoplus_{z_1\in\mathrm{Dom}(z_1)}^{(z_1)} \cdots \bigoplus_{z_w\in\mathrm{Dom}(z_w)}^{(z_w)} R_{e'}\right). \tag{18}$$

We summarize the implication of the above corollary here. In particular, if there exists a relation $R_{e'}$ for $e' \in E(\mathcal{H})$ with $v(e') \supseteq (z_1, \ldots, z_w)$, then we can "push down" the aggregations $\left(\bigoplus_{z_1\in\mathrm{Dom}(z_1)}^{(z_1)} \cdots \bigoplus_{z_w\in\mathrm{Dom}(z_w)}^{(z_w)}\right)$ inside every tuple in $R_{e'}$.

We consider the standard centralized model and prove the following result when $\mathcal{H}$ is $\alpha$-acyclic. We would like to note that this result follows from FAQ/AJAR [38, 39] but we state it here explicitly for completeness. Further, this result will be used crucially in our distributed algorithm later.

Theorem H.2. *When $\mathcal{H}$ is $\alpha$-acyclic, the deterministic complexity of computing $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\emptyset}$ is $\widetilde{O}(N)$.*

Proof. For any input function $f$ such that $f_e : \prod_{v\in e} \mathrm{Dom}(v) \to \mathbb{D}$ and an arbitrary set of operators $(\oplus^{(i)})_{i\in[n]}$, we can write

$$Q = \bigoplus_{x_1\in\mathrm{Dom}(x_1)}^{(1)} \cdots \bigoplus_{x_n\in\mathrm{Dom}(x_n)}^{(n)} \bigotimes_{e\in E(\mathcal{H})} R_e \tag{19}$$

using (3). Here, $Q$ is an instance of $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\emptyset,(\oplus^{(i)})_{i\in[n]}}$. Recall that $R_e$ is the *listing representation* of $f_e : \{(\mathbf{y}, f_e(\mathbf{y}))|\mathbf{y} \in \prod_{u\in e} Dom(u) : f_e(\mathbf{y}) \neq \mathbf{0}\}$ for every $e \in E(\mathcal{H})$. We now use Construction 2.9 on $\mathcal{H}$ obtaining a GHD $\mathcal{T}$ where each node $v \in \mathcal{T}$ corresponds to a hyperedge $\chi(v) \in E(\mathcal{H})$ (see Definition 2.6).

We describe the algorithm here, which uses a message-passing algorithm (upward pass) on a GHD $\mathcal{T}$. In particular, in a bottom-up fashion, every node $v \in \mathcal{T}$ performs two computations – first, it updates the relation $R_{\chi(v)}$ based on the messages received from all its neighbors. Second, if it is not a root, then it computes the message $v$ needs to send to its parent $v' = \mathrm{parent}(v)$. We obtain the final answer for $Q$ in the root.

We now formalize the algorithm above. Since we are considering the centralized model, we can assume that all relations $R_{\chi(v)}$ : $v$ is a node in $\in \mathcal{T}$ can be accessed at any point in time without any additional communication. Let $v$ be current node in consideration in our algorithm. We update the relation $R_{\chi(v)}$ as follows:

$$R_{\chi(v)} = R_{\chi(v)} \cdot \bigotimes_{u\in\Gamma(v)} m_{u,v}, \tag{20}$$

where $\Gamma(v)$ and $m_{u,v}$ denote the neighborhood of $v$ and the message sent from $u$ to $v$ respectively. Initialize $w = \mathbf{1}$. For every tuple $\mathbf{t} \in R_{\chi(v)}$ and for all tuples $\mathbf{t'} \in m_{u,v}$ with $\pi_{\chi(u)\cap\chi(v)}(\mathbf{t'}) = \pi_{\chi(u)\cap\chi(v)}(\mathbf{t})$ for every $u \in \Gamma(v)$, we compute the running product $w = w \cdot f(\mathbf{t'})$. Then, the tuple $\mathbf{t}$ in $R_{\chi(v)}$ is updated as $(\mathbf{t'}, f(\mathbf{t}) \cdot w)$. Since $|R_{\chi(v)}| \leq N$ and $|m_{u,v}| \leq N$ with $\chi(v) \subseteq \chi(u) \cap \chi(v)$ for every $v \in \mathcal{T}, u \in \Gamma(v)$, we claim that (20) can be computed in $\widetilde{O}(N)$ time. To prove this, observe that for a fixed tuple $\mathbf{t} \in R_{\chi(v)}$, there exists at most one tuple $\mathbf{t'} \in m_{u,v}$ such that $\pi_{\chi(u)\cap\chi(v)}(\mathbf{t'}) = \pi_{\chi(u)\cap\chi(v)}(\mathbf{t})$ for every $u \in \Gamma(v)$. Then, we traverse through all tuples in $R_{\chi(v)}$in the worst-case and our stated claim follows. We call this *Step 1.*

If $v$ is not the root of $\mathcal{T}$, the message $m_{v,v'}$ that $v$ needs to send to its parent $v' = \text{parent}(v) \in \mathcal{T}$ is computed as follows. Notice that the variables in the set $\chi(v) \setminus \chi(v') = (z_1, \ldots, z_w)$ are *private* to the node $v$. In particular, all variables in $\chi(v) \setminus \chi(v')$ are not present anywhere apart from the subtree of $\mathcal{T}$ rooted at $v$ (follows from the running intersection property of $\mathcal{T}$). Notice that the attributes $(z_1, \ldots, z_w)$ are present in $\subseteq (x_1, \ldots, x_n)$. Consider the *reduced* FAQ query at $v$ given by

$$Q_v = \left( \bigoplus_{g \in \text{Dom}(g)}^{(g)} \right)_{g \in \cup_{y \in \mathcal{T}'} \chi(y)} \otimes \underset{y \in \mathcal{T}'}{} R_{\chi(y)},$$

where $\mathcal{T}'$ denotes the set of nodes that haven't been processed in $\mathcal{T}$ so far in the message-up algorithm (which includes $v$). We can rewrite $Q_v$ by invoking Corollary H.1 as follows:

$$Q_v = \left( \left( \bigoplus_{g \in \text{Dom}(g)}^{(g)} \right)_{g \in \cup_{y \in \mathcal{T}' \setminus \{v\}} \chi(y)} \prod_{y \in \mathcal{T}' \setminus \{v\}} R_{\chi(y)} \right) \otimes \left( \bigoplus_{z_1 \in \text{Dom}(z_1)}^{(z_1)} \cdots \bigoplus_{z_w \in \text{Dom}(z_w)}^{(z_w)} R_{\chi(v)} \right). \tag{21}$$

In particular, we are "pushing down" the aggregations $\left( \bigoplus_{z_1 \in \text{Dom}(z_1)}^{(z_1)}, \ldots, \bigoplus_{z_w \in \text{Dom}(z_w)}^{(z_w)} \right)$ inside every tuple in the relation $R_{\chi(v)}$ since they are not contained in any relation $R_{\chi(y)}$ for every $y \in \mathcal{T}' \setminus \{v\}$. In other words, the attributes $(z_1, \ldots, z_w)$ belong to only relations in the subtree of $\mathcal{T}$ rooted at $v$. Further, observe that this computation is performed at node $v$. Note that the aggregations are computed on the annotated values of the relations as follows. For every tuple $\mathbf{t} \in \pi_{\chi(v)\cap\chi(v')} R_{\chi(v)}$, the tuple $\left( \mathbf{t}, \forall \mathbf{t'} \in R_{\chi(v)} : \bigoplus_{\pi_{z_1}(\mathbf{t'})}^{(z_1)} \cdots \bigoplus_{\pi_{z_w}(\mathbf{t'})}^{(z_w)} f(\mathbf{t'}) \text{ if } \pi_{\chi(v)\cap\chi(v')}(\mathbf{t'}) = \mathbf{t} \right)$ is added to the message $m_{v,v'}$. If $v$ is the root of $\mathcal{T}$, we have $\chi(v) = (z_1, \ldots, z_w)$ and as a result, (21) will have only the right hand side of the product. Thus, the final answer for $Q$ can be computed from $v$. Notice that this computation can be done in $\widetilde{O}(N)$ time since $|m_{v,v'}| \leq |R_{\chi(v)}|$ and we might traverse through all tuples in $R_{\chi(v)}$ in the worst case. We call this *Step 2.*

Finally, when the algorithm terminates, we need to argue that we obtain the correct result for $Q$. Consider the first node $v \in \mathcal{T}$ considered in our message up process. The *reduced* FAQ query $Q_v = Q$'s correctness follows from Corollary H.1. Since we repeatedly apply the same procedure for all other nodes in $v \in \mathcal{T} \setminus \{v\}$, the correctness follows. Since both *Step 1* and *Step 2* take only $\widetilde{O}(N)$ time and our choice of $(\oplus^{(i)})_{i \in [n]}$ was arbitrary, this completes our proof. □

## H.2 Main Theorem

We prove the following theorem in our model assuming that any hypergraph can be decomposed into a forest $\text{F}(\mathcal{H})$ and a core $\text{C}(\mathcal{H})$ using Construction 2.9.

Theorem H.3. *For arbitrary $G$, subset of players $K$, any $\mathcal{F} \subseteq V(\text{C}(\mathcal{H})) : |\mathcal{F}| = \ell$, and $d$-degenerate hypergraphs $\mathcal{H}$ with arity at most $r$, we have*

$$\mathcal{D}\left(\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K\right) = O\left( y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left( \frac{N \cdot r}{\text{ST}(G,K,\Delta)} + \Delta \right) + \tau_{\text{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N) \right). \tag{22}$$

*Further, we have*

$$\mathcal{R}\left(\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K\right) \geq \widetilde{\Omega}\left( \frac{y(\mathcal{H}) \cdot N}{r \cdot \text{MinCut}(G,K)} + \frac{n_2(\mathcal{H}) \cdot N}{d \cdot r \cdot \text{MinCut}(G,K)} \right). \tag{23}$$

*Both the results hold for any $\mathbb{D}$ and any choice of operators $(\oplus^{\ell+1}, \ldots, \oplus^{n})$ over $\mathbb{D}$ as defined in Section H.1.*

We would like to note here that for simple graphs $\mathcal{H}$, we can overcome the factor of $d$ in the lower bound (see Theorem 4.8). In particular, we can use similar ideas from there to prove Theorem 5.1.

For both the upper and lower arguments, we consider an arbitrary set of operators $(\oplus^{\ell+1}, \ldots, \oplus^{n})$ over $\mathbb{D}$.

## H.3 Upper Bound for General FAQs

The upper bound follows from a slight modification of our algorithm to compute $\text{BCQ}_{\mathcal{H},N}$ and uses ideas from the Proof of Theorem H.2 to "push down" a specific subset of operators in $(\oplus^{\ell+1}, \ldots, \oplus^{n})$. We present a proof sketch here. Let's fix an input function $f$ such that $f_e : \prod_{v \in e} \text{Dom}(v) \to \mathbb{D}$. Using (3), we can write

$$\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}} = \bigoplus_{x_{\ell+1} \in \text{Dom}(x_{\ell+1})}^{(\ell+1)} \cdots \bigoplus_{x_n \in \text{Dom}(x_n)}^{(n)} \left( \prod_{e \in E(\mathcal{H}) \setminus \text{F}(\mathcal{H})} R_{v(e)} \prod_{e' \in \text{F}(\mathcal{H})} R_{v(e')} \right)$$

since $\text{F}(\mathcal{H})$ is a sequence of hyperedges in $\mathcal{H}$. Recall that $R_e : e \in E(\mathcal{H})$ is the *listing representation* of $f_e : \{(\mathbf{y}, f_e(\mathbf{y})) | \mathbf{y} \in \prod_{u \in e} Dom(u) : f_e(\mathbf{y}) \neq \mathbf{0}\}$.

We use the same ideas from the Proof of Lemma G.4 but for each removed star $P$, we use Algorithm 3 to compute it. In particular, we show that computing $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ can be solved by computing the product $R'_P = \prod_{i=2}^{k} R'_{v_i}$. Note that this product can be computed on a Steiner tree using known results for *set-intersection*. We basically perform two steps – compute the intersection of tuples in each $R'_{v_i}$ and multiply the annotated values if there is a tuple present in every $R'_{v_i}$. It is easy to see that $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}} = R'_P$.

We describe our algorithm here. We perform a message-passing algorithm (upward pass) starting with a broadcast of the function $R_{\chi(v_1)}$ to all players in $G$. For every $i \in [2,k]$, the player containing $R_{\chi(v_i)}$ computes the up message $m_{v_i,v_1}$ it needs to send $v_1$ internally (Step 4 in the Algorithm). Notice that the variables in the set $\Gamma(v_{i,1}) = \{\chi(v_i) \setminus \chi(v_1)\} = (z_1, \ldots, z_w)$ are *private* to the node $v_i$. In particular, the variables in $\Gamma(v_{i,1})$ are not present anywhere in the remaining hypergraph. Further, note that $(z_1, \ldots, z_w) \subseteq (x_{\ell+1}, \ldots, x_n)$. We and "push down" the aggregations $\left( \bigoplus_{z_1 \in \text{Dom}(z_1)}^{(z_1)} , \ldots , \bigoplus_{z_\in \text{Dom}(z_w)}^{(z_w)} \right)$ inside every tuple in the function $R_{\chi(v_i)}$ since these variables do not occur anywhere in the remaining hypergraph. Further, the aggregations are computed on the annotated values of the relations as follows. In particular, for every tuple $\mathbf{t} \in \pi_{\chi(v_i) \cap \chi(v_1)} R_{\chi(v_i)}$, the tuple $\left( \mathbf{t}, \forall \mathbf{t}' \in R_{\chi(v_i)} : \bigoplus_{\pi_{z_1}(\mathbf{t}')}^{(z_1)} \ldots \bigoplus_{\pi_{z_w}(\mathbf{t}')}^{(z_w)} f(\mathbf{t}') \text{ if } \pi_{\chi(v_i) \cap \chi(v_1)}(\mathbf{t}') = \mathbf{t} \right)$ is appended to the message $m_{v_i,v_1}$.

Then, only one player retains the original $R_{\chi(v_1)}$ (say the player containing $R_{\chi(v_2)}$) and all others store an identity map of $R_{\chi(v_1)}$ (with all entries set to a function value of 1) to ensure we don't multiply $R_{\chi(v_1)}$ more than once. Finally, all the players containing $R_{\chi(v_i)} : i \in [2,k]$ compute $R'_{v_i} = R_{\chi(v_1)} \times m_{v_i,v_1}$ with their own version of $R_{\chi(v_1)}$ (either actual or the identity map) as follows. For every tuple $\mathbf{t} \in R_{\chi(v_1)}$ and for all tuples $\mathbf{t}' \in m_{v_i,v_1} : \pi_{\chi(v_i) \cap \chi(v_1)}(\mathbf{t}') = \pi_{\chi(v_i) \cap \chi(v_1)}(\mathbf{t})$, the tuple $\mathbf{t}'' = (\mathbf{t}'', f(\mathbf{t}) \cdot f(\mathbf{t}'))$ is appended to $R'_{v_i}$.

**Algorithm 3** Algorithm for $\mathcal{T}$ is a Star

1: **Input:** A star $P = (v_1, \ldots, v_k) \in \mathcal{T}$ and functions $\{R_{\chi(v_i)} : i \in [k]\}$. Note that $v_1$ is the center and the others are leaves.
2: **Output:** $R'_P$
3: The player containing $R_{\chi(v_1)}$ broadcasts it to all players in $G$.
4: For every $i \in [2, k]$, the player containing $R_{\chi(v_i)}$ internally computes the the Up Message from $v_i$ to $v_1$, $m_{v_i,v_1} = \bigoplus_{z_1 \in \text{Dom}(z_1)}^{(z_1)} \ldots \bigoplus_{z_w \in \text{Dom}(z_w)}^{(z_w)} R_{\chi(v_i)}$, where $\Gamma(v_{i,1}) = \chi(v_i) \setminus \chi(v_1) = (z_1, \ldots, z_w) \subseteq (x_{\ell+1}, \ldots, x_n)$. ▷ All the $\oplus^{(z_j)}$s for every $j \in [m]$ are computed on the values annotated with the tuples in the function.
5: **if** $i = 2$ **then**
6: The player containing $R_{\chi(v_2)}$ computes $R'_{v_2} = R_{\chi(v_1)} \cdot m_{v_2,v_1}$ internally. ▷ This product is computed on the annotated values on the function and the message.
7: **else**
8: Converts $R_{\chi(v_1)}$ to an identity map i.e., all entries in it are assigned a value of 1.
9: The player containing $R_{\chi(v_i)}$ computes $R'_{v_i} = R_{\chi(v_1)} \cdot m_{v_i,v_1}$ internally. ▷ This product is computed on the annotated values on the function and the message.
10: $R'_P = \prod_{i=2}^{k} R'_{v_i}$ ▷ This product is computed on a Steiner Tree packing like Theorem 3.11.
11: **return** $R'_P$

Since all the $R'_{v_i}$s are computed on the same attribute set $\chi(v_i)$ and the annotated tuples in each $R'_{v_i}$ can be multiplied in constant time, then Step 10 of our Algorithm can be computed on Steiner Tree, resulting in an upper bound of

$$O\left( \min_{\Delta \in [|V|]} \left( \frac{N \cdot r}{\text{ST}(G,K,\Delta)} + \Delta \right) \right).$$

We can now repeat the same arguments from the proof of Lemma G.4 (as stated earlier) until the root of $\mathcal{T}$, which gives us the first term in the required upper bound. We can then apply the *naive* protocol on the root of $\mathcal{T}$ (that contains $\text{C}(\mathcal{H})$), solving

it in $\tau_{\text{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N)$ rounds (using Lemma 3.13). Since $\mathcal{F} \subseteq V(\text{C}(\mathcal{H}))$, we do not require a downward pass in our message-passing algorithm. In total, we have the desired upper bound for computing $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$. Note that our choices of $f$ and operator sequence $(\oplus^{(i)})_{\ell<i\leq n}$ were arbitrary and thus, our results hold for general $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$. Finally, since our choice of a GHD was arbitrary, we have $y(\mathcal{T}) \leq y(\mathcal{H})$. This completes the proof.

## H.4 Lower Bound for General FAQs

The lower bound follows from the fact that our hard $\text{BCQ}_{\mathcal{H},N}$ instance for a $d$-degenerate $\mathcal{H}$ is a hard $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ instance for the operator set $\left(\oplus^{(\ell+1)}, \oplus^{(n)}\right)$ with $\oplus^{(i)} = \otimes$ or $(\mathbb{D}, \oplus^{(i)}, \otimes)$ forms a commutative semiring with the same additive identity $\mathbf{0}$ and multiplicative identity $\mathbf{1}$ for every $\ell < i \leq n$.

We argue $\mathcal{R}\left(\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K\right) \geq \mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K)$ and the above result follows. We start with the $\text{BCQ}_{\mathcal{H},N}$ instance from Section G.3.4. We construct a $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ instance from a given $\text{BCQ}_{\mathcal{H},N}$ instance as follows. For each function $R_e : e \in E(\mathcal{H})$, we apply the following function $f$ on every tuple $\mathbf{t} \in \prod_{u\in v(e)} \text{Dom}(u)$: we set $f(\mathbf{t}) = 1$ if $\mathbf{t} \in R_e$ and 0 otherwise. Note that this implies we can define functions of the form $R_e = \{(\mathbf{t}, 1)) : \mathbf{t} \in R_{v(e)}\}$ for every $e \in E(\mathcal{H})$. We now have a $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ instance of the form

$$\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}} = \bigoplus_{x_{\ell+1}\in\text{Dom}(x_{\ell+1})}^{(\ell+1)} \cdots \bigoplus_{x_n\in\text{Dom}(x_n)}^{(n)} \otimes \phi_S(\mathbf{x}_S). \tag{24}$$

Given this setup, we claim that $\text{BCQ}_{\mathcal{H},N}$ is 1 iff $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ is 1 and 0 otherwise. To see why this is true, notice that in all our hard instances of $\text{BCQ}_{\mathcal{H},N}$, the corresponding join output $|\bowtie_{e\in E(\mathcal{H})} R_e| \leq 1$ (from Remark G.13). As a result, we can apply the sequence of operators $(\oplus^{(i)})_{n\leq i<\ell}$ one-by-one from right to left. If $\oplus^{(i)} = \otimes$ for $n \leq i < \ell$ , applying it on at most one value does not make any difference. Otherwise, since all commutative semirings of the form $(\mathbb{D}, \oplus^{(i)}, \otimes)$ have the same additive identity $\mathbf{0}$ and multiplicative identity $\mathbf{1}$, we can conclude that $\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}} = 1$ iff $\text{BCQ}_{\mathcal{H},N} = 1$. Note that the choices of operators $(\oplus^{(i)})_{\ell<i\leq n}$ and $\mathbb{D}$ was arbitrary. Thus, we have $\mathcal{R}\left(\text{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K\right) \geq \mathcal{R}(\text{BCQ}_{\mathcal{H},N}, G, K)$.

## H.5 Restriction on Choice of $\mathcal{F}$

Recall that any (hyper)graph $\mathcal{H}$ can be decomposed into a core $\text{C}(\mathcal{H})$ and a forest $\text{F}(\mathcal{H})$ using Construction 2.9. We would like to mention here that our upper and lower bounds in this paper hold only for the case when $\mathcal{F} \subseteq V(\text{C}(\mathcal{H}))$. For our upper bounds, we believe that this is due to the fact that we apply different algorithms on $\text{F}(\mathcal{H})$ and $\text{C}(\mathcal{H})$. For the lower bounds, we once again deal with $\text{C}(\mathcal{H})$ and $\text{F}(\mathcal{H})$ independently and sum the bounds obtain from either of them. We believe that expanding the choices of $\mathcal{F}$ needs new techniques for both the upper and lower bounds.

## H.6 Hash-based Split of Relations

In this section, we address the assumption that the input functions in $\mathcal{H}$ are completely assigned to players in $G$. We prove upper and lower bounds when the input relations are split based on certain kind of hashes. As a by-product, our lower bounds techniques help us prove bounds when input functions are not split but randomly assigned to players in $G$ (overcoming the assumption of worst-case assignment of functions to players in $G$).

We define our setup in detail here.

*H.6.1 Our Setup.* We first state the condition we need on a hashes used to split relations that is sufficient for our bounds. Then, we state some realistic scenarios where these conditions are satisfies.

Definition H.4. *Given a hypergraph $\mathcal{H}$ and GHD $\mathcal{T}$ such that the root of $\mathcal{T}$ is $\text{C}(\mathcal{H})$, we say a family of hash functions $\widetilde{H} = \{h_e : \prod_{v\in e} \text{Dom}(v) \to K | e \in \overline{\mathcal{E}}\}$ is* consistent *with $\mathcal{T}$ and $K$ if the following holds. Let $r'$ be the root of $\mathcal{T}$. If $e \subseteq \chi(r')$ (i.e. $e$ is assigned to the root of $\mathcal{T}$), then $h_e$ can be arbitrary. Now consider a non-root node $v$ in $\mathcal{T}$ and let $u$ be its parent.*

*First, we consider the projection $S_{u,v} = \pi_{\chi(u)\cap\chi(v)} R_{\chi(v)}$. Then, for every tuple $\mathbf{s} \in S_{u,v}$, we have that $h_{\chi(v)}(\mathbf{t})$ is the same for* every $\{\mathbf{t} \in R_{\chi(v)} : \pi_{\chi(v_2)\cap\chi(v)}(\mathbf{u}) = \mathbf{s}\}$.

*Further, we say the set of relations $\{R_e\}_{e\in\overline{\mathcal{E}}}$ are* split according to $\widetilde{H}$, *if for any $e \in \overline{\mathcal{E}}$ and $\mathbf{t} \in R_e$, the tuple $\mathbf{t}$ is assigned to player $h_e(\mathbf{t})$.*

It turns out that if tuples in the relations are split according to a family of hash functions as in the above definition, then we can generalize Algorithm 3 to this case.

Next we observe that our condition on a family of hash functions being consistent with a GHD $\mathcal{T}$ and $K$ is reasonable. In other words, we are assuming that all attributes of $R_e$ for every $e \in \overline{\mathcal{E}}$ are stored in a global variable elimination order that is compatible with $\mathcal{T}$. In particular, this implies for any non-root $v$ in $\mathcal{T}$ and its parent $u$, we have that $\chi(u) \cap \chi(v)$ is a prefix of $\chi(v)$ according to this variable elimination order. This assumption on the variables in $R_e$ being stored in the variable elimination order of $\mathcal{T}$ has

been made before for GHD-based algorithms used to solve FAQ [38, 39]. Further, "bit-map based" [52] hash functions $h_e$ do indeed satisfy the consistency property in Definition H.4.

We note that if the relations themselves are free of skew (which is an assumption made in [9]), then a consistent family of hash functions will also distribute the tuples in a relation (near) equally among players in $K$.

### H.6.2 Main Theorem.

Theorem H.5. *For arbitrary $G$, subset of players $K$, any $\mathcal{F} \subseteq V(\mathrm{C}(\mathcal{H}))$ and $d$-degenerate hypergraphs $\mathcal{H}$ with arity at most $r$. Further, assume that the set of relations are split according to hash family $\widetilde{H}$ that is consistent with $\mathcal{T}$ (where $y(\mathcal{T}) = y(\mathcal{H})$) and $K$. Then, we have*

$$\mathcal{D}(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K) = O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N \cdot r}{\mathrm{ST}(G,K,\Delta)} + |K| \cdot \Delta\right) + \tau_{\mathrm{MCF}}\left(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N\right)\right). \tag{25}$$

*Further, if $\widetilde{H}$ is is a random hash family where $h_e$ for $e \in \overline{\mathcal{E}}$ are chosen independently and uniformly (conditioned on $\widetilde{H}$ being consistent). we have (with high probability over the randomness in $\widetilde{H}$):*

$$\mathcal{R}(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K) \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot N}{r \cdot \gamma(G,K)} + \frac{n_2(\mathcal{H}) \cdot N}{d \cdot r \cdot \gamma(G,K)}\right), \tag{26}$$

*where $\gamma(G,K)$ is the minimum over all cuts $(A,B)$ separating $K$ of the quantity $\frac{|E(A,B)||K|^2}{(\min(|A|,|B|))^2}$, where $E(A,B)$ is the set of edges crossing the cut. Here, $y(\mathcal{H})$ and $n_2(\mathcal{H})$ are defined as in Definition 3.1.*

We note that when $G$ is a line $\gamma(G,K)$ is attained at the cut that equally cuts $K$ into two parts and since $|E(A,B)| = 1$, we get that $\gamma(G,K) = O(1)$.

We would like to note here that for simple graphs $\mathcal{H}$, we can overcome the factor of $d$ in the lower bound (see Theorem 4.8). In particular, we can use similar ideas from the Proof of Theorem 4.8 for upper bounds to obtain the following corollaries.

Corollary H.6. *For arbitrary $G$, subset of players $K$ and any star $\mathcal{H}$. Further, assume that the set of relations are split according to hash family $\widetilde{H}$ that is consistent with $\mathcal{T}$ (where $y(\mathcal{T}) = y(\mathcal{H})$) and $K$. Then, we have*

$$\mathcal{D}(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K) = O\left(\min_{\Delta \in [|V|]} \left(\frac{N \cdot r}{\mathrm{ST}(G,K,\Delta)} + |K| \cdot \Delta\right)\right).$$

Corollary H.7. *For arbitrary $G$, subset of players $K$ and any forest $\mathcal{H}$. Further, assume that the set of relations are split according to hash family $\widetilde{H}$ that is consistent with $\mathcal{T}$ (where $y(\mathcal{T}) = y(\mathcal{H})$) and $K$. Then, we have*

$$\mathcal{D}(\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}, G, K) = O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left(\frac{N \cdot r}{\mathrm{ST}(G,K,\Delta)} + |K| \cdot \Delta\right)\right).$$

### H.6.3 Upper Bound.

The upper bounds follows from a slight modification of our algorithm to compute $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ when relations are not partitioned (Section H.3).

We present a proof sketch here. The idea is very similar to the proof in Section H.3. To that end, we modify Algorithm 3 as follows. Instead of broadcasting $R_{\chi(v_1)}$, the Steps 4 to 10 are applied individually on each tuple in $R_{\chi(v_1)}$ as follows.

Let us first start with the case when $\mathcal{T}$ is a star with $v_1$ as the center and $v_2, \ldots, v_{|P|}$ as the leaves. Let's fix a tuple $\mathbf{t} \in R_{\chi(v_1)}$. It is broadcast to all players in $K$ along with a counter $c_{\mathbf{t}} \in [0, |K|]$.[23] Initially, we set $c_{\mathbf{t}} = 0$. For any player $\ell \in K$ and $j \in [2, |P|]$, define $R^{(\ell)}_{\chi(v_j)}$ to be the set of tuples in $R_{\chi(v_j)}$ mapped to $i$ by $h_{\chi(v_j)}$. Upon receiving $(\mathbf{t}, c_{\mathbf{t}})$, player $\ell$ checks if there exists a tuple $\mathbf{t}' \in R^{(\ell)}_{\chi(v_j)}$ such that $\pi_{\chi(v_j)\cap\chi(v_1)}\mathbf{t}' = \pi_{\chi(v_j)\cap\chi(v_1)}(\mathbf{t})$. If so, then Player $i$ increments $c$ by one, (internally) computes the sum

$$\sum_{\mathbf{t}' \in R^{\ell}_{vj} : \pi_{\chi(v_j)\cap\chi(v_1)}\mathbf{t}' = \pi_{\chi(v_j)\cap\chi(v_1)}(\mathbf{t})} f_{\chi(v_j)}(\mathbf{t}')$$

so that it can contribute to the running product:

$$v(\mathbf{t}) = f_{\chi(v_1)}(\mathbf{t}) \cdot \prod_{j=2}^{|P|} \left( \sum_{\mathbf{t}' \in R^{\ell}_{vj} : \pi_{\chi(v_j)\cap\chi(v_1)}\mathbf{t}' = \pi_{\chi(v_j)\cap\chi(v_1)}(\mathbf{t})} f_{\chi(v_j)}(\mathbf{t}') \right),$$

which is the value corresponding to $\mathbf{t}$ for the corresponding FAQ-SS query. Note that the sums and products are computed on the values corresponding to the tuples as in Section H.3. At the end of the procedure, if $c = |K|$, then $(\mathbf{t}, v(\mathbf{t}))$ is added to the result $R'_P$ and we continue. Note that we can repeat the above procedure for each star in $\mathcal{T}$ until we reach the root as we did in Section H.3.

[23] This can be done via a Steiner tree packing with $\min_{\Delta \in [|V|]} \left(\frac{N \cdot r}{\mathrm{ST}(G,K,\Delta)} + \Delta\right)$ rounds.

The correctness of the above procedure follows from our setup defined in Section H.6.1. Further, the sums can be computed internally by each player and the products can be computed on a Steiner Tree packing as in Theorem 3.11,[24] resulting in an upper bound of

$$O\left(y(\mathcal{H}) \cdot \min_{\Delta \in [|V|]} \left( \frac{N \cdot (r + \log(|K|))}{\mathrm{ST}(G, K, \Delta)} + |K| \cdot \Delta \right)\right),$$

where the $\log(|K|)$ additive term is to keep track of the counter $c_{\mathbf{t}}$ (and in the final bound is absorbed into the $O(\cdot)$).

To complete the proof, we use the following *trivial protocol* on the root of $\mathcal{T}$. In particular, any one designated player should still receive all the partitions from all relations. Since each player has $|K|$ partitions of all the remaining relations, the round complexity is given by $\tau_{\mathrm{MCF}}(G, K, n_2(\mathcal{H}) \cdot d \cdot r \cdot N)$ (using Lemma 3.13).

*H.6.4 Lower Bound.* The lower bound follows similarly from ideas in Section H.4.

We have already shown in Section H.4 that our hard $\mathrm{BCQ}_{\mathcal{H},N}$ for a $d$-degenerate $\mathcal{H}$ is a hard instance for $\mathsf{FAQ}_{\mathbb{D},\mathcal{H},N,\mathcal{F}}$ as well. The only difference is that we cannot apply lower bounds on worst-case assignment directly anymore since the relations are partitioned now. We address the issue here.

Similar to the proof of Lemma G.10, we consider an arbitrary cut $(A, B)$ of $G$ that separates $K$, where $A$ and $B$ denote the set of vertices in each partition ($A \cup B = V(G)$). For simplicity, we assume $|A| = |B|$ and later show how to get around this restriction. Using notation given in the proof of Lemma G.4, let $q_{\mathcal{H},\hat{S},\hat{T}}$ be the query corresponding to a given instance $\mathrm{TRIBES}_{\frac{y(\mathcal{H})}{r},N}(\hat{S}, \hat{T})$. Note that we partition all relations, which includes $\{R_{S_i}\}_{p_i \in I}$ and $\{R_{T_i}\}_{p_i \in I}$. Since all the set pairs $(R_{S_i}, R_{T_i})$ for every $p_i \in I$ in the TRIBES instance are independent, we start by considering one such pair. In particular, let's consider $R_{S_i}$. We first note that the way we have defined $R_{S_i}$ every prefix has exactly *one* extension. Since $\widetilde{H}$ is chosen so that the individual hash functions are independent and uniformly distributed (conditioned on $\widetilde{H}$ being consistent), in this particular case because of the afore-mentioned property of the prefixes, each hash function is a uniformly random hash function. Thus, any tuple in $R_{S_i}$ is uniformly distributed among the players in $K$.

We now see how the tuples in $R_{S_i}$ and $R_{T_i}$ are split. In particular, since any tuple in $R_{S_i}$ (or $T_{S_i}$) is assigned uniformly to players in $K$. In particular, each tuple $\mathbf{t} \in R_{S_i}$ is assigned to either $A$ or $B$ with probability $\frac{1}{2}$ (since $|A| = |B|$). Likewise for tuples in $R_{T_i}$ (since the hash functions for different relations are in independent). More formally, we write "$\mathbf{t} \in A$" if a given tuple $\mathbf{t}$ is assigned to vertices in $A$ (similarly for vertices in $B$). We now have

$$\Pr[\mathbf{t} \in A] = \Pr[\mathbf{t} \in B] = \Pr[\mathbf{t}' \in A] = \Pr[\mathbf{t}' \in B] = \frac{1}{2},$$

where $\mathbf{t} \in R_{S_i}$ and $\mathbf{t}' \in R_{T_i}$. Note that this implies

$$\Pr[(\mathbf{t} \in A) \wedge (\mathbf{t}' \in B)] = \Pr[(\mathbf{t}' \in A) \wedge (\mathbf{t} \in B)] = \frac{1}{4}.$$

Thus, in expectation the total number of tuples that satisfy the above property in $R_{S_o}$ and $R_{T_o}$ is $\frac{N}{4}$. Moreover, this number is at least $\frac{N}{8}$ with probability $1 - 2^{-\Omega(N)}$. Abusing notation, let $R_{S_i}$ and $R_{T_i}$ denote the tuples of the original $R_{S_i}$ and $R_{T_i}$ that were split between $A$ and $B$, and assume that these relations are exactly of size $\frac{N}{8}$.

In addition to the above, following on Remark G.11, the distribution $\hat{\mathbb{D}}$ on $(\hat{S}, \hat{T})$ we use in a black-box manner has a property that $|S_i \cap T_i| \leq 1$, i.e., there is at most only one value $a' \in p_i$ such that $S_i \cap T_i = a'$ (follows from Remark G.12). Thus, we need this particular tuple to always be split (i.e. one copy goes to $A$ and other goes to $B$). Otherwise, we can only pass that value resulting in a protocol with constant number of rounds. We can ensure this by conditioning our expectation on the event that the tuple containing $a'$ in both $R_{S_i}$ and $R_{T_i}$ is always split. Further, it is easy to see that (1) $a'$ is split with probability at least $\frac{1}{4}$ and (2) Conditioned on being split, $a'$ is still uniformly distributed over the $\frac{N}{8}$ locations in $R_{S_i}$ and $R_{T_i}$. By another application of the Chernoff bound, for at least $\frac{1}{8}$ of the $(S_i, T_i)$ pairs this special value $a'$ is split. In other words, we now have a smaller TRIBES instance across the $(A, B)$ cut, where we have $\frac{1}{8}$th the number of set Disjointness instances (let these be indexed by $I'$ with $|I'| = \frac{|I|}{8}$) where each set disjointness instance is $\frac{1}{8}$th the original size. The other relations in $q_{\mathcal{H},\hat{S},\hat{T}}$ can be partitioned randomly and assigned arbitrarily across $\mathrm{MinCut}(G, K)$.

We now consider the induced TRIBES function based on $I'$. Note that we have argued that with high probability, we have a set $I'$ of size $\Omega\left(\frac{y(\mathcal{H})}{r}\right)$. In particular, we have argued that the relations $\{R_{S_i}\}_{p_i \in I'}$ are assigned to vertices in $A$ and $\{R_{T_i}\}_{p_i \in I'}$ to B. In particular, Alice gets the $\frac{N}{8}$ tuples in the sets $\{S_i\}_{p_i \in I'}$ (corresponding to $R_{S_i}$) assigned to $A$ and Bob gets the $\frac{N}{8}$ tuples in the sets $\{T_i\}_{p_i \in I'}$ (corresponding to $R_{T_i}$) assigned to vertices in $B$ (ignoring the additional relations). It is not too hard to see that if there exists a $z$ round protocol on $G$, then there is an $O(z \cdot |E(A, B)|)$ two-party protocol (see proof of Lemma 4.7 for a detailed discussion). Since $z \cdot |E(A, B)|$ is lower bounded from Theorem 2.3 by $\widetilde{\Omega}\left(\frac{y(\mathcal{T}) \cdot N}{r}\right)$, we have a lower bound of $\widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot N}{r \cdot |E(A,B)|}\right)$.

[24] The algorithm for Theorem 3.11 first thinks of its input as a vector and computes its component-wise AND. These vectors are sub-divided among the edge disjoint Steiner trees and then the component-wise AND of the smaller vectors is done in a bottom up fashion in a dedicated Steiner tree from the packing. In the current case we want to compute component-wise product and we can just run the set intersection algorithms where instead of computing component-wise AND we use component-wise product.

Finally, we remove the restriction that $|A| = |B|$. More generally, instead of a uniform probability of $\frac{1}{4}$ of two tuples in $S_i$ and $T_i$ being split, we would have a probability of $\frac{\min(|A|,|B|)\cdot\max(|A|,|B|)}{|K|^2} \geq \frac{\min(|A|,|B|)}{2\cdot|K|}$. Generalizing the above argument where we replace the $\frac{1}{4}$ with the above probability, we get that if $z$ is the round complexity of a protocol to compute $q_{\mathcal{H},\hat{S},\hat{T}}$, then we have

$$z \cdot |E(A,B)| \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H}) \cdot (\min(|A|,|B|))^2}{r \cdot |K|^2} \cdot N\right).$$

Our definition of $\gamma(G,K)$ implies that we have $z \geq \widetilde{\Omega}\left(\frac{y(\mathcal{H})\cdot N}{r\cdot\gamma(G,K)}\right)$. Similar arguments can be applied to other TRIBES instances, completing the proof.

We would like to state here the similar ideas in the proof above can be used to obtain a lower bound of the form $\widetilde{\Omega}\left(\frac{y(\mathcal{H})\cdot N}{r\cdot\gamma(G,K)}\right)$ for the case when the relations $\{R_{S_i}\}_{p_i\in I}$ and $\{R_{T_i}\}_{p_i\in I}$ are not split but only randomly assigned to players in $K$ (instead of a worst-case assignment). In particular, this removes the assumption of worst-case assignment of functions in $\mathcal{H}$ to players in $G$.

## H.7 Assumptions on $G$ and $\mathcal{H}$

We note that if $N$ is much larger than size of $G$, then all players can send their information (either about $\mathcal{H}$ or their locality in $G$) to one player who can then broadcast the common knowledge back to all players. Unfortunately, for smaller values of $N$, the state-of-the-art results in the CONGEST model do not give tight bounds for Steiner tree packing and multi-commodity flow for arbitrary $G$ as we need [26]. We consider our work to provide further motivation to solve these two fundamental problems in the CONGEST model.

# I PROOF OF PROPOSITION 6.1

We describe our algorithm here. We start by computing $\mathbf{y}_1 = A_1 \cdot \mathbf{x}$, which can be done in $O(N^2)$ rounds. We then successively compute $\mathbf{y}_i = A_i \cdot \mathbf{y}_{i-1}$ for every $i \in [2,k]$. Note that this takes $O(k \cdot N^2)$ rounds in total and we would get the final answer in $\mathbf{y}_k$.

# J DIFFERENCE FROM ONLINE MATRIX VECTOR MULTIPLICATION

We recall the definition of Online Matrix Vector Multiplication (OuMv) here [35]. Given an $N \times N$ Boolean matrix $M$, we receive $N$ Boolean $N \times 1$ vectors $\mathbf{v_1} \ldots, \mathbf{v_n}$ one at a time, and are required to output $M \cdot \mathbf{v_i}$ (over the Boolean semiring ($\mathbb{D} = \{0,1\}, \vee, \wedge$)) before seeing the vector $\mathbf{v_{i+1}}$, for all $i \in [n-1]$.
On the other hand, the Matrix Chain-Vector Multiplication (MCM) we consider in this paper is as follows. Given $k$ matrices $A_i \in \mathbb{F}_2^{N\times N}$ for every $i \in [k]$ and one vector $\mathbf{x} \in \mathbb{F}_2^N$, our goal is to compute $\mathbf{A}_k \cdot \mathbf{A}_{k-1} \cdot \ldots \cdot \mathbf{A}_1 \cdot \mathbf{x}$ over $\mathbb{F}_2$.
These are in some sense dual problems and our results do not imply anything for OuMv.

# K MIN-ENTROPY OF MATRIX-VECTOR MULTIPLICATION

We now prove Theorem 6.3.

## K.1 Preliminaries

*K.1.1 Min-entropy.* To be consistent with usual terminology used in the pseudorandomness literature, we will use the following equivalent definition as (5):

$$H_\infty^\epsilon(X) = \sup_{X' \sim_\epsilon X} H_\infty(X'),$$

where $X' \sim_\epsilon X$ is overall all distributions $X'$ that have statistical distance at most $\epsilon$ from $X$. Notice that in the expression, we *do not* require $\mathrm{supp}(X') \subseteq \mathrm{supp}(X)$, neither do we restrict the domain of $X'$. The equivalence of the two definitions is easy to see as in the latter definition, we can form the distribution $X'$ by moving $\epsilon$ probability from $X$ and distribute it evenly on sufficiently many newly introduced elements outside $\mathrm{supp}(X)$.

The following result will be useful:

Proposition K.1. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two distributions. Let $f$ be a deterministic function on the* $\mathrm{supp}(\mathcal{D}_1) \cup \mathrm{supp}(\mathcal{D}_2)$. *If $\mathcal{D}_1 \sim_\epsilon \mathcal{D}_2$, then*

$$f(\mathcal{D}_1) \sim_\epsilon f(\mathcal{D}_2).$$

For any event $\overline{\mathcal{E}}$, we will use $\mathbf{1}_{\overline{\mathcal{E}}}$ to denote the 0/1-indicator variable for $\overline{\mathcal{E}}$. We will use $\mathcal{U}_m$ to denote the uniform distribution on $\mathbb{F}_2^m$, where $\mathbb{F}_2$ is the finite field of two elements.

*K.1.2 Matrices and Vectors.* We will deal with vectors $\mathbf{x} \in \mathbb{F}_2^n$ in this section as well as matrices $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ for $m \leq n$.[25] All vectors by default are column vectors and all indices start from 1. We will use $\mathbf{A}_i$ to denote the $i$th row of $\mathbf{A}$ and for any subset $S \subseteq [m] \stackrel{\text{def}}{=} \{1, \ldots, m\}$, $\mathbf{A}_S$ denotes the submatrix indexed by the rows of $\mathbf{A}$ indexed by $S$.

Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, we will use $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote their inner product over $\mathbb{F}_2$.

## K.2 Proof of Theorem 6.3

We will argue the general version of Theorem 6.3 (note Theorem 6.3 follows from the result below for $n = m = N$):

Theorem K.2. *Let the constant $\gamma$ be small enough. Let $\mathbf{x} \in \mathbb{F}_2^n$ and $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ (for $m \leq n$) be distributed such that there exists a random variable $\mathbf{Y}$ such that for every $y \in supp(\mathbf{Y})$, conditioned on $\mathbf{Y} = y$, $\mathbf{x}$ and $\mathbf{A}$ are independent. Further, if for some reals $\epsilon_1, \epsilon_2 \geq 0$,*

$$H_\infty^{\epsilon_1}(\mathbf{A}|\mathbf{Y}) \geq (1-\gamma)mn,$$

*and*

$$H_\infty^{\epsilon_2}(\mathbf{x}|\mathbf{Y}) \geq \alpha \cdot n,$$

*where*

$$\alpha \stackrel{def}{=} 3\gamma + \sqrt{2\gamma} + h(\sqrt{2\gamma}).$$

*Then,*

$$H_\infty^{\epsilon_1+\epsilon_2+2^{-\Omega(\gamma m)}}(\mathbf{A}\mathbf{x}|\mathbf{Y}) \geq \left(1 - \sqrt{2\gamma}\right) \cdot m.$$

In the rest of the section we will argue Theorem K.2.

(1) First, we prove the theorem for the case where $\epsilon_1 = \epsilon_2 = 0$ and $\mathbf{Y}$ is deterministic. This is done as follows.
  (1a) We will argue that $\mathbf{A}$ has high enough min-entropy in "most" rows. This will define what is called a "block-source." The details are in Section K.3.
  (1b) We then argue that any $\mathbf{A}$ that is a sufficiently good block source, has the following property: the inner product $\langle \mathbf{A}_i, \mathbf{x} \rangle$ is close to a random bit as long as $\mathbf{x}$ has min-entropy at least $\alpha \cdot n$. Further, we can make this argument for each row with high enough min-entropy by only adding up the "closeness" for each such row. The details are in Section K.4.
(2) Then, in Section K.5, we remove the assumption that $\epsilon_1 = \epsilon_2 = 0$ and $\mathbf{Y}$ is deterministic.

## K.3 A is a good enough block source

We begin with the definition of a block-source[26]

Definition K.3. *A random variable $\mathbf{A}'$ over $\mathbb{F}_2^{m \times n}$ is an $(\eta, n')$-block source for some $\eta \in [0,1]$ and $n' \leq n$ if there exists a subset*

$$S \subseteq [m] \text{ with } |S| \leq \eta m$$

*such that for every $A \in supp(\mathbf{A}')$ and every $i \notin S$, we have*

$$H_\infty\left(\mathbf{A}'_i | \mathbf{A}'_{[i-1]} = A_{[i-1]}\right) \geq n'.$$

We remark that in the above definition we do condition on all rows in $[i-1]$ (and not just $[i-1] \setminus S$).

Ideally, we would like to argue that our $\mathbf{A}$ is a $(\gamma, (1-\gamma)n)$-block source. We will instead argue something a bit weaker, which is nonetheless powerful enough to help us prove Theorem K.2. In particular, we will argue that for a certain notion of "badness," (1) There are very few bad matrices (Section K.3.1) and (2) matrices that are not bad are indeed good block sources (Section K.3.2).

*K.3.1 Bad matrices.* Before we proceed, we will need couple of other definitions:

Definition K.4. *For every $A \in supp(\mathbf{A})$ and $i \in [m]$, define*

$$p_i(A) \stackrel{def}{=} \Pr\left[\mathbf{A}_i = A_i \middle| \mathbf{A}_{[i-1]} = A_{[i-1]}\right], \qquad \text{and} \qquad q_i(A) \stackrel{def}{=} -\log_2(p_i(A)).$$

Definition K.5. *For any $\tau > 0$, we refer to $A \in supp(\mathbf{A})$ as $\tau$-rare if there exists an $i \in [m]$ such that*

$$p_i(A) < 2^{-n(1+\tau)} \left(\text{or equivalently, } q_i(A) > n(1+\tau)\right).$$

The next lemma justifies the naming above:

Lemma K.6. $\Pr[\mathbf{A} \text{ is } \tau\text{-rare}] < m \cdot 2^{-\tau \cdot n}$.

[25] We are changing notation only for this section of the appendix. We have used $n$ to denote the number of variables in a query but in this section we will use it to define the dimension of vector and matrices as is the norm in linear algebra.

[26] This is a more specific definition than the usual definition. We go with the more specific definition since it suffices for our purposes.

Proof. Call a matrix $A \in \text{supp}(\mathbf{A})$ to be *$\tau$-rare at $i \in [m]$* if $p_i(A) < 2^{-n(1+\tau)}$– denote this event by $\overline{\mathcal{E}}(i, A)$. We next show that

$$\Pr[\mathbf{A} \text{ is } \tau\text{-rare at } i] < 2^{-\tau \cdot n},$$

which would complete the proof via the union bound. In the rest of the proof we prove the above bound.

Indeed, consider the following sequence of relations:

$$\begin{aligned}
\Pr[\mathbf{A} \text{ is } \tau\text{-rare at } i] &= \sum_{A \in \mathbb{F}_2^{m \times n}} \Pr[\mathbf{A} = A] \cdot \mathbf{1}_{\overline{\mathcal{E}}(i,A)} \\
&= \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{A_i \in \mathbb{F}_2^n} \sum_{A_{[i+1:m]} \in \mathbb{F}_2^{(m-i)\times n}} \Pr\left[\mathbf{A} = A | \mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \mathbf{1}_{\overline{\mathcal{E}}(i,A)} \qquad (27) \\
&= \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{A_i \in \mathbb{F}_2^n} \Pr\left[\mathbf{A}_i = A_i | \mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \mathbf{1}_{\overline{\mathcal{E}}(i,A)} \qquad (28) \\
&\le \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{A_i \in \mathbb{F}_2^n} 2^{-n(1+\tau)} \qquad (29) \\
&\le 2^{-\tau \cdot n}, \qquad (30)
\end{aligned}$$

as desired. In the above, (27) follows from definition of conditional probability. (28) follows from the fact that $\mathbf{1}_{\overline{\mathcal{E}}(i,A)}$ is the same for all matrices that agree in $A_{[i]}$ while (29) follows from the definition of the event $\overline{\mathcal{E}}(i, A)$. Finally, (30) follows from the fact that there are $2^n$ possibilities for $A_i$. □

Next, we argue that every matrix $A$ that is not $\tau$-rare has few rows $i \in [m]$ for which $q_i(A)$ is small. It is crucial to note the difference between the situation here and what we need from block sources (in Definition K.3). In Definition K.3, the set of "bad" rows is the same for all matrices in the support of the distribution. On the other hand, in the lemma below, we show that for every matrix $A$, there exists a set of "bad rows." (So ultimately we want to flip the order of quantifiers.)

Lemma K.7. *Let $A \in \text{supp}(\mathbf{A})$ be such that it is not $\tau$-rare. Then there exists a subset $S \subseteq [m]$ of size*

$$|S| \le \sqrt{\tau + \gamma} \cdot m$$

*such that for every $i \notin S$, we have*

$$q_i(A) \ge \left(1 - \sqrt{\tau + \gamma}\right) \cdot n.$$

Proof. Define for every $i \in [m]$,

$$q_i'(A) = n(1 + \tau) - q_i(A).$$

Note that since $H_\infty(\mathbf{A}) \ge (1 - \gamma)mn$, we have

$$\sum_{i=1}^{m} q_i(A) \ge (1 - \gamma)mn.$$

Since $A$ is not $\tau$-rare, this in turn implies that

$$q_i'(A) \ge 0, \forall i \in [m], \qquad \text{and} \qquad \sum_{i=1}^{m} q_i'(A) \le (\tau + \gamma)mn.$$

Thus, by a Markov argument we have that the fraction of rows $i \in [m]$ for which we have $q_i'(A) \ge \sqrt{\tau + \gamma} \cdot n$ is at most $\sqrt{\tau + \gamma}$. Let this set of rows be $S$. Then note that we have for every $i \notin S$,

$$q_i(A) = (1 + \tau)n - q_i'(A) \ge n(1 + \tau - \sqrt{\tau + \gamma}) \ge n(1 - \sqrt{\tau + \gamma}),$$

as desired. □

Before we proceed for notational convenience, define

$$\eta \stackrel{\text{def}}{=} \sqrt{\tau + \gamma}.$$

We are now ready for our final set of definitions:

Definition K.8. *For every $A \in \text{supp}(\mathbf{A})$ that is not $\tau$-rare, define $B(A)$ to be a subset $S \subseteq [m]$ such that*

*(1) $|S| \le \eta \cdot m$; and*

*(2) For every $i \notin S$, $q_i(A) \ge (1 - \eta)n$.*

*If $A$ is $\tau$-rare, then we set $B(A) = \perp$.*

Note that Lemma K.7 shows that the function $B(A)$ is well defined for $A \in \text{supp}(\mathbf{A})$ which is not $\tau$-rare. For other $A$'s, $B(A)$ is defined to be the "exception" symbol $\perp$.

Definition K.9. *Let $\tau, \delta > 0$. We call $A \in supp(\mathbf{A})$ to be $(\tau, \delta)$-*bad *if*

*(1) $A$ is $\tau$-rare; or*

*(2) There exists an $i \in [m]$ such that*

$$\Pr\left[B(\mathbf{A}) = B(A) \,|\mathbf{A}_{[i-1]} = A_{[i-1]}\right] < \delta. \tag{31}$$

We will argue in Section K.3.2 that $\mathbf{A}$ conditioned on $B(\mathbf{A})$ leads to a block source. But first we argue (using arguments similar to those used in the proof of Lemma K.6) that the total probability mass on bad matrices is small.

Lemma K.10. *For every $\tau, \delta > 0$,*

$$\epsilon_{bad} \stackrel{def}{=} \Pr[\mathbf{A} \text{ is } (\tau, \delta)\text{-bad}] \leq m \cdot 2^{-\tau n} + m \cdot \delta \cdot 2^{h(\eta)m}.$$

Proof. Call a matrix $A \in \text{supp}(\mathbf{A})$ to be bad at $i \in [m]$ if (31) holds (and $A$ is not $\tau$-rare). Further, denote this event by $\overline{\mathcal{E}}'(i, A)$. Then consider the following sequence of relations:

$$\begin{aligned}
\Pr[\mathbf{A} \text{ is bad at } i] &= \sum_{A \in \mathbb{F}_2^{n \times n}} \Pr[\mathbf{A} = A] \cdot \mathbf{1}_{\overline{\mathcal{E}}'(i,A)} \\
&= \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{A_{[i:m]} \in \mathbb{F}_2^{(m-i+1)\times n}} \Pr\left[\mathbf{A} = A | \mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \mathbf{1}_{\overline{\mathcal{E}}'(i,A)} && (32) \\
&= \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{\substack{S \subseteq [m], \\ |S| \leq \eta m}} \sum_{\substack{A_{[i:m]} \in \mathbb{F}_2^{(m-i+1)\times n}, \\ B(A) = S}} \Pr\left[\mathbf{A} = A | \mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \mathbf{1}_{\overline{\mathcal{E}}'(i,A)} && (33) \\
&= \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{\substack{S \subseteq [m], \\ |S| \leq \eta m}} \Pr\left[B(\mathbf{A}) = S | \mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \mathbf{1}_{\overline{\mathcal{E}}'(i,A)} && (34) \\
&< \sum_{A_{[i-1]} \in \mathbb{F}_2^{(i-1)\times n}} \Pr\left[\mathbf{A}_{[i-1]} = A_{[i-1]}\right] \cdot \sum_{\substack{S \subseteq [m], \\ |S| \leq \eta m}} \delta && (35) \\
&\leq 2^{h(\eta)m} \cdot \delta. && (36)
\end{aligned}$$

In the above (32) follows from definition of conditional probability, (33) follows by re-arranging terms, (34) follows by noting that $\overline{\mathcal{E}}'(i, A)$ is the same for all matrices that agree on $A_{[i-1]}$ and have the same $B(A)$, (35) follows from definition of $\overline{\mathcal{E}}'$ and (36) follows from the fact that the number of subsets of $[m]$ of size at most $\eta m$ (for $\eta < 1/2$) is upper bounded by $2^{h(\eta)m}$.

Taking union bound over all $m$ values of $i$ over the bound in (36) along with Lemma K.6 completes the proof. □

*K.3.2 A good matrix is a block source.* We now argue that $\mathbf{A}$ conditioned on $B(\mathbf{A})$ leads to a block source (that will suffice for our purposes):

Lemma K.11. *Let $S \in \text{supp}(B(\mathbf{A}))$ be a subset of $[m]$ (thus $|S| \leq \eta m$) or $S = \perp$ and $\epsilon_{\text{bad}}(S) \stackrel{def}{=} \Pr[\mathbf{A}$ is $(\tau, \delta)$-bad $|B(\mathbf{A}) = S]$. Then $\mathbf{A}|B(\mathbf{A}) = S$ is $\epsilon_{\text{bad}}(S)$-close to a $(\eta, n(1-\eta) - \log(1/\delta))$-block source.*

The proof of the above lemma follows from a similar argument in Claim 9 from [58] though there is a bug in the published proof [53] (which we correct below).

Proof of Lemma K.11. Let $A$ be a matrix with $B(A) = S$ and assume $A$ is not $(\tau, \delta)$-bad (note that this implies $S \neq \perp$). By definition, for every $i \notin S$, we have

$$q_i(A) \geq (1 - \eta)n. \tag{37}$$

Now fix any $i \notin S$. Then note that

$$\begin{aligned}
\Pr\left[\mathbf{A}_i = A_i | \mathbf{A}_{[i-1]} = A_{[i-1]}, B(\mathbf{A}) = S\right] &\leq \frac{\Pr\left[\mathbf{A}_i = A_i | \mathbf{A}_{[i-1]} = A_{[i-1]}\right]}{\Pr\left[B(\mathbf{A}) = S | \mathbf{A}_{[i-1]} = A_{[i-1]}\right]} && (38) \\
&\leq \frac{p_i(A)}{\delta} && (39) \\
&\leq 2^{-n(1-\eta)+\log(1/\delta)}. && (40)
\end{aligned}$$

In the above, (38) follows from the fact that for any three events $\overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2, \overline{\mathcal{E}}_3$, we have $\Pr\left[\overline{\mathcal{E}}_1 | \overline{\mathcal{E}}_2, \overline{\mathcal{E}}_3\right] \leq \Pr\left[\overline{\mathcal{E}}_1 | \overline{\mathcal{E}}_3\right] / \Pr\left[\overline{\mathcal{E}}_2 | \overline{\mathcal{E}}_3\right]$, (39) follows from definition of $p_i(\cdot)$ and $A$ not being $(\tau, \delta)$-bad and (40) follows from (37).

Taking into account that $A$ can be $(\tau, \delta)$-bad, we have that $\mathbf{A}|B(\mathbf{A}) = S$ is $\epsilon_{\text{bad}}(S)$-close to an $(\eta, n(1-\eta) - \log(1/\delta))$-block source. □

## K.4 A good block source A leads to Ax with high min-entropy

We finally prove that a good block source is enough for our purposes:

Lemma K.12. *Let $\mathbf{A}'$ be an $(\eta, n(1-\zeta))$-block source and $\mathbf{x} \in \mathbb{F}_2^n$ with $H_\infty(\mathbf{x}) = \alpha n$ such that*

$$\alpha \geq 2\Delta + \zeta.$$

*Further, the distributions $\mathbf{A}'$ and $\mathbf{x}$ are independent. Then there exists a subset $T \subseteq [m]$ with $|T| \geq (1-\eta)m$ such that $(\mathbf{A}'\mathbf{x})_T$ is $\epsilon$-close to $\mathcal{U}_{|T|}$ for*

$$\epsilon \leq |T| \cdot 2^{-\Delta n}.$$

We will need to use the following result to prove the above lemma:

Theorem K.13 ([23]). *Let $\mathbf{y}$ and $\mathbf{z}$ be independent random variables on $\mathbb{F}_2^n$ such that*

$$H_\infty(\mathbf{y}) + H_\infty(\mathbf{z}) \geq (1+\Delta)n.$$

*Then $(\mathbf{y}, \langle \mathbf{y}, \mathbf{z} \rangle)$ is $\epsilon_{IP}$-close to $\mathcal{D}_\mathbf{y} \times \mathcal{U}_1$, where $\mathcal{D}_\mathbf{y}$ is the distribution for $\mathbf{y}$ and*

$$\epsilon_{IP} \leq 2^{-\Delta n/2-1}.$$

We now prove Lemma K.12 via a very simple modification of Lemma 6 in [67]:

Proof of Lemma K.12. Since $\mathbf{A}'$ is an $(\eta, n(1-\zeta))$-block source, there exists a subset $T \subseteq [m]$ of size at least $(1-\eta)m$ such that for all $i \in T$ and for every $A_{[i-1]}$:

$$H_\infty\left(\mathbf{A}'_i | \mathbf{A}'_{[i-1]} = A_{[i-1]}\right) \geq n(1-\zeta). \tag{41}$$

For notational simplicity assume $T = [n']$ where $n' = (1-\eta)m$. To prove the lemma we will prove by induction on $i$ from $n'$ to 0 that for every $A_{[i]}$, the distribution of $\left(\mathbf{x}, \mathbf{A}'_{[i+1,n']}\mathbf{x}\right)$ conditioned on $\mathbf{A}'_{[i]} = A_{[i]}$ is $(n'-i)\cdot\epsilon_{\mathrm{IP}}$-close to $\mathcal{D}_{\mathbf{x}|A_{[i]}} \times \mathcal{U}_{n'-i}$, where $\mathcal{D}_{\mathbf{x}|A_{[i]}}$ is the distribution for $\mathbf{x}|\mathbf{A}'_{[i]} = A_{[i]}$. Note that this claim for $i = 0$ and Proposition K.1 (where the deterministic function just drops the $\mathbf{x}$ "part") is enough to prove the lemma.

The base case of $i = n'$ is trivial. Let us assume that the induction hypothesis is true for $i+1$. That is for every $A_{[i+1]}$, we have that the distribution of $\left(\mathbf{x}, \mathbf{A}'_{[i+2,n']}\mathbf{x}\right) \left| \mathbf{A}'_{[i+1]} = A_{[i+1]} \right.$ is $(n'-i-1)\cdot\epsilon_{\mathrm{IP}}$-close to $\mathcal{D}_{\mathbf{x}|A_{[i+1]}} \times \mathcal{U}_{n'-i-1}$.

We will now argue the claim for $i$. Towards that end fix an arbitrary $A_{[i]}$ and let $\hat{\mathcal{D}}$ be the distribution for $\mathbf{A}'_{i+1}|\mathbf{A}'_{[i]} = A_{[i]}$. Then since the claim on the distribution in the above paragraph holds for every $A_{i+1}$, we have that $\left(\mathbf{A}'_{i+1}, \mathbf{x}, \mathbf{A}'_{[i+2,n']}\mathbf{x}\right) \left| \mathbf{A}'_{[i]} = A_{[i]} \right.$ (call the correspondent distribution $\mathcal{D}_1$) is $(n'-i-1)\cdot\epsilon_{\mathrm{IP}}$-close to $\mathcal{D}_2 \stackrel{\text{def}}{=} \hat{\mathcal{D}} \times \mathcal{D}_{\mathbf{x}|A_{[i]}} \times \mathcal{U}_{n'-i-1}$.

We now apply Proposition K.1 on $\mathcal{D}_1$ and $\mathcal{D}_2$ (where the deterministic function puts the second component as the new first component and the new second component is the inner product of the earlier first two components), to get that

$$\left(\mathbf{x}, \mathbf{A}'_{[i+1,n']}\mathbf{x}\right) \left| \mathbf{A}'_{[i]} = A_{[i]} \right. \quad \sim_{(n'-i-1)\cdot\epsilon_{\mathrm{IP}}} \quad \left(\mathbf{x}, \left(\left\langle \mathbf{A}'_{i+1}, \mathbf{x} \right\rangle, u_{i+2}, \ldots, u_{n'}\right)^{\mathrm{T}}\right) \left| \mathbf{A}'_{[i]} = A_{[i]}, \right.$$

where the $u_j$ are independent and uniformly random bits. Since these bits are independent of $(\mathbf{x}, \left\langle \mathbf{A}'_{i+1}, \mathbf{x} \right\rangle)$, we have by Theorem K.13 that $\left(\mathbf{x}, \left(\left\langle \mathbf{A}'_{i+1}, \mathbf{x} \right\rangle, u_{i+2}, \ldots, u_{n'}\right)^{\mathrm{T}}\right) \left| \mathbf{A}'_{[i]} = A_{[i]} \right.$ is $\epsilon_{\mathrm{IP}}$-close to $\left(\mathbf{x}, (u_{i+1}, u_{i+2}, \cdots, u_{n'})^{\mathrm{T}}\right) \left| \mathbf{A}'_{[i]} = A_{[i]} \right.$, which is exactly $\mathcal{D}_{\mathbf{x}|A_{[i]}} \times \mathcal{U}_{n'-i}$. Then by the triangle inequality we have that the distribution of $\left(\mathbf{x}, \mathbf{A}'_{[i+1,n']}\mathbf{x}\right) \left| \mathbf{A}'_{[i]} = A_{[i]} \right.$ is $(n'-i)\cdot\epsilon_{\mathrm{IP}}$-close to $\mathcal{D}_{\mathbf{x}|A_{[i]}} \times \mathcal{U}_{n'-i}$, as desired.

We finally note that the assumption of $T = [n']$ is almost WLOG since in the more general case we do the above argument for all $i \in [m]$ but make the above argument only for indices in $T$ (while the conditioning also happens for rows not in $T$). □

## K.5 Putting everything together

We now have all the pieces at our disposal and are finally ready to prove Theorem K.2. We note that Lemmas K.11 and K.12 imply the following result (if $\epsilon_1 = \epsilon_2 = 0$).

Lemma K.14. *Let $\mathbf{x}$ and $\mathbf{A}$ be independent variables such that*

$$H_\infty(\mathbf{x}) \geq (2\Delta + \sqrt{\tau+\gamma}) \cdot n + \log(1/\delta),$$

*and*

$$H_\infty(\mathbf{A}) \geq (1-\gamma)mn.$$

*Then the distribution on $\mathbf{Ax}$ is $\epsilon_{\mathrm{bad}} + m \cdot 2^{-\Delta n}$-close to a distribution with min entropy at least $(1-\sqrt{\tau+\gamma})n$.*

Proof. For each fixing of $B(\mathbf{A}) = S$, Lemmas K.11 and K.12 imply that $\mathbf{Ax}$ conditioned on $B(\mathbf{A}) = S$ is $\epsilon_{\mathrm{bad}}(S) + m \cdot 2^{-\Delta n}$-close to a distribution with min entropy at least $(1 - \sqrt{\tau + \gamma})n$, where we defined $\epsilon_{\mathrm{bad}}(S) = \Pr[\mathbf{A} \text{ is } (\tau, \delta)\text{-bad} | B(\mathbf{A}) = S]$.

Then taking into account all possibilities of $B(\mathbf{A})$, the distribution on $\mathbf{Ax}$, which is a convex combination of distributions, is $\epsilon'$-close to a distribution with min entropy at least $(1 - \sqrt{\tau + \gamma})n$, where

$$\epsilon' \le \sum_{S \subseteq [m], |S| \le \eta m \text{ or } S = \perp} \left(\epsilon_{\mathrm{bad}}(S) + m \cdot 2^{-\Delta n}\right) \cdot \Pr[B(\mathbf{A}) = S] = \epsilon_{\mathrm{bad}} + m \cdot 2^{-\Delta n},$$

as desired. □

Finally to prove Theorem K.2, we will extend Lemma K.14. Before we do that we note that we use the following instantiation of parameters

$$\Delta = \tau = \gamma,$$
$$\delta = 2^{-\gamma m - h(\sqrt{2\gamma})m},$$

in Lemma K.14 and this implies the claimed parameters in Theorem K.2.

Now assume we are given $H_\infty^{\epsilon_1}(\mathbf{A}'|\mathbf{Y}) \ge (1-\gamma)nm, H_\infty^{\epsilon_2}(\mathbf{x}'|\mathbf{Y}) \ge \alpha n$. Moreover, for every $y \in \mathrm{supp}(\mathbf{Y})$, we have that conditioned on $\mathbf{Y} = y$, $\mathbf{A}'$ and $\mathbf{x}'$ are independent. Our goal is to prove $H_\infty^{\epsilon_1+\epsilon_2+2^{-\Omega(\gamma m)}}(\mathbf{A}'\mathbf{x}'|\mathbf{Y}) \ge (1 - \sqrt{2\gamma})m$.

We can assume that there are two events $\mathcal{E}_1, \mathcal{E}_2$ with $\Pr[\mathcal{E}_1] \ge 1 - \epsilon_1$ and $\Pr[\mathcal{E}_2] \ge 1 - \epsilon_2$ and for every $y \in \mathrm{supp}(\mathbf{Y})$, we have

(D1) $H_\infty(\mathcal{E}_1\mathbf{A}'|\mathbf{Y} = y) \ge (1-\gamma)nm$,
(D2) $H_\infty(\mathcal{E}_2\mathbf{x}'|\mathbf{Y} = y) \ge \alpha n$,
(D3) conditioned on $\mathbf{Y} = y$, $(\mathbf{A}', \mathbf{1}_{\mathcal{E}_1})$ and $(\mathbf{x}', \mathbf{1}_{\mathcal{E}_2})$ are independent.

(D1) and (D2) are satisfied by the definition of conditional smooth min-entropy. (D3) can be assumed due to the facts that $\mathbf{A}'$ and $\mathbf{x}'$ are independent conditioned on $\mathbf{Y} = y$, (D1) only involves $\mathcal{E}_1$ and $\mathbf{A}'$, and (D2) only involves $\mathcal{E}_2$ and $\mathbf{x}'$.

Then we can construct a distribution $(\mathbf{A}, \mathbf{x})$ joint with $\mathbf{Y}$ such that for every $y \in \mathrm{supp}(\mathbf{Y})$, we have

(E1) $\Pr[\mathbf{A} = A|\mathbf{Y} = y] \ge \Pr[\mathcal{E}_1, \mathbf{A}' = A|\mathbf{Y} = y]$ for every $A \in \mathbb{F}^{m \times n}$, and moreover $H_\infty(\mathbf{A}|\mathbf{Y} = y) \ge (1-\gamma)nm$,
(E2) $\Pr[\mathbf{x} = x|\mathbf{Y} = y] \ge \Pr[\mathcal{E}_2, \mathbf{x}' = x|\mathbf{Y} = y]$ for every $x \in \mathbb{F}^n$, and moreover $H_\infty(\mathbf{x}|\mathbf{Y} = y) \ge \alpha n$,
(E3) conditioned on $\mathbf{Y} = y$, $\mathbf{A}$ and $\mathbf{x}$ are independent.

To see how to guarantee (E1), we focus on a $y \in \mathrm{supp}(\mathbf{Y})$. Start with the vector $\theta$ such that $\theta_A = \Pr[\mathcal{E}_1, \mathbf{A}' = A|\mathbf{Y} = y]$ for every $A \in \mathbb{F}^{m \times n}$. Notice that $\|\theta\|_1 = \Pr[\mathcal{E}_1|\mathbf{Y} = y] \le 1$ and $\|\theta\|_\infty \le 2^{-(1-\gamma)nm}$. We then increase coordinates of $\theta$ so that $\|\theta\|_1 = 1$ and $\|\theta\|_\infty \le 2^{-(1-\gamma)nm}$ is maintained. This is doable since $2^{-(1-\gamma)nm} \cdot 2^{nm} \ge 1$. Then we can guarantee (E1) by defining $\Pr[\mathbf{A} = A|\mathbf{Y} = y] = \theta_A$ for the new vector $\theta$. Similarly we can guarantee (E2). (E3) can be guaranteed easily. Thus, for every $y \in \mathrm{supp}(\mathbf{Y})$ and $z \in \mathbb{F}^m$, we have

$$\begin{aligned}
\Pr[\mathcal{E}_1, \mathcal{E}_2, \mathbf{A}'\mathbf{x}' = z|\mathbf{Y} = y] &= \sum_{A, x: Ax = z} \Pr[\mathcal{E}_1, \mathcal{E}_2, \mathbf{A}' = A, \mathbf{x}' = x|\mathbf{Y} = y] \\
&= \sum_{A, x: Ax = z} \Pr[\mathcal{E}_1, \mathbf{A}' = A|\mathbf{Y} = y] \Pr[\mathcal{E}_2, \mathbf{x}' = x|\mathbf{Y} = y] \\
&\le \sum_{A, x: Ax = z} \Pr[\mathbf{A} = A|\mathbf{Y} = y] \Pr[\mathbf{x} = x|\mathbf{Y} = y] \\
&= \sum_{A, x: Ax = z} \Pr[\mathbf{A} = A, \mathbf{x} = x|\mathbf{Y} = y] \\
&= \Pr[\mathbf{Ax} = z|\mathbf{Y} = y]. \qquad (42)
\end{aligned}$$

The second and third equalities are by (D3) and (E3) respectively and the inequality is by (E1) and (E2).

Our proof already shows that $H_\infty^{\epsilon_*}(\mathbf{Ax}) \ge (1 - \sqrt{2\gamma})m$ for some $\epsilon_* = 2^{-\Omega(\gamma m)}$, as we have (E1), (E2) and (E3). By the definition of $H_\infty^{\epsilon_*}$, there exists an event $\mathcal{E}_*$ such that $\Pr[\mathcal{E}_*] \ge 1 - \epsilon_*$ and $\Pr[\mathcal{E}_*, \mathbf{Ax} = z|\mathbf{Y} = y] \le 2^{-(1-\sqrt{2\gamma})m}$ for every $y \in \mathrm{supp}(\mathbf{Y})$ and $z \in \mathbb{F}^m$. Thus, by (42), there exists an event $\mathcal{E}_*'$ with $\Pr[\mathcal{E}_*'] = \Pr[\mathcal{E}_*] \ge 1 - \epsilon_*$ such that for every $y \in \mathrm{supp}(\mathbf{Y})$ and $z \in \mathbb{F}^m$:

$$\Pr[\mathcal{E}_*', \mathcal{E}_1, \mathcal{E}_2, \mathbf{A}'\mathbf{x}' = z|\mathbf{Y} = y] \le \Pr[\mathcal{E}_*, \mathbf{Ax} = z|\mathbf{Y} = y] \le 2^{-(1-\sqrt{2\gamma})m},$$

Notice that $\Pr[\mathcal{E}_*', \mathcal{E}_1, \mathcal{E}_2] \ge 1 - (\epsilon_* + \epsilon_1 + \epsilon_2)$ by union bound. By the definition of conditional smooth min-entropy, we have $H^{\epsilon_* + \epsilon_1 + \epsilon_2}(\mathbf{A}'\mathbf{x}'|\mathbf{Y}) \ge (1 - \sqrt{2\gamma})m$, which completes the proof of Theorem K.2.

# L MISSING DETAILS FROM SECTION 6

## L.1 The case of $k \ge N$

For simplicity we assume $k$ is an integer power of 2. In the first iteration, each $P_i$ with odd $i$ sends $A_i$ to $P_{i+1}$, who then computes $B_{i+1}^1 := A_i A_{i+1}$; this iteration takes $N^2$ rounds. In the second iteration, each $P_i$ with $i \mod 4 = 2$ sends $B_i^1$ to $P_{i+2}$, who then computes $B_{i+2}^2 = B_i^1 B_{i+2}^1$; this iteration takes $N^2 + 1$ rounds. In general, in the $t$-th iteration for $t \in [\log k]$, each $P_i$ with $i$

mod $2^t = 2^{t-1}$ sends $B_i^{t-1}$ to player $P_{i+2^{t-1}}$, who then computes $B_{i+2^{t-1}}^t = B_i^{t-1}B_{i+2^{t-1}}^{t-1}$; the iteration takes $N^2 + 2^{t-1} - 1$ rounds. So in a total of $O(N^2 \log k + k)$ rounds, player $P_k$ will know the product $A_1A_2 \cdots A_k$. Using additional $O(k+N)$ rounds, $P_0$ can send $x$ to $P_k$. The whole protocol takes $O(N^2 \log k + k)$ rounds. The bound can be slightly improved to $O(N^2 \log(k/N) + k)$ by running the merging procedure for only $\log(k/N)$ iterations.

## L.2 Proof of Lemma 6.4

Proof. We will prove the claim by induction. For the base case of $i = 1$, Lemma 6.2 implies that (recall that $\widetilde{\mathbf{m}}^1 = \mathbf{m}^1(t_1) = \mathbf{m}^1(\gamma N/4)$ and $\mathbf{y}_0 = \mathbf{x}$): $H_\infty^{\epsilon^*}\left(\mathbf{y}_0|\widetilde{\mathbf{m}}^1\right) \geq H_\infty(\mathbf{x}) - \gamma N/4 - \log(1/\epsilon^*) \geq N(1-\gamma/4-\gamma/2) \geq N(1-\gamma-\sqrt{2\gamma})$. Thus, (7) holds for $i = 1$.

We assume (7) holds for some $i \geq 1$; we prove that it also holds with $i$ replaced by $i+1$. For any interval $[\ell, r]$ we use $\mathbf{A}_{[\ell:r]}$ to denote the tuple $(\mathbf{A}_\ell, \ldots, \mathbf{A}_r)$. Conditioned on $\widetilde{\mathbf{m}}^i = \widetilde{m}^i$, since all communication between $P_1, \ldots, P_{i-1}$ and $P_i, \ldots, P_{k+1}$ are independent, we have

(B1) $(\mathbf{x}, \mathbf{A}_{[1:i-1]})$ is independent of $\mathbf{A}_{[i,k]}$.
(B2) $\mathbf{y}_{i-1}$ and $\widetilde{\mathbf{m}}^{[i-1]}$ are determined by $(\mathbf{x}, \mathbf{A}_{[1:i-1]})$.
(B3) $\mathbf{m}^{i+1}(t_i)$ is determined by $\mathbf{A}_{[i:k]}$.

The above properties imply the following, which will be used many times in our analysis:

(C) Conditioned on $\widetilde{\mathbf{m}}^i = \widetilde{m}^i$, $(\mathbf{x}, \mathbf{A}_{[1:i-1]}, \mathbf{y}_{i-1}, \widetilde{\mathbf{m}}^{[i-1]})$ and $(\mathbf{A}_{[i,k]}, \mathbf{m}^{i+1}(t_i))$ are independent.

By (C), $\mathbf{y}_{i-1}\left|\left(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)\right)\right. = \left(\widetilde{m}^{[i]}, m^{i+1}(t_i)\right)$ has the same distribution as $\mathbf{y}_{i-1}\left|\widetilde{\mathbf{m}}^{[i]} = \widetilde{m}^{[i]}\right.$. By the inductive hypothesis, we have

$$\begin{aligned} H_\infty^{i\epsilon^*}\left(\mathbf{y}_{i-1}\left|\left(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)\right)\right.\right) &= H_\infty^{i\epsilon^*}\left(\mathbf{y}_{i-1}\left|\widetilde{\mathbf{m}}^{[i]}\right.\right) \\ &\geq N(1-\gamma-\sqrt{2\gamma}), \end{aligned} \tag{43}$$

where the equality is by Lemma **??**. Further, Lemma 6.2 (with $\epsilon = 0$ and $\epsilon' = \epsilon^*/3$) implies that

$$\begin{aligned} H_\infty^{\epsilon^*/3}\left(\mathbf{A}_i\left|\left(\widetilde{\mathbf{m}}^i, \mathbf{m}^{i+1}(t_i)\right)\right.\right) &\geq N^2 - 2i \cdot \frac{\gamma}{4} \cdot N - \log(\epsilon^*/3) \\ &\geq N^2(1-\gamma). \end{aligned} \tag{44}$$

Again by (C), $\mathbf{A}_i\left|\left(\widetilde{\mathbf{m}}^i, \mathbf{m}^{i+1}(t_i)\right)\right. = \left(\widetilde{m}^i, m^{i+1}(t_i)\right)$ has the same distribution as $\mathbf{A}_i\left|\left(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)\right)\right. = \left(\widetilde{m}^{[i]}, m^{i+1}(t_i)\right)$. Lemma **??** and (44) implies that

$$H_\infty^{\epsilon^*/3}\left(\mathbf{A}_i\left|\left(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)\right)\right.\right) \geq N^2(1-\gamma). \tag{45}$$

By (43), (45), and (by (C)) the fact that $\mathbf{A}_i$ and $\mathbf{y}_{i-1}$ are independent conditioned on $(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)) = (\widetilde{m}^{[i]}, m^{i+1}(t_i))$, we have the following via Theorem 6.3:

$$H_\infty^{(i+2/3)\epsilon^*}\left(\mathbf{y}_i = \mathbf{A}_i\mathbf{y}_{i-1}\left|\left(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)\right)\right.\right) \geq N(1-\sqrt{2\gamma}),$$

as long as $1-\gamma-\sqrt{2\gamma} \geq 3\gamma+\sqrt{2\gamma}+h(\sqrt{2\gamma})$, which follows from (6). By applying Lemma 6.2 again (with $\epsilon = (i+2/3)\epsilon^*$ and $\epsilon' = \epsilon^*/3$), we get that[27]: $H_\infty^{(i+1)\epsilon^*}\left(\mathbf{y}_i\left|\widetilde{\mathbf{m}}^{[i+1]}\right.\right) \geq H_\infty^{(i+2/3)\epsilon^*}\left(\mathbf{y}_i\left|\left(\widetilde{\mathbf{m}}^{[i]}, \mathbf{m}^{i+1}(t_i)\right)\right.\right) - N\gamma/4 - \log(\epsilon^*/3) \geq N(1-\sqrt{2\gamma}-\frac{\gamma}{4}-\frac{\gamma}{2}) \geq N(1-\gamma-\sqrt{2\gamma})$, as desired. □

## L.3 Proof of Theorem 6.5

We will need the following result in our proof.

Lemma L.1. *Assume $H_\infty^\epsilon(X|Y) \geq L$. Then for every function $f : \mathrm{supp}(Y) \to \mathrm{supp}(X)$, we have $\Pr[f(Y) = X] \leq \epsilon + 2^{-L}$.*

Proof. By the definition of $H_\infty^\epsilon(X|Y)$, there exists an event $\mathcal{E}$ such that $\Pr[\mathcal{E}] \geq 1-\epsilon$ and for every $x \in \mathrm{supp}(X), y \in \mathrm{supp}(Y)$, we have $\Pr[\mathcal{E}, X = x | Y = y] \leq 2^{-L}$. In particular, $\Pr[\mathcal{E}, X = f(y)|Y = y] \leq 2^{-L}$ for every $y \in \mathrm{supp}(Y)$. So $\Pr[\mathcal{E}, f(Y) = X] \leq 2^{-L}$, which implies $\Pr[f(Y) = X] \leq 2^{-L} + \epsilon$ since $\Pr[\mathcal{E}] \geq 1-\epsilon$. □

Proof of Theorem 6.5. Let $\Pi$ be any protocol with at most $t_{k+1} = \gamma(k+1)N/4$ rounds. Lemma 6.4 implies that at the end of the protocol, we have

$$H_\infty^{(k+1)\epsilon^*}\left(\mathbf{y}_k\left|\widetilde{\mathbf{m}}^{[k+1]}\right.\right) \geq N(1-\gamma-\sqrt{2\gamma}).$$

This implies that even if the player $k+1$ is given $\widetilde{\mathbf{m}}^{[k+1]}$ (instead of only $\widetilde{\mathbf{m}}^{k+1} = \mathbf{m}^{k+1}(t_{k+1})$), it can only output the correct answer with probability at most

$$(k+1)\epsilon^* + 2^{-N(1-\gamma-\sqrt{2\gamma})},$$

[27] Note that we are not conditioning on $\mathbf{m}^{i+1}(t_{i+1}) = \mathbf{m}^{i+1}(t_i + \gamma N/4)$ instead of the earlier $\mathbf{m}^{i+1}(t_i)$.

by Lemma L.1 (here $f(Y)$ is the output at $P_{k+1}$ for $Y = \widetilde{\mathbf{m}}^{[k+1]}$ and $X = \mathbf{y}_k$). For large enough $N$ the above quantity is less than $1/2$. □

## L.4 Why Shannon entropy does not work for our proof of Lemma 6.4

We will use $H_{\mathrm{Sh}}(\mathcal{D})$ to denote the Shannon entropy of $\mathcal{D}$, which is defined as follows:

$$H_{\mathrm{Sh}}(\mathcal{D}) = -\sum_{x \in \mathrm{supp}(\mathcal{D})} \Pr_{X \sim \mathcal{D}}[X = x] \log_2 \left( \Pr_{X \sim \mathcal{D}}[X = x] \right).$$

If we had used Shannon entropy instead of (smooth) min-entropy, we would have to prove a bound of the following form. Let $f : \mathbb{F}_2^{N \times N} \to \mathbb{F}_2^m$ be an arbitrary function. Then as long as $H_{\mathrm{Sh}}(\mathbf{x})$ and $H_{\mathrm{Sh}}(\mathbf{A})$ are big enough, $\mathbf{Ax}|f(\mathbf{A})$ should have entropy strictly bigger than $H_{\mathrm{Sh}}(\mathbf{x})$ (or be close to a distribution that has high enough entropy) as long as $m$ is a small fraction of $N^2$. We now give an example to show this is not possible.

Fix arbitrary linearly independent vectors $\mathbf{x}_1^*, \ldots, \mathbf{x}_t^*$ for $t = \alpha N$. Then define the following distribution on $\mathbf{x}$: probability mass of $1 - \alpha$ is distributed uniformly over the span of $\mathbf{x}_1^*, \ldots, \mathbf{x}_t^*$ (call this span $S$) and the remaining mass of $\alpha$ is distributed uniformly over a null space of $S$. Then note that

$$H_{\mathrm{Sh}}(\mathbf{x}) = (1 - \alpha) \cdot t + \alpha \cdot (N - t) = 2\alpha(1 - \alpha) \cdot N.$$

Now consider the case when $\mathbf{A}$ is uniformly distributed (i.e. $H_{\mathrm{Sh}}(\mathbf{A}) = N^2$) and $f(\mathbf{A}) = \left(\mathbf{Ax}_1^*, \ldots, \mathbf{Ax}_t^*\right)$. Then note that if $\mathbf{x} \in S$, then $H_{\mathrm{Sh}}(\mathbf{Ax}|f(\mathbf{A})) = 0$. This implies that

$$H_{\mathrm{Sh}}(\mathbf{Ax}|f(\mathbf{A})) \leq (1 - \alpha) \cdot 0 + \alpha \cdot N,$$

which is about a factor two smaller than $H_{\mathrm{Sh}}(\mathbf{x})$ (for small enough $\alpha > 0$).

## L.5 Existing lower bounds techniques for the matrix-chain problem

We remark that the existing technique of [18] that "stitches" the lower bounds induced by cuts can only give a lower bound of $\Omega(N)$: for any edge $(P_i, P_{i+1})$ on the path, we can only prove a lower bound of $\Omega(N)$ on the number of bits that need to be exchanged between $P_i$ and $P_{i+1}$, since if suffices for $P_i$ to send the product $\mathbf{A}_k\mathbf{A}_{k-1}\cdots\mathbf{A}_1\mathbf{x}$ to $P_{i+1}$. The lower bound given by [18] is then the minimum number of rounds needed to make sure that $\Omega(N)$ bits are exchanged between $\{P_0, P_1, \cdots, P_i\}$ and $\{P_{i+1}, P_{i+2}, \cdots, P_{k+1}\}$ for every $i$, which can only be $\Omega(N)$. However, this analysis does not capture a very simple fact: $P_i$ needs to know $\mathbf{A}_k\mathbf{A}_{k-1}\cdots\mathbf{A}_1\mathbf{x}$ before it can be sent to $P_{i+1}$.