Electronic Colloquium on Computational Complexity, Report No. 111 (2020)

Lifting: As Easy As 1,2,3

Ian Mertz
University of Toronto

Toniann Pitassi
University of Toronto & IAS

July 19, 2020

Abstract

Query-to-communication lifting theorems translate lower bounds on query complexity to lower bounds for the corresponding communication model. In this paper, we give a simplified proof of deterministic lifting (in both the tree-like and dag-like settings). Whereas previous proofs used sophisticated Fourier analytic techniques, our proof uses elementary counting together with the sunflower lemma.

Contents

ISSN 1433-8092

1 Introduction

A *query-to-communication* lifting theorem is a reductive lower bound technique that translates lower bounds on query complexity (such as decision tree complexity) to lower bounds for the corresponding communication complexity model. There is a substantial body of work proving lifting theorems for a variety of flavors of query-to-communication, including: deterministic [RM99, GPW15, dRNV16, WYY17, CKLM17], nondeterministic [GLM+16, Göö15], randomized [GPW17, CFK+19] degree-to-rank [She11, PR17, PR18, RPRC16] and nonnegative degree to nonnegative rank [CLRS16, KMR17].

In these papers and others, lifting theorems have been applied to simplify and resolve some long-standing open problems, including new separations in communication complexity, [GP18, GPW15, GPW17, CKLM17, CFK+19], proof complexity [GLM+16, HN12, GP18, dRNV16, dRMN+19, GKMP20] monotone circuit complexity [GGKS18], monotone span programs and linear secret sharing schemes [RPRC16, PR17, PR18], and lower bounds on the extension complexity of linear and semi-definite programs [CLRS16, KMR17, LRS15].

At the heart of these proofs is a *simulation theorem*. For a function $f : \{0,1\}^n \to R$, and a function $g : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ (called the *gadget*), their composition $f \circ g^n : \mathcal{X}^n \times \mathcal{Y}^n \to R$ is defined by

$$(f \circ g^n)(x, y) := f(g(x_1, y_1), \ldots, g(x_n, y_n)).$$

Here, Alice holds $x \in \mathcal{X}^n$ and Bob holds $y \in \mathcal{Y}^n$. Typically g is the popular *index* gadget $\textsc{Ind}_m : [m] \times \{0,1\}^m \to \{0,1\}$ mapping (x, y) to the xth bit of y. For $m = n^{O(1)}$, and for every f the deterministic simulation theorem [RM99, GPW15] states that:

$$\mathbf{P}^{cc}(f \circ \textsc{Ind}_m^n) = \mathbf{P}^{dt}(f) \cdot \Theta(\log n).$$

The proof of this theorem has evolved considerably since [RM99], applying to a wider range of gadgets, and with more sharpened results giving somewhat improved parameters and simulation theorems for the more difficult settings of randomized and dag-like lifting. However, nearly all proofs of even the basic deterministic simulation theorem are fairly involved, and use tools from the Fourier analysis of Boolean functions, together with somewhat intricate counting arguments.

Lifting using the sunflower lemma. The primary purpose of this paper is to give a readable, self-contained and simplified proof of the deterministic query-to-communication lifting theorem. Our proof uses the same basic setup as in previous arguments, but our proof of the main invariant – showing that any large rectangle can be decomposed into a part that has structure and a part that is pseudo-random – is proven by a direct reduction to the famous sunflower lemma.

The sunflower lemma is one of the most important examples of a structure-versus-randomness theorem in combinatorics. A sunflower with r petals is a collection of r sets such that the intersection of each pair is equal to the intersection of all of them. The sunflower lemma of Erdös and Rado [ER60] roughly states that in any sufficiently large k-uniform set system (of size about w^w) must contain a sunflower. A recent breakthrough result due to Alweiss et al. [ALWZ20] proves the sunflower lemma with significantly improved parameters, making a huge step towards resolving the longstanding open problem obtaining optimal parameters.

Both the original Sunflower Lemma as well as Rossman's robust version [Ros10] have played an important role in recent advances in theoretical computer science. Most famously, Razborov proved the first superpolynomial lower bounds for monotone circuits computing the Clique function, using the Sunflower Lemma. It has also been a fundamental tool used to obtain a wide variety of other hardness results including: hardness of approximation, matrix multiplication, cryptography, and data structure lower bounds. (See [ALWZ20] for a nice survey of the many applications to Computer Science.)

In all of these lower bounds, the central idea is to use the sunflower lemma in order to "tame" a protocol or algorithm, in order to show that at each step of the computation, the underlying set of inputs consistent so far can be partitioned into a structured part and a random part. This allows the algorithm to be massaged into a simpler form, where the lower bound is easier to prove. Since lifting theorems are attempting to do precisely the same thing, it is natural to expect that there should be a connection between the two lines of research. Indeed, [LLZ18], made an explicit connection between sunflowers and randomness extractors where the latter is again a primary tool used in many if not all of the proofs of lifting.

Our main result is a self-contained proof of the deterministic lifting theorem, using the sunflower lemma plus an elementary counting argument. As a nice side effect, our simplified proof easily extends to the following generalizations:

- a gadget of quasilinear size; most previous constructions required gadget size at least n^2
- a lifting theorem for dag-like protocols, as proven in [GGKS18]
- a gadget of size depending only on the query complexity, as proven in [GKMP20]

For all corresponding previous theorems, our proof simplifies one of the main technical components of the result, as well as improving on the stated parameters. We note that our results extend straightforwardly to the real communication setting as well.

Organization for the rest of the paper. After setting up the preliminaries, in Section 3 we present our main contribution: a simplified proof of lifting via the sunflower lemma. Then for the remainder of the paper we investigate various extensions of this basic lifting theorem. In Section 4 we show that the gadget size m can be improved to $n^{1+\epsilon}$, and by sacrificing in the strength of the lifting theorem we can even push it down to $O(n \log n)$. In Section 5 we show that we can lift dag-like query complexity to dag-like communication complexity. In Section 6 we show that m can be made $\mathrm{poly}(\mathbf{P}^{dt}(f))$ with no other dependence on n. For these extensions we give only a sketch of how the proof differs from the proof of the basic lifting theorem, and where necessary how our results fit into the context of their original proofs.

Related results. After writing a first draft of our sunflower-based proof, Lovett et al. [LMZ20] independently published a proof similar to the one we present, stated entirely in the language of sunflowers by drawing a connection between the central subroutines in lifting theorems and sunflower lemmas. Our proof is stated in the language of lifting theorems, which will hopefully give it utility to serve as an entry point to the literature on lifting theorems, as well as covering some of the recent generalizations and improvements to this central result. They also achieve a similar tradeoff of the gadget size and lower bound obtained, which gives them a $\tilde{O}(n \log^2 n)$ sized gadget at one extreme and $n^{1+\epsilon}$ at the other.

2 Preliminaries

We will use n to denote the length of the input, $N \leq n$ to denote an arbitrary number, m to denote an external parameter, and for this preliminaries section we will use $\mathcal{S}$ to denote an arbitrary set. We will mostly focus on two types of universes, $\mathcal{S}^N$ and $(\mathcal{S}^m)^N$. In the case of $\mathcal{S}^N$ we often refer to $i \in [N]$ as being a *coordinate*, while in the case of $(\mathcal{S}^m)^N$ we often refer to $i \in [N]$ as being a *block*.

Basic notation. For a set $S \subseteq \mathcal{S}$ we write $\bar{S} \coloneqq \mathcal{S} \setminus S$. For a set $\mathcal{S}$ and a set $I \subseteq [N]$ we say a string x is in $\mathcal{S}^I$ if each value in x is an element of $\mathcal{S}$ indexed by a unique element of I. For a string

$x \in \mathcal{S}^N$ and $I \subseteq [N]$ we define $x[I] \in \mathcal{S}^I$ to be the values of x at the locations in I, and for a string $y \in (\mathcal{S}^m)^N$ and $I \subseteq [N]$, $\alpha \in [m]^I$ we define $y[I, \alpha] \in \mathcal{S}^I$ to be the values of y at the locations α_i for each $i \in I$. For a set $X \subseteq \mathcal{S}^N$ we define $X_I \subseteq \mathcal{S}^I$ to be the set that is the projection of X onto coordinates I, and for a set $Y \subseteq (\mathcal{S}^m)^N$ we define $Y_I \in (\mathcal{S}^m)^I$ likewise. For a set system $\mathcal{F}$ over $\mathcal{S}$ and a set $S \subseteq \mathcal{S}$, we define $\mathcal{F}_{\overline{S}} := \{\gamma \setminus S : S \subseteq \gamma \in \mathcal{F}\}$.[1]

Definition 2.1. Let $\gamma \subseteq [mN]$. Treating each element in γ as being a pair (i, a) where $i \in [N]$ and $a \in [m]$, we say γ is *over* $(\mathcal{S}^m)^N$, meaning that for $s \in (\mathcal{S}^m)^N$ each $(i, a) \in \gamma$ indicates an element $s[i, a] \in \mathcal{S}$. We sometimes say (i, a) is a *pointer*.

For γ over $(\mathcal{S}^m)^N$, γ is a *block-respecting* subset of $[mN]$ if γ contains at most one element per block, or in other words if $i \neq i'$ for all distinct $(i, a), (i', a') \in \gamma$. We can represent γ by a pair (I, α), where $I \subseteq [N]$ and $\alpha \in [m]^I$; here γ chooses one element (indicated by α_i) from each block $i \in I$. A set system $\mathcal{F}$ over $(\mathcal{S}^m)^N$ is block respecting if all elements $\gamma \in \mathcal{F}$ are block respecting.

We say that a set $\rho \in \{0, 1, *\}^N$ is a *restriction*, or sometimes a *partial assignment*. We denote by $\text{free}(\rho) \subseteq [N]$ the variables assigned a star, and define $\text{fix}(\rho) := [N] \setminus \text{free}(\rho)$. If we have two restrictions ρ, ρ' such that $\text{fix}(\rho) \cap \text{fix}(\rho') = \emptyset$, then we define $\rho \cup \rho'$ to be the restriction which assigns $\text{fix}(\rho)$ to $\rho[\text{fix}(\rho)]$ and $\text{fix}(\rho')$ to $\rho'[\text{fix}(\rho')]$, with all other coordinates being assigned $*$.

In general in this paper we will use bold letters to denote random variables. For a set S we denote by $\boldsymbol{S} \in S$ the random variable that is uniform over S. For $S \subseteq \mathcal{S}^N$ and $I \subseteq [N]$ we denote by $\boldsymbol{S}_I$ the marginal distribution over coordinates I of the uniform distribution over S; note that the random draw is taken over the original set S before marginalizing to the coordinates I, rather than being the uniform distribution over S_I.

Definition 2.2. Let S be a set. For a random variable $\boldsymbol{s} \in S$ we define its *min-entropy* by $\mathbf{H}_\infty(\boldsymbol{s}) := \min_s \log(1/\mathbf{Pr}[\,\boldsymbol{s} = s\,])$. We also define the *deficiency* of $\boldsymbol{s}$ by $\mathbf{D}_\infty(\boldsymbol{s}) := \log|S| - \mathbf{H}_\infty(\boldsymbol{s}) \geq 0$. When $\boldsymbol{s}$ is chosen from a set $S \subseteq \mathcal{S}^N$ or from a set system $\mathcal{F}$ over $\mathcal{S}^N$, we define its *blockwise min-entropy* by $\min_{\emptyset \neq I \subseteq [N]} \frac{1}{|I|}\mathbf{H}_\infty(\boldsymbol{s}_I)$, or in other words the least (normalized) marginal min-entropy over all subsets I of the coordinates $[N]$.

Search problems A *search problem* is a relation $f \subseteq \mathcal{Z} \times \mathcal{O}$ such that for every $z \in \mathcal{Z}$ there exists some $o \in \mathcal{O}$ such that $(z, o) \in f$. Let $f(z) \neq \emptyset$ denote the set of all $o \in \mathcal{O}$ such that $(z, o) \in f$. Likewise a *bipartite search problem* is a relation $F \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$ such that $F(x, y) \neq \emptyset$, where $F(x, y)$ is defined analogously to $f(z)$. We say that f is on $\mathcal{Z}$ and F is on $\mathcal{X} \times \mathcal{Y}$.

Definition 2.3. Let $m \in \mathbb{N}$. The *index gadget*, denoted $\textsc{Ind}_m$, is a Boolean function which takes two inputs $x \in [m]$ and $y \in \{0, 1\}^m$, and outputs $y[x]$. We will often have N separate instances of the index gadget, which we denote by $\textsc{Ind}_m^N$ and which is a function which takes two inputs $x \in [m]^N$ and $y \in (\{0, 1\}^m)^N$ and outputs the Boolean string $(y[i, x_i])_{i \in [N]}$. For a search problem f with $\mathcal{Z} = \{0, 1\}^n$, the *lifted search problem* $f \circ \textsc{Ind}_m^n$ is a bipartite search problem defined by $\mathcal{X} := [m]^n$, $\mathcal{Y} := (\{0, 1\}^m)^n$, and $f \circ \textsc{Ind}_m^n(x, y) = \{o \in \mathcal{O} : o \in f(\textsc{Ind}_m^n(x, y))\}$.

Intuitively, each $x \in \mathcal{X}$ can be viewed as a block-respecting subset over the universe $[mn]$ where n elements are chosen, one from each block of size m. For each $i \in [n]$, to determine the value of the variable z_i in the original problem f, we restrict ourselves to the ith block of y and take the bit indexed by the ith coordinate of x.

Consider a search problem $f \subseteq \{0, 1\}^n \times \mathcal{O}$. A *decision tree* T is a binary tree such that each non-leaf node v is labeled with an input variable z_i, and each leaf v of Π is labeled with a solution

[1] This is known in the sunflower literature as the *link* of $\mathcal{F}$ at S, and is usually written there as $\mathcal{F}_S$, but for the sake of consistency we use the notation $\mathcal{F}_{\overline{S}}$. Hopefully this won't cause any confusion.

$o_v \in \mathcal{O}$. The tree T solves f if, for any input $z \in \{0,1\}^n$ the unique root-to-leaf path, generated by walking left at node v if the variable z_i that v is labeled with is 0 (and right otherwise), terminates at a leaf u with $o_u \in f(z)$. We define

$$\mathbf{P}^{dt}(f) \;\coloneqq\; \text{least } \textit{depth} \text{ of a decision tree solving } f.$$

Consider a bipartite search problem F. A *communication protocol* Π is a binary tree where now each non-leaf node v is labeled with a binary function $v(x,y)$ which depends only on either $\mathcal{X}$ or $\mathcal{Y}$ but not both. This is informally viewed as two players Alice and Bob jointly computing a function, where Alice receives $x \in \mathcal{X}$ and Bob receives $y \in \mathcal{Y}$, and where at each node in the protocol either Alice or Bob computes $v(x)$ or $v(y)$, respectively, and "speaks" as to which child to go to, depending on whose turn it is. The protocol Π solves F if, for any input $(x,y) \in \mathcal{X} \times \mathcal{Y}$ the unique root-to-leaf path, generated by walking left at node v if $v(x,y) = 1$ (and right otherwise), terminates at a leaf u with $o_u \in F(x,y)$. We define

$$\mathbf{P}^{cc}(F) \;\coloneqq\; \text{least } \textit{depth} \text{ of a communication protocol solving } F.$$

An alternative characterization of communication protocols, which will be useful for proving our main theorem, is as follows. Each non-leaf node v is labeled with a *(combinatorial) rectangle* $R_v = X_v \times Y_v \subseteq \mathcal{X} \times \mathcal{Y}$, such that if v_ℓ and v_r are the children of v, R_{v_ℓ} and R_{v_r} partition R_v. Furthermore this partition is either of the form $X_{v_\ell} \times Y_v \sqcup X_{v_r} \times Y_v$ or $X_v \times Y_{v_\ell} \sqcup X_v \times Y_{v_r}$. The unique root-to-leaf path on input (x,y) is generated by walking to whichever child v of the current node satisfies $(x,y) \in R_v$.

Sunflowers. Let $\mathcal{F}$ be a set system over some universe $\mathcal{S}$, and for a set $S \subseteq \mathcal{S}$ recall the definition of $\mathcal{F}_{\bar{S}}$. A (p,ϵ)-*approximate sunflower* is the set system $\mathcal{F}_{\bar{S}}$ such that

$$\mathbf{Pr}_{\boldsymbol{y} \subseteq_p \mathcal{S} \setminus S}(\forall \gamma \in \mathcal{F}_{\bar{S}} : \gamma \not\subseteq y) \leq \epsilon$$

where $\subseteq_p$ means that each element is added to y independently with probability p. For this paper we will always be using $p = 1/2$, and so for convenience we simply write $\boldsymbol{y} \subseteq \mathcal{S} \setminus S$ instead of $\subseteq_{1/2}$ and call $\mathcal{F}_{\bar{S}}$ an ϵ-approximate sunflower instead of an $(1/2,\epsilon)$-approximate sunflower. An analogue of the famed sunflower lemma of Erdős was proved for approximate sunflowers by Rossman [Ros10]:

Lemma 2.1 (Approximate Sunflower Lemma). *Let $s \in \mathbb{N}$ and let $\epsilon > 0$. Let $\mathcal{F}$ be a set system over $\mathcal{S}$ such that a) $|\gamma| \leq s$ for all $\gamma \in \mathcal{F}$; and b) $|\mathcal{F}| \geq (s \log 1/\epsilon)^s$. Then $\mathcal{F}$ contains an ϵ-approximate sunflower.*

A recent breakthrough result (proved in [ALWZ20] and simplified in [Rao19]) proves the sunflower lemma with significantly improved parameters. As a stepping stone they also prove an improvement on Approximate Sunflower Lemma assuming a condition called *spreadness*; a set system $\mathcal{F}$ over $\mathcal{S}$ is r-*spread* if $|\mathcal{F}_{\bar{S}}| \leq r^{|\mathcal{S}|-|S|}$ for every $S \subseteq \mathcal{S}$. Note that if the blockwise min-entropy of $\mathcal{F}$ is $\log r$, then that $|\mathcal{F}| \geq r^N$ by averaging, and $|\mathcal{F}_{\bar{S}}| \leq |\mathcal{F}| \cdot r^{-|S|}$ for all $S \subseteq \mathcal{S}$ by the definition of blockwise min-entropy. Therefore we can state Lemma 4 in [Rao19] in the following way.

Lemma 2.2 (Blockwise Approximate Sunflower Lemma). *Let $s \in \mathbb{N}$ and let $\epsilon > 0$. Let $\mathcal{F}$ be a set system over $\mathcal{S}$ such that a) $|\gamma| \leq s$ for all $\gamma \in \mathcal{F}$; and b) $\mathcal{F}$ has blockwise min-entropy at least $\log(K \log s/\epsilon)$ for some absolute constant K. Then $\mathcal{F}$ contains an ϵ-approximate sunflower. Furthermore the core of this sunflower is empty, and so by extension*

$$\mathbf{Pr}_{\boldsymbol{y} \subseteq \mathcal{S}}(\forall \gamma \in \mathcal{F} : \gamma \not\subseteq y) \leq \epsilon$$

In our main argument we will use a simple and general statement about the satisfiability of monotone CNFs in order to connect sunflowers to restrictions.

Claim 2.3. *Let $\mathcal{C} = C_1 \dots C_m$ be a CNF on the variables $x_1 \dots x_n$ such that no clause contains both the literals x_i and $\overline{x_i}$ for any i. Let $\mathcal{C}^{mon}$ be the result of replacing, for every i, every occurrence of $\overline{x_i}$ in $\mathcal{C}$ with x_i. Then*

$$|\{x \in \{0,1\}^n : \mathcal{C}(x) = 1\}| \leq |\{x \in \{0,1\}^n : \mathcal{C}^{mon}(x) = 1\}|$$

Proof. Let $\mathcal{C}^i$ be the result of replacing every occurrence of $\overline{x_i}$ in $\mathcal{C}$ with x_i. It is enough to show that for any i, $\mathcal{C}^i(x)$ is satisfied by at least as many assignments $\beta \in \{0,1\}^n$ to x as $\mathcal{C}(x)$ is, as we can then apply the argument inductively for $i = 1 \dots n$. Let $\beta^{-i} \in \{0,1\}^{[n] \setminus \{i\}}$ be an assignment to every variable except x_i. We claim that for every β^{-i}, $\mathcal{C}^i(\beta^{-i}, x_i)$ is satisfied by at least as many assignments $\beta_i \in \{0,1\}$ to x_i as $\mathcal{C}(\beta^{-i}, x_i)$.

Since there are no clauses with both x_i and $\overline{x_i}$, each clause in $\mathcal{C}$ is of the form $x_i \vee A$, $\overline{x_i} \vee B$, or C, where A, B, and C don't depend on x_i; the corresponding clauses in $\mathcal{C}^i$ are $x_i \vee A$, $x_i \vee B$, and C. If $\mathcal{C}^i(\beta^{-i}, 0) = 1$, then $A(\beta^{-i}) = B(\beta^{-i}) = C(\beta^{-i}) = 1$ for all A, B, and C, and so $\mathcal{C}^i(\beta^{-i}, x_i)$ is always satisfied. If $\mathcal{C}^i(\beta^{-i}, 1) = 0$, then it must be that $C(\beta^{-i}) = 0$ for some C, and so $\mathcal{C}(\beta^{-i}, x_i)$ has no satisfying assignments. Finally assume neither of these cases hold, and so $\mathcal{C}^i(\beta^{-i}, 0) = 0$ and $\mathcal{C}^i(\beta^{-i}, 1) = 1$. Then it must be that either $A(\beta^{-i}) = 0$ for some A, in which case $\mathcal{C}(\beta^{-i}, 0) = 0$, or $B(\beta^{-i}) = 0$ for some B, in which case $\mathcal{C}(\beta^{-i}, 1) = 0$. Therefore $\mathcal{C}(\beta^{-i}, x_i)$ has at least one falsifying assignment, while $\mathcal{C}^i(\beta^{-i}, x_i)$ has exactly one. □

3 The basic lifting theorem

The following determininistic lifting theorem was proven in [RM99], and more explicitly in [GPW15].

Lemma 3.1 (**Basic Lifting Theorem** [RM99, GPW15])**.** *Let f be a search problem over $\{0,1\}^n$, and let $m = n^{1.1}$. Then*

$$\mathbf{P}^{cc}(f \circ \textsc{Ind}_m^n) \;=\; \mathbf{P}^{dt}(f) \cdot \Theta(\log m)$$

To prove Basic Lifting Theorem, we prove that a) a decision tree of depth d for f can be simulated by a communication protocol of depth $O(d \log m)$ for the composed problem $f \circ \textsc{Ind}_m^n$, and b) a communication protocol of depth $d \log m$ for the composed problem $f \circ \textsc{Ind}_m^n$ can be simulated by a decision-tree of depth $O(d)$ for f. The forward direction is obvious: given a decision tree T for f, Alice and Bob can simply trace down T and compute the appropriate variable z_i at each node $v \in T$ visited, spending $\log m$ bits to compute $\textsc{Ind}_m(x_i, y_i)$ to do so. Thus we focus on simulating a communication protocol Π of depth $d \log m$. Let $\{z_i\}_i$ be the variables of f and let $\{x_i\}_i, \{y_i\}_i$ be the variables of $f \circ \textsc{Ind}_m^n$. Recall that each z_i takes values in $\{0,1\}$, x_i takes values in $[m]$, and y_i takes values in $\{0,1\}^m$.

Our proof will follow the basic structure of previous works [GPW17, GGKS18]. We first define a procedure, called the rectangle partition, which forms the main technical tool in our simulation. We then prove that with this tool and a few useful facts about its output, we can efficiently simulate the protocol Π by a decision tree T, using a number of invariants to show the efficiency and correctness of T. We give a complete and self-contained proof as well as some high-level intuition.

Where our proof differs is in the proof of one crucial lemma about the rectangle partition. Before we begin, we prove a very useful lemma that shows that when $\boldsymbol{X}$ has high blockwise min-entropy and $\boldsymbol{Y}$ has low deficiency, then it's possible to find an $x^* \in X$ such that the full image of the index gadget is available to x^*, or in other words $\textsc{Ind}_m^N(x^*, Y) = \{0,1\}^N$. This appears as Lemma 7 in

[GGKS18] for dag-like lifting and is stronger than is necessary for proving Basic Lifting Theorem, but the proof highlights our new counting strategy and will be a useful tool throughout the rest of the paper. While the original proof used tools from Fourier analysis, our proof is a simple application of Blockwise Approximate Sunflower Lemma.[2]

Lemma 3.2 (Full Range Lemma). *Let $m \geq n^{1.1}$ and let $N \leq n$. Let $X \times Y \subseteq [m]^N \times (\{0,1\}^m)^N$ be such that $\boldsymbol{X}$ has blockwise min-entropy at least $0.95 \log m - O(1)$ and $|Y| > 2^{mN - n \log m}$. Then there exists an $x^* \in X$ such that for every $\beta \in \{0,1\}^N$, there exists a $y_\beta \in Y$ such that $\textsc{Ind}_m^N(x^*, y_\beta) = \beta$.*

Proof. Assume for contradiction that for all x there exists a β_x such that $|\{y \in Y : y[x] = \beta_x\}| = 0$, or in other words for all $(x, y) \in X \times Y$, $y[x] \neq \beta_x$. Consider the CNF over $y_1 \dots y_{mN}$ where clause C_x is the clause uniquely falsified by $y[x] = \beta_x$; then by Claim 2.3 we see that $|\{y \in (\{0,1\}^m)^N : \forall x, y[x] \neq \beta_x\}|$ is maximized when $\beta_x = 1^N$. Thus because $Y \subseteq (\{0,1\}^m)^N$,

$$|\{y \in Y : \forall x, y[x] \neq \beta_x\}| \leq |\{y \in (\{0,1\}^m)^N : \forall j, y[x] \neq 1^N\}|$$

By the fact that $m^{0.95} \gg O(n \log m)$ and $N \leq n$, $\boldsymbol{X}$ has blockwise min-entropy $0.95 \log m - O(1) > \log(Kn \log m)) \geq \log(K \log(N/\epsilon))$, where $\epsilon := 2^{-n \log m}$ and K is the constant given by Blockwise Approximate Sunflower Lemma. Thus we can apply Blockwise Approximate Sunflower Lemma to X and get that $\mathbf{Pr}_{\boldsymbol{y} \subseteq [mN]}(\forall x \in X, x \not\subseteq y) \leq \epsilon$, and so

$$|Y| \leq \{y \in (\{0,1\}^m)^N : \forall j, y[x] \neq 1^N\}| \leq \epsilon \cdot 2^{mN} = 2^{mN - n \log m}$$

which is a contradiction as $|Y| > 2^{mN - n \log m}$ by assumption. □

3.1 Density-restoring partition

Before going into the simulation, we define our essential tool, which is usually called the *density-restoring partition* or *rectangle partition* as per [GPW17]. Let $N \leq n$ and let $X \times Y \subseteq [m]^N \times (\{0,1\}^m)^N$. Our goal will be to output a set of rectangles $X^j \times Y^{j,\beta}$ which cover most of $X \times Y$ such that each $X^j \times Y^{j,\beta}$ is "good" in a similar sense to the statement of Full Range Lemma. More specifically, for each $X^j \times Y^{j,\beta}$ there is some set of coordinates $J \subseteq [N]$ such that X and Y are completely fixed on J and "very unfixed" on $[N] \setminus J$. For X this means high blockwise min-entropy of $\boldsymbol{X}_{\bar{J}}$, meaning that every joint setting of some set of free coordinates is roughly equally likely. For Y the universe $(\{0,1\}^m)^N$ is so large in comparison to $[m]^N$ that a lower bound on $|Y_{\bar{J}}^{j,\beta}|$ is enough to assure $Y^{j,\beta}$ is free enough in the unfixed coordinates.

Definition 3.1. Let $N \leq n$ and let $\rho \in \{0,1,*\}^N$ be a partial assignment with $J := \text{fix}(\rho) \subseteq [N]$. A rectangle $R = X \times Y \subseteq [m]^N \times (\{0,1\}^m)^N$ is *ρ-structured* if the following conditions hold:

- X and Y are fixed on the blocks J and $\textsc{Ind}_m^J(X_J, Y_J) = \rho[J]$
- $\boldsymbol{X}_{\bar{J}}$ has blockwise min-entropy at least $0.95 \log m$
- $|Y_{\bar{J}}| > 2^{m \cdot |\bar{J}| - n \log m}$

To perform the partition we will need to find the sets $X^j \times Y^{j,\beta}$ along with a corresponding assignment $\rho^{j,\beta}$ for which they are $\rho^{j,\beta}$-structured. This is done in two phases. Our goal in Phase I will be to break up X into disjoint parts X^j, such that each X^j is fixed on some set $I_j \subseteq [N]$ and

[2] While we simplify things in this section by using $m = n^{1.1}$, our improved gadget size (see Section 4) crucially uses the improvements in Blockwise Approximate Sunflower Lemma over the basic Approximate Sunflower Lemma; the same improvements also give us a very short proof of our main lemma. However, these improvements aren't strictly necessary for our techniques; in Section 6 we provide an alternate proof just using Approximate Sunflower Lemma.

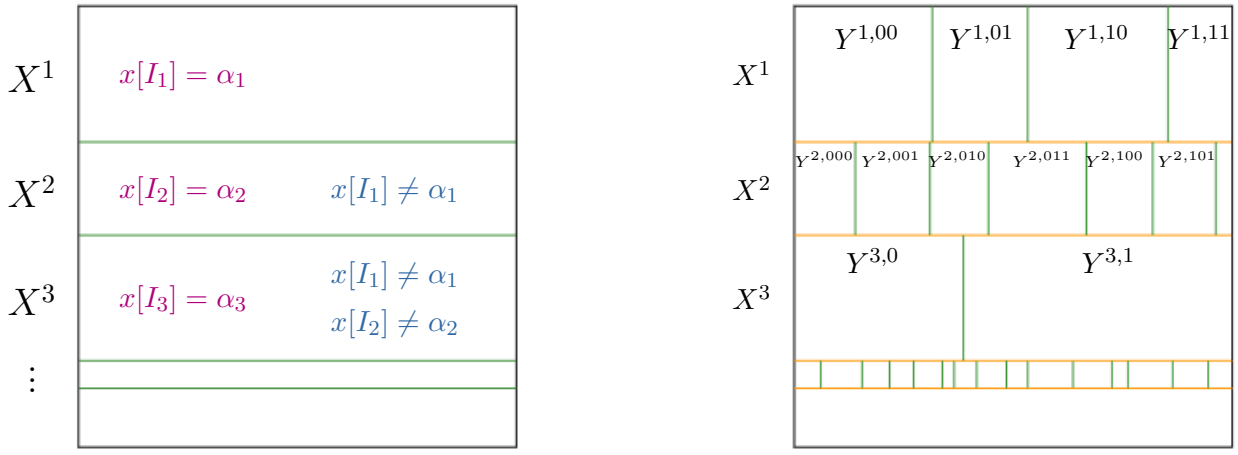

Figure 1: Phases I and II of Rectangle Partition. In each $X^j \times Y^{j,\beta}$, $x[I_j]$ is fixed to α_j and $y[I_j]$ is fixed so that $\textsc{Ind}_m^{I_j}(X^j_{I_j}, Y^{j,\beta}_{I_j}) = \beta$.

has blockwise min-entropy $0.95 \log m$ on $\overline{I_j}$—hence this partition is "density-restoring" when $\boldsymbol{X}$ starts off with blockwise min-entropy below $0.95 \log m$. To do this, the procedure iteratively finds a maximal partial assigment (I_j, α_j) such that the assignment $x[I_j] = \alpha_j$ violates $0.95 \log m$ blockwise min-entropy in $\boldsymbol{X}$, splits the remaining X into the part X^j satisfying this assignment and the part $X \setminus X^j$ not satisfying it, and recurses on the latter part. We do this until we've covered at least half of X by X^j subsets.

Our goal in Phase II will be to break up Y into disjoint parts $Y^{j,\beta}$ for each X^j from Phase I, such that each $X^j \times Y^{j,\beta}$ is $\rho^{j,\beta}$-structured for some restriction $\rho^{j,\beta}$. We already have the blockwise min-entropy of X^j in the coordinates $[N] \setminus I_j$ by our first goal, so clearly $\mathrm{fix}(\rho^{j,\beta}) = I_j$ for any k. Thus we need to fix the coordinates of Y within the blocks I_j, and within each $Y^{j,\beta}$ it should be the case that $y[I_j, \alpha_j] = \beta$ for all $y \in Y^{j,\beta}$, at which point $\rho^{j,\beta}$ can be fixed to β on I_j and left free everywhere else.

Algorithm 1: Rectangle Partition

Initialize $\mathcal{F} = \emptyset$, $j = 1$, and $X^{\geq 1} := X$;
PHASE I (X^j): while $|X^{\geq j}| \geq |X|/2$ **do**
  Let I_j be a maximal (possibly empty) subset of $[N]$ such that $\boldsymbol{X}^{\geq j}$ violates $0.95 \log m$-blockwise min-entropy on I_j, and let $\alpha_j \in [m]^{I_j}$ be an outcome witnessing this: $\mathbf{Pr}_{x \sim \boldsymbol{X}^{\geq j}}(x[I_j] = \alpha_j) > 2^{0.95|I_j| \log m}$;
  Define $X^j := \{x \in X^{\geq j} : x[I_j] = \alpha_j\}$;
  Update $\mathcal{F} \leftarrow \mathcal{F} \cup \{(I_j, \alpha_j)\}$, $X^{\geq j+1} := X^{\geq j} \setminus X^j$, and $j \leftarrow j + 1$;
end
PHASE II ($Y^{j,\beta}$): for j, $\beta \in \{0,1\}^{I_j}$ **do**
  Let $Y' = \{y \in Y : y[I_j, \alpha_j] = \beta\}$, and let $\eta^{j,\beta} \in (\{0,1\}^m)^{|I_j|}$ be the string which maximizes $|\{y \in Y' : y[I_j] = \eta^{j,\beta}\}|$;
  Define $Y^{j,\beta} := \{y \in Y : y[I_j] = \eta^{j,\beta}\}$;
end
return $\mathcal{F}$, $\{X^j\}_j$, $\{Y^{j,\beta}\}_{j;\beta}$;

Our algorithm is formally described in Rectangle Partition.[3] Let $X \subseteq [m]^N$, let $Y \subseteq (\{0,1\}^m)^N$, and let $\mathcal{F}$, $\{X^j\}_j$, $\{Y^{j,\beta}\}_{j;\beta}$ be the outputs of the rectangle partition on $X \times Y$. Recall that our goal was to break $X \times Y$ up into $\rho^{j,\beta}$-structured rectangles $X^j \times Y^{j,\beta}$; the following simple claims show that the obvious choice of $\rho^{j,\beta}$ achieves two of the three conditions needed.

Claim 3.3. *For all j and for all $\beta \in \{0,1\}^{I_j}$, define $\rho^{j,\beta} \in \{0,1,*\}^N$ to be the restriction where $\mathrm{fix}(\rho^{j,\beta}) = I_j$ and $\rho^{j,\beta}[I_j] = \beta$. Then $X^j \times Y^{j,\beta}$ is fixed on I_j and outputs $\textsc{Ind}_m^{I_j}(X^j_{I_j}, Y^{j,\beta}_{I_j}) = \rho^{j,\beta}[I_j]$.*

Proof. By definition X^j is fixed to α_j on the coordinates I_j, while $Y^{j,\beta}$ is fixed to $\eta^{j,\beta}$ on the blocks I_j. Since $\eta^{j,\beta} \in \{0,1\}^{I_j}$ clearly satisfies $\eta^{j,\beta}[\alpha_j] = \beta$, it holds that $\textsc{Ind}_m^{I_j}(X^j_{I_j}, Y^{j,\beta}_{I_j}) = \beta = \rho^{j,\beta}[I_j]$. □

Claim 3.4. *For all j, $\boldsymbol{X}^j_{\bar{I}_j}$ has blockwise min-entropy at least $0.95 \log m$.*

Proof. Assume for contradiction that $I^* \subseteq [N] \setminus I_j$ such that $\boldsymbol{X}^j$ violates $0.95 \log m$-blockwise min-entropy on I^*, and let α^* be an outcome witnessing this. Then

$$\begin{aligned} \mathbf{Pr}_{x \sim \boldsymbol{X}^{\geq j}}(x[I_j] = \alpha_j \wedge x[I^*] = \alpha^*) &> 2^{-0.95|I_j| \log m} \cdot \mathbf{Pr}_{x \sim \boldsymbol{X}^j}(x[I^*] = \alpha^*) \\ &> 2^{-0.95|I_j| \log m - 0.95|I^*| \log m} = 2^{-0.95|I_j \cup I^*| \log m} \end{aligned}$$

which contradicts the maximality of I_j. □

Before moving to the third condition, the size of $Y^{j,\beta}_{\bar{I}_j}$, we show that the deficiency of each $\boldsymbol{X}^j$ drops by $\Omega(|I_j| \log m)$. This will be used later to show the efficiency of our simulation.

Claim 3.5. *For all $(I_j, \alpha_j) \in \mathcal{F}$, $\mathbf{D}_\infty(\boldsymbol{X}^j_{\bar{I}_j}) \leq \mathbf{D}_\infty(\boldsymbol{X}) - 0.05|I_j| \log m + 1$.*

Proof. By our choice of (I_j, α_j) it must be that $|X^j| = |X^{\geq j}| \cdot \mathbf{Pr}_{x \sim \boldsymbol{X}^{\geq j}}(x[I_j] = \alpha_j) \leq |X^{\geq j}| \cdot 2^{-0.95 \log m}$. Then by a simple calculation

$$\begin{aligned} \mathbf{D}_\infty(\boldsymbol{X}^j_{\bar{I}_j}) &= |\bar{I}_j| \log m - \log |X^j| \\ &\leq (N - |I_j|) \log m - \log(|X^{\geq j}| \cdot 2^{-0.95|I_j| \log m}) \\ &\leq (N \log m - |I_j| \log m) - \log |X^{\geq j}| + 0.95|I_j| \log m - \log |X| + \log |X| \\ &= (N \log m - \log |X|) - 0.05|I_j| \log m + \log(|X|/|X^{\geq j}|) \\ &\leq \mathbf{D}_\infty(\boldsymbol{X}) - 0.05|I_j| \log m + 1 \end{aligned}$$

where the last step used the fact that $|X^{\geq j}| \geq |X|/2$, since we terminate as soon as $|X^{\geq j}| < |X|/2$ at the start of the jth iteration. □

For our last lemma before going into the simulation, instead of showing that $|Y^{j,\beta}_{\bar{I}_j}| = |Y^{j,\beta}|$ is large for *every* j and every β, we want to show that every $|Y^{j,\beta}|$ is large for *some* j and every β. If every β were equally likely then $|Y^{j,\beta}| \approx |Y|/2^{m \cdot |I_j|}$; for us it is enough that the smallest $Y^{j,\beta}$ be a factor of $2^{-O(|I_j| \log m)}$ away from this. We add two new assumptions on $X \times Y$: 1) $\boldsymbol{X}$ starts with blockwise min-entropy very close to $0.95 \log m$; and 2) Y is initially large. For convenience we redefine X to only be the union of the X^j parts; since we terminate after $|X^{\geq j}| < |X|/2$ we can do this and only decrease the blockwise min-entropy of $\boldsymbol{X}$ by 1. This lemma is new and is a fairly direct application of Full Range Lemma, which only used the sunflower lemma.

[3] For those familiar with previous works [GPW17], Rectangle Partition varies in two ways: 1) we truncate Phase I once we've partitioned at least half of X; and 2) in Phase II we fix the rest of $Y^{j,\beta}$ inside the blocks I_j.

Lemma 3.6. *Let $X := \cup_j X^j$ be such that $\boldsymbol{X}$ has blockwise min-entropy $0.95 \log m - O(1)$, and let Y be such that $|Y| > 2^{mN - n \log m + 1}$. Then there is a j such that for all $\beta \in \{0,1\}^{I_j}$,*

$$|Y^{j,\beta}_{\overline{I_j}}| \geq |Y|/2^{m \cdot |I_j| + 2|I_j| \log m}$$

Proof. We will show that there exists a j such that for every $\beta \in \{0,1\}^{I_j}$, $|\{y \in Y : y[I_j, \alpha_j] = \beta\}| \geq |Y|/2^{2|I_j| \log m}$. If this is true, then by averaging there is some assignment to I_j—aka $\eta^{j,\beta}$—such that

$$|Y^{j,\beta}_{\overline{I_j}}| = |Y^{j,\beta}| \geq (|Y|/2^{2|I_j| \log m})/2^{m \cdot |I_j|} = |Y|/2^{m \cdot |I_j| + 2|I_j| \log m}$$

Assume for contradiction that for every j there exists a β_j such that $|\{y \in Y : y[I_j, \alpha_j] = \beta_j\}| < |Y|/2^{2|I_j| \log m}$. Define $Y_= := \{y \in Y : \exists j, y[I_j, \alpha_j] = \beta_j\}$. We will show that $|Y_=| < |Y|/2$. If this is the case then for $Y_{\neq} := Y \setminus Y_=$, it must be that $|Y_{\neq}| \geq |Y|/2 > 2^{mN - n \log m}$. By Full Range Lemma there must exist some $x^* \in X$ such that for every $\beta \in \{0,1\}^N$ there exists $y_\beta \in Y_{\neq}$ such that $y_\beta[x^*] = \beta$. Since $x^* \in X$, $x^* \in X^j$, for some j, and so for any $\beta \in \{0,1\}^N$ such that $\beta[I_j] = \beta_j$, there exists a $y_\beta \in Y_{\neq}$ such that $y_\beta[x^*] = \beta$. But since $x^* \in X^j$, $x^*[I_j] = \alpha_j$, so $y_\beta[I_j, \alpha_j] = \beta_j$ which is a contradiction since $Y_{\neq} = \{y \in Y : \forall j, y[I_j, \alpha_j] \neq \beta_j\}$.

We now show that $|Y_=| < |Y|/2$. Define $\mathcal{F}(k) := \{(I_j, \alpha_j) \in \mathcal{F} : |I_j| = k\}$. Clearly $|\mathcal{F}(k)| \leq 2^{1.9k \log m - 1}$ since there are at most $\binom{N}{k} \ll 2^{0.9k \log m - 1}$ possible sets I_j, and for each there are m^k possible assignments α_j. Furthermore $\mathcal{F}(0)$ must be empty, because if the empty restriction where $I_j = \emptyset$ is in $\mathcal{F}$, then the corresponding β_j would be empty and $\{y \in Y : y[\emptyset, \emptyset] = \emptyset\} = Y$ would be of size $|Y|/2^0$, which contradicts our choice of β_j. Since we assumed that $|\{y \in Y : y[I_j, \alpha_j] = \beta\}| < |Y|/2^{2|I_j| \log m}$ for all j, by our bound on $|\mathcal{F}(k)|$

$$\begin{aligned}
|Y_=| &< \sum_{k=1}^{N} (2^{1.9k \log m - 1} \cdot \frac{|Y|}{2^{2k \log m}}) \\
&\leq \frac{|Y|}{2} \cdot \sum_{k=1}^{N} (2^{0.1 \cdot \log m})^{-k} \\
&< \frac{|Y|}{2} \cdot \sum_{k=1}^{\infty} 2^{-k} = \frac{|Y|}{2}
\end{aligned}$$

which completes the proof. □

3.2 Simulation

Proof of Basic Lifting Theorem. To recall, we start with a protocol Π of depth $d \log m$ for the composed problem $f \circ \textsc{Ind}^n_m$ and want to construct a decision-tree of depth $O(d)$ for f. Note that we can assume that $d = o(n)$ as the theorem is trivial otherwise. The decision-tree is naturally constructed by starting at the root of Π and taking a walk down the protocol tree guided by occasional queries to the variables $z = (z_1, \ldots, z_n)$ of f. During the walk, we maintain a ρ-structured rectangle $R = X \times Y \subseteq [m]^n \times (\{0,1\}^m)^n$ which will be a subset of the inputs that reach the current node in the protocol tree, where ρ corresponds to the restriction induced by the decision tree at the current step. Thus our goal is to ensure that the image $\textsc{Ind}^n_m(X \times Y)$ has some of its bits fixed according to the queries to z made so far, and no information has been leaked about the remaining free bits of z.

To choose which bits to fix, we use the density restoring partition to identify any assignments to some of the x variables that have occurred with too high a probability; by the way the rectangle partition is defined the corresponding sets X^j regain blockwise min-entropy. Then using Lemma 3.6, we pick one of these assignments and query all the corresponding z variables, and for the resulting

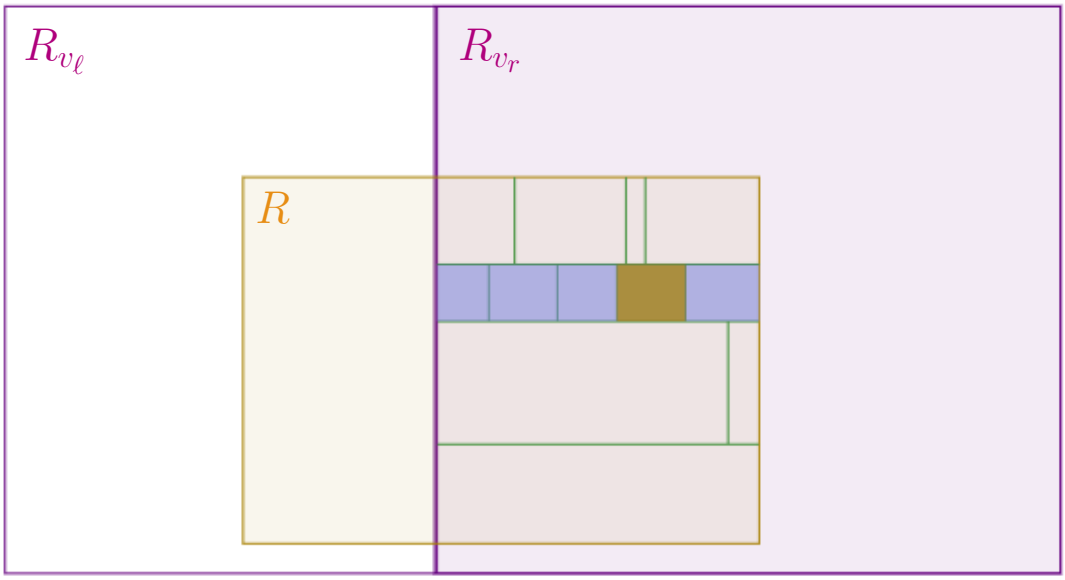

Figure 2: One iteration of Simulation Protocol. We perform Rectangle Partition (green lines) on the larger half of R after moving from v to its child (shaded in purple), use Lemma 3.6 to identify a part j (shaded in blue), and then query I_j and set R to $X^j \times Y^{j,\beta}$ for the result $z[I_j] = \beta$ (shaded in brown).

β we know $X^j \times Y^{j,\beta}$ is $\rho^{j,\beta}$-structured since the size of $Y^{j,\beta}$ doesn't decrease too much. With the blockwise min-entropy of $\boldsymbol{X}$ restored and the size of Y kept high, we can update ρ to include $\rho^{j,\beta}$ and continue to run the rectangle partition at the next node, and so we proceed in this way down the whole communication protocol.

We describe our query simulation of the communication protocol Π in Simulation Protocol. For all $v \in \Pi$ let $R_v = X_v \times Y_v$ be the rectangle induced at node v by the protocol Π. The query and output actions listed in bold are the ones performed by our decision tree.

Algorithm 2: Simulation protocol

Initialize $v :=$ root of Π; $R := [m]^n \times (\{0,1\}^m)^n$; $\rho = *^n$;
while *v is not a leaf* **do**
  Precondition: $R = X \times Y$ is ρ-structured; for convenience define $J := \mathrm{fix}(\rho)$;
  Let v_ℓ, v_r be the children of v, and update $v \leftarrow v_\ell$ if $|R \cap R_{v_\ell}| \geq |R|/2$ and $v \leftarrow v_r$ otherwise;
  Execute Rectangle Partition on $(X \cap X_v)_{\bar{J}} \times (Y \cap Y_v)_{\bar{J}}$ and let $\mathcal{F} = \{(I_j, \alpha_j)\}_j$, $\{X^j\}_j$, $\{Y^{j,\beta}\}_{j;\beta}$ be the outputs;
  Apply Lemma 3.6 to $\mathcal{F}$, $\{X^j\}_j$, $\{Y^{j,\beta}\}_{j;\beta}$ to get some index j corresponding to $(I_j, \alpha_j) \in \mathcal{F}$;
  Query each variable z_i for every $i \in I_j$, and let $\beta \in \{0,1\}^{I_j}$ be the result;
  Update $X_{\bar{J}} \leftarrow X^j$ and $Y_{\bar{J}} \leftarrow Y^{j,\beta}$, and update $\rho \leftarrow \rho \cup \rho^{j,\beta}$ (recall that $\rho^{j,\beta} \in \{0,1,*\}^n$ is the restriction where $\mathrm{fix}(\rho^{j,\beta}) = I_j$ and $\rho^{j,\beta}[I_j] = \beta$);
end
Output the same value as v does;

Before we prove the correctness and efficiency of our algorithm, we note that we make no

distinction between Alice speaking and Bob speaking in our procedure. Here we note that each R_v is a rectangle induced by the protocol Π, and so updating v only splits X or Y—corresponding to when Alice and Bob speak respectively—but not both, and so since $R \subseteq R_v$ we get that $|X \cap X_v| \geq |X|/2$ and $|Y \cap Y_v| \geq |Y|/2$.

Efficiency and correctness. To prove the efficiency and correctness of our algorithm, consider the start of the tth iteration, where we are at a node v and maintaining $R^t = X^t \times Y^t$ and ρ^t.[4] Again for convenience we write $J^t := \mathrm{fix}(\rho^t)$. Let (I^t, α^t) be the (possibly empty) assignment returned by Lemma 3.6 corresponding to index j^t, and let β^t be the result of querying $z[I^t]$. Note that $J^{t+1} = J^t \sqcup I^t$, $X^{t+1}_{\overline{J^{t+1}}} = X^{j^t}_{\overline{I^t}}$, and $Y^{t+1}_{\overline{J^{t+1}}} = Y^{j^t,\beta^t}_{\overline{I^t}}$. Also note that $t \leq d \log m$ by the depth of the protocol Π.

We show that our precondition that R^t is ρ^t-structured holds for all t, assuming for the moment that that $|J^t| \leq O(d)$ for all t. We show this by the following three invariants:

(i) X^t, Y^t are fixed on J^t and $\textsc{Ind}^{J^t}_m(X^t_{J^t}, Y^t_{J^t}) = \rho^t[J^t]$
(ii) $\boldsymbol{X}^t_{\overline{J^t}}$ has blockwise min-entropy at least $0.95 \log m$
(iii) $|Y^t_{\overline{J^t}}| \geq 2^{m \cdot |\overline{J^t}| - t - 2|J^t| \log m}$.

This is enough to show R^t is ρ^t structured as $2^{m \cdot |\overline{J^t}| - t - 2|J^t| \log m} > 2^{m \cdot |\overline{J^t}| - n \log m}$ by assumption on $|J^t|$. All invariants hold at the start of the algorithm since $\rho^0 = *^n$ and $X^0 \times Y^0 = [m]^n \times (\{0,1\}^m)^n$. Inductively consider the $t+1$st iteration assuming all invariants holds for the tth iteration. After applying Rectangle Partition invariant *(i)* follows by Claim 3.3 and invariant *(ii)* follows by Claim 3.4. For invariant *(iii)* we first show that it is valid to apply Lemma 3.6 in the $t+1$st iteration. First, because $|X^t \cap X_v| \geq |X^t|/2$ we know that the blockwise min-entropy of $(\boldsymbol{X}^t \cap \boldsymbol{X}_v)_{\overline{J^t}}$ is at most one less than the blockwise min-entropy of $\boldsymbol{X}^t_{\overline{J^t}}$, which is at least $0.95 \log m$. Second, we have $|(Y^t \cap Y_v)_{\overline{J^t}}| \geq |Y^t|/2 > 2^{m \cdot |\overline{J^t}| - t - 2|J^t| \log m - 1} > 2^{m \cdot |\overline{J^t}| - O(d \log m)}$, and recall that $d = o(n)$. Thus we can apply Lemma 3.6 and we get

$$\begin{aligned}
|Y^{t+1}_{\overline{J^{t+1}}}| &= |Y^{j^t,\beta^t}_{\overline{I^t}}| \\
&\geq |(Y^t \cap Y_v)_{\overline{J^t}}|/2^{m \cdot |I^t| + 2|I^t| \log m} \\
&\geq 2^{m \cdot |\overline{J^t}| - t - 2|J^t| \log m - 1}/2^{m \cdot |I^t| + 2|I^t| \log m} \\
&\geq 2^{m \cdot (|\overline{J^t}| - |I_t|) - t - 1 - 2(|J^t| + I^t|) \log m} = 2^{m \cdot |\overline{J^{t+1}}| - (t-1) - 2|J^{t+1}| \log m}
\end{aligned}$$

To show that $|J^t| \leq O(d)$—and by extension that our simulation is guaranteed to be efficient—it is enough to show that $\mathbf{D}_\infty(\boldsymbol{X}^t_{\overline{J^t}}) \leq 2t - 0.05|J^t| \log m$ for every $t \leq d \log m$, as this gives a bound of $|J^t| \leq 2t/0.05 \log m = O(d)$ by the non-negativity of deficiency. When $t = 0$ then J^t is empty and $\mathbf{D}_\infty(\boldsymbol{X}) = 0$. Now in the tth iteration recall that we query the set I^t, and by Claim 3.5 we get that

$$\begin{aligned}
\mathbf{D}_\infty(\boldsymbol{X}^{t+1}_{\overline{J^{t+1}}}) &= \mathbf{D}_\infty(\boldsymbol{X}^{j^t}_{\overline{I^t}}) \\
&\leq \mathbf{D}_\infty((\boldsymbol{X} \cap \boldsymbol{X}_v)_{\overline{J^t}}) - 0.05|I^t| \log m + 1 \\
&\leq 1 + (2t - 0.05|J^t| \log m) - 0.05|I^t| \log m + 1 = 2(t+1) - 0.05|J^{t+1}| \log m
\end{aligned}$$

We finally have to argue that if we reach a leaf v of Π while maintaining R and ρ, then the solution $o \in \mathcal{O}$ output by Π is also valid solution to the values of z, of which the decision-tree knows that

[4] We understand that this notation is somewhat overloaded with X^j, $Y^{j,\beta}$, and $\rho^{j,\beta}$. Since the proof that the invariants hold is short and we only ever use t (or $t+1$) for the time stamps and j for the indices, hopefully this won't cause any confusion.

$z[\text{fix}(\rho)] = \rho[\text{fix}(\rho)]$. Suppose Π outputs $o \in \mathcal{O}$ at the leaf v, and assume for contradiction that there exists $\beta \in \{0,1\}^n$ consistent with ρ such that $\beta \notin f^{-1}(o)$. Since $\textsc{Ind}_m^{\text{fix}(\rho)}(x,y) = \rho[\text{fix}(\rho)] = \beta[\text{fix}(\rho)]$ for all $(x,y) \in R$, we focus on $\text{free}(\rho)$, and let $N := |\text{free}(\rho)|$. Since R is ρ-structured, $\boldsymbol{X}_{\text{free}(\rho)}$ has blockwise min-entropy $0.95 \log m$ and $|Y_{\text{free}(\rho)}| > 2^{m \cdot |\text{free}(\rho)| - n \log m}$. Thus applying Full Range Lemma, we know that that there exists $(x,y) \in R$ such that $\textsc{Ind}_m^n(x,y) = \beta$, which is a contradiction as $R \subseteq R_v \subseteq (f \circ \textsc{Ind}_m^n)^{-1}(o)$. □

4 Optimizing the gadget size

In Section 3 we loosely chose $m = n^{1.1}$ for the purpose of showing the basic lifting statement. In this section we improve from $n^{1.1}$; more specifically we show a tradeoff between the gadget size and the strength of the lifting theorem. Ultimately our tradeoff gives an optimal gadget size of m being quasilinear in n.

Theorem 4.1 (Improved Lifting Theorem). *Let f be a search problem over $\{0,1\}^n$, and let $m = \Omega(n \log n)$. Then*

$$\mathbf{P}^{cc}(f \circ \textsc{Ind}_m) \;\geq\; \Omega(\mathbf{P}^{dt}(f))$$

Warm-up: $m = n^{1+\epsilon}$. First we improve on Basic Lifting Theorem to get a gadget of size $n^{1+\epsilon}$ for any $\epsilon > 0$, with no changes in the asymptotic strength of the lifting theorem nor anything non-trivial in the proof. This comes from two observations. First, we only use the size of m in the two places we apply Full Range Lemma, and in both cases we can apply Blockwise Approximate Sunflower Lemma as long as $2^{0.95 \log m - O(1)} \geq \Omega(n \log m)$. Second, from the perspective of our simulation, the constant 0.95 is only used to set the blockwise min-entropy threshold for the density-restoring partition, and was chosen arbitrarily.

So for $\delta > 0$ we can instead choose to put the threshold at $(1-\delta)\log m$, at which point our condition on m changes to $m^{1-\delta} \geq \Omega(n \log m)$. Clearly this can be made to fulfill our condition $m \geq n^{1+\epsilon}$ with an appropriate choice of δ. The proof itself then simply becomes a matter of replacing 0.95 with $1-\delta$ and 0.05 with δ throughout the proof, as well as a few other constants. Since Claim 3.5 now gives a drop in deficiency of δ for every coordinate we query, the non-negativity of deficiency gives us $|\text{fix}(\rho^t)| \leq 2t/\delta \log m$ at any time $t \leq d \log m$, which gives us a decision tree of depth $(2/\delta) \cdot d = O(d)$ as required.

Optimal gadget: $m = \Theta(n \log n)$. Building off the intuition from our warm-up, what happens if δ is chosen to be subconstant? We cannot hope to get a tight lifting theorem, as our decision tree will be of depth $(2/\delta) \cdot d$, and choosing $\delta = o(1/\log m)$ makes our blockwise min-entropy threshold $(1-\delta)\log m$ trivial, as $\log m$ is the maximum possible blockwise min-entropy for $\boldsymbol{X}$. However it turns out there are no other constraints to worry about, and so we can get the following general lower bound, which gives Improved Basic Lifting Theorem as a special case.

Theorem 4.2 (Scaling Basic Lifting Theorem). *Let f be a search problem over $\{0,1\}^n$, and let m, δ be such that $\delta \geq \Omega(\frac{1}{\log m})$ and $m^{1-\delta} \geq \Omega(n \log m)$. Then*

$$\mathbf{P}^{cc}(f \circ \textsc{Ind}_m) \;\geq\; \mathbf{P}^{dt}(f) \cdot \Omega(\delta \log m)$$

Proof sketch. We start with a given communication protocol Π of depth $d \cdot \delta \log m$ for the composed problem $f \circ \textsc{Ind}_m^n$ and construct a decision-tree of depth $O(d)$ for f. We define a ρ-structured rectangle R as before except now with the condition that $\boldsymbol{X}$ has blockwise min-entropy $(1-\delta)\log m$.

Then in Rectangle Partition we set the blockwise min-entropy threshold for a violating assignment (I_j, α_j) at $(1-\delta)\log m$ as well.

To prove Full Range Lemma, note that we can apply Claim 2.3 regardless of m and N, and we can still apply Blockwise Approximate Sunflower Lemma as long as we can choose m such that $(1-\delta)\log m - 2 > \log(Kn\log m)$. Thus for this altered rectangle partition procedure, by the same proofs as before, Claim 3.4 states that $\boldsymbol{X}^j_{\bar{I}_j}$ has blockwise min-entropy at least $(1-\delta)\log m$, Claim 3.5 states that $\mathbf{D}_\infty(\boldsymbol{X}^j_{\bar{I}_j}) \leq \mathbf{D}_\infty(\boldsymbol{X}) - |I_j| \cdot \delta \log m + 1$, and Lemma 3.6 states that if $\boldsymbol{X}$ has blockwise min-entropy $(1-\delta)\log m - 2$ and $|Y| > 2^{mN - n\log m}$, then there exists a j such that for all β, $|Y^{j,\beta}| \geq |Y|/2^{m\cdot|I_j| + 2|I_j|\log m}$.

Now our simulation procedure is the same as Simulation Protocol. Again at the start of the tth iteration we are maintaining $R^t = X^t \times Y^t$, ρ^t, and $J^t := \mathrm{fix}(\rho^t)$, where now $t \leq d \cdot \delta \log m$. By the same argument our procedure is well-defined as long as the precondition of R^t being ρ^t-structured holds, and by a deficiency argument using our new Claim 3.5 we get that $\mathbf{D}_\infty(\boldsymbol{X}^t_{\bar{J}^t}) \leq 2t - |J^t| \cdot \delta \log m$, which implies $|J^t| \leq 2t/\delta \log m \leq 2d$. Our precondition holds by applying the new versions of Claim 3.3, Claim 3.4, and Lemma 3.6 as before. Finally our simulation is correct again by the invariants and Full Range Lemma. □

5 Lifting for dag-like protocols

In this section we show that we can perform our lifting theorem in the dag-like model, going from decision dags to communication dags. This was originally proven by Garg et al. [GGKS18], who required two main lemmas. The first is Full Range Lemma, and the second is the following lemma.

Lemma 5.1 (**Rectangle Lemma**, [GGKS18])**.** *Given a rectangle $R \subseteq [m]^n \times \{0,1\}^{mn}$, let $\{X^j \times Y^{j,\beta}\}_{j;\beta}$ be the output of Rectangle Partition on R. Then there exist sets $X_{err} \subseteq X$ and $Y_{err} \subseteq Y$, both of which have density $2^{-2d\log m}$ in $[m]^n$ and $(\{0,1\}^m)^n$ respectively, such that for each j, β one of the following holds:*

- **structured**: *$X^j \times Y^{j,\beta}$ is $\rho^{j,\beta}$ structured for some $\rho^{j,\beta}$ of width at most $O(d)$*
- **error**: *$X^j \times Y^{j,\beta} \subseteq X_{err} \times \{0,1\}^{mn} \cup [m]^n \times Y_{err}$*

Finally, a query alignment property holds: for every $x \in [m]^n \setminus X_{err}$ there exists a subset $I_x \subseteq [n]$ with $|I_x| \leq O(d)$ such that every "structured" $X^j \times Y^{j,\beta}$ intersecting $\{x\} \times \{0,1\}^{mn}$ has $\mathrm{fix}\,\rho^{j,\beta} \subseteq I_x$.

Our proof of Lemma 5.1 is nearly identical to [GGKS18].[5] For completeness we introduce their main result, sketch the simulation and proof of their main result using Lemma 5.1 and Full Range Lemma, and reprove Lemma 5.1. We refer those interested in the details to [GGKS18].

5.1 Preliminaries

To begin we briefly redefine decision trees and communication protocols in a slightly different way. Consider a search problem $f \subseteq \mathcal{Z} \times \mathcal{O}$ where $\mathcal{Z} = \{0,1\}^n$, and let $\mathcal{Q}$ be a family of functions from $\mathcal{Z}$ to $\{0,1\}$. More specifically let $\mathcal{Q}$ be the set of all conjunctions over $\mathcal{Z}$. We can think of a decision tree T for f as being a tree where each node v is labeled with a function $v(z) \in \mathcal{Q}$ such that

- $v(z) \equiv 1$ when v is the root of T

[5] In fact the only difference is that our definition of ρ-structured has a slightly stricter condition on $|Y|$; in [GGKS18] a coarse lower bound of $|Y^{j,\beta}| \geq 2^{mn-n^2}$ is enough. We stick to our definition for consistency and to show that the same improvements as in Section 4 hold in the dag-like setting as well.

– $v^{-1}(1) \subseteq u^{-1}(1) \cup w^{-1}(1)$ for any node v with children u and w
– $v^{-1}(1) \subseteq f^{-1}(o)$ for any leaf node v labeled with $o \in \mathcal{O}$

We can see that this exactly recaptures our notion of a decision tree. The root corresponds to the trivially satisfiable conjunction 1, and the leaves are labeled with some conjunction that is sufficient to guarantee some answer $o \in \mathcal{O}$. At any node v with children u and w, since $v(z)$, $u(z)$, and $w(z)$ are all conjunctions and $v^{-1}(1) \subseteq u^{-1}(1) \cup w^{-1}(1)$, it is not hard to see that there is some variable z_i such that $u(z)$ is a relaxation of $v(z) \wedge z_i$ and $w(z)$ is a relaxation of $v(z) \wedge \overline{z_i}$, or vice-versa.

This notion also generalizes to bipartite search problems $F \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$, where now $\mathcal{Q}$ is the family of functions from $\mathcal{X} \times \mathcal{Y}$ to $\{0,1\}$ corresponding to membership in $X \times Y \subseteq \mathcal{X} \times \mathcal{Y}$. Since $v^{-1}(1) \subseteq u^{-1}(1) \cup w^{-1}(1)$, it must be that the rectangles we test membership for at at u and w cover the rectangle being tested at v, and again it is not hard to see that this corresponds to a relaxation of testing membership in $X_v = X_u \sqcup X_w$ or $Y_v = Y_u \sqcup Y_w$.

Now we generalize this notion to dags, and for convenience we also generalize it to any family of functions $\mathcal{Q}$. For a search problem $f \subseteq \mathcal{Z} \times \mathcal{O}$ and a family of functions $\mathcal{Q}$ from $\mathcal{Z}$ to $\{0,1\}$, a *$\mathcal{Q}$-dag* is a directed acyclic graph where each node v is labeled with a function $v(z) \in \mathcal{Q}$ such that

– $v(z) \equiv 1$ when v is the root of T
– $v^{-1}(1) \subseteq u^{-1}(1) \cup w^{-1}(1)$ for any node v with children u and w
– $v^{-1}(1) \subseteq f^{-1}(o)$ for any leaf node v labeled with $o \in \mathcal{O}$

For $\mathcal{Z} = \{0,1\}^n$ a *conjunction dag* D solving f is a $\mathcal{Q}$-dag where $\mathcal{Q}$ is the set of all conjunctions over $\mathcal{Z}$. For conjunction dags our measure of complexity will be a bit different than size. The *width* of Π is the maximum number of variables occurring in any conjunction $v(z)$. We define

$$w(f) \;\coloneqq\; \text{least } \textit{width} \text{ of a conjunction dag solving } f.$$

For a bipartite search problem $F \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$ and a family of functions $\mathcal{Q}$ from $\mathcal{X} \times \mathcal{Y}$ to $\{0,1\}$, we define a $\mathcal{Q}$-dag solving F analogously. A *rectangle dag* Π solving F is a $\mathcal{Q}$-dag where $\mathcal{Q}$ is the set of all indicator vectors of rectangles $X \times Y \subseteq \mathcal{X} \times \mathcal{Y}$. We define

$$\mathsf{rect\text{-}dag}(F) \;\coloneqq\; \text{least } \textit{size} \text{ of a rectangle dag solving } F.$$

5.2 Main theorem

The main lifting theorem in [GGKS18] can be stated as follows:

Theorem 5.2 (**Dag-like Lifting Theorem** [GGKS18])**.** *Let f be a search problem over $\{0,1\}^n$, and let $m = n^{1+\epsilon}$. Then*

$$\log \mathsf{rect\text{-}dag}(f \circ \textsc{Ind}_m^n) \;=\; w(f) \cdot \Theta(\log m)$$

In fact one can easily check that the same scaling argument as Scaling Basic Lifting Theorem can also be applied to the proof of Dag-like Lifting Theorem.

Proof sketch. Given a rectangle dag Π of size m^d, we construct a conjunction dag D of width $O(d)$ to simulate Π. Our procedure is roughly the same as before, but with a slight twist: the protocol may have depth greater than d and can decide to "forget" some bits at each stage, at which point we will have to make sure the assignment ρ we maintain also stays small.

To start, at each node $v \in \Pi$ we partition the rectangle R_v using Lemma 5.1, and for the moment assume that the sets X_{err} and Y_{err} are empty. Now at the current node v we apply Full Range

Lemma with the guarantee that the x^* we find is in structured rectangles for *both* children of v. Then we use the query alignment property for both children and query all unknown bits for both sets. Then because of the full range of x^* we find a y^* compatible with all bits fixed, and move to the (structured) rectangle output by the partition at whichever child of v contains y^*, since R is in the union of the rectangles at v's children. Thus we maintain our invariant of being in a ρ-structured rectangle with at most $O(d)$ fixed bits, and when we arrive at a leaf the correctness is fairly immediate.

To deal with the error sets, we include a preprocessing step where we partition each v using Lemma 5.1 in a bottom-up fashion, and for each v we remove from R_v all error sets appearing in *descendants* of v. This guarantees that at the very root of Π, by losing an $m^d \cdot 2^{-2d \log m} \ll 1/4$ fraction of the rectangle associated with the root we will never encounter an error rectangle in our procedure.

Below we present Dag-like Simulation Protocol assuming we have Lemma 5.1 and Full Range Lemma in hand. We leave the verification of all details stated above to the reader we present the dag-like version of Simulation Protocol and leave the verification to the reader.

Algorithm 3: Dag-like Simulation Protocol

PREPROCESSING: initialize $X^*_{err} = \emptyset$ and $Y^*_{err} = \emptyset$;
for $v \in \Pi$ *starting from the leaves and going up to the root* **do**
  Update $X_v \leftarrow X_v \setminus X^*_{err}$ and $Y_v \leftarrow Y_v \setminus Y^*_{err}$;
  Apply Lemma 5.1 to $X_v \times Y_v$ and let $\{X_v^j\}_j$, $\{Y_v^{j,\beta}\}_{j;\beta}$, X_{err}, Y_{err}, $\{I_x\}_x$ be the outputs;
  Update $X^*_{err} \leftarrow X^*_{err} \cup X_{err}$ and $Y^*_{err} \leftarrow Y^*_{err} \cup Y_{err}$;
end
Initialize $v :=$ root of Π; $R := R_v$; $\rho = *^n$;
while v *is not a leaf* **do**
  Apply Full Range Lemma to $X_{\bar{J}} \times Y_{\bar{J}}$ to get $x^* \in X$;
  Let v_ℓ, v_r be the children of v, let j_ℓ, j_r be the indices such that $x^* \in X_{v_\ell}^{j_\ell}$ and $x^* \in X_{v_r}^{j_r}$, and let I_{j_ℓ} and I_{j_r} be the query alignment sets I_{x^*} for v_ℓ and v_r respectively;
  Query each variable z_i for every $i \in (I_{j_\ell} \cup I_{j_r}) \setminus J$, let $\beta_{j_\ell} \in \{0,1\}^{I_{j_\ell}}$ be the result concatenated with $\rho[J]$ and restricted to I_{j_ℓ}, and let $\beta_{j_r} \in \{0,1\}^{I_{j_r}}$ be defined analogously;
  Let $y^* \in Y$ be such that $\textsc{Ind}_m^{I_{j_\ell}}(x^*, y^*) = \beta_{j_\ell}$ and $\textsc{Ind}_m^{I_{j_r}}(x^*, y^*) = \beta_{j_r}$, and let $c \in \{\ell, r\}$ be such that $(x^*, y^*) \in X_{v_c}^{j_c} \times Y_{v_c}^{j_c,\beta_{j_c}}$;
  Update $X \times Y = X_{v_c}^{j_c} \times Y_{v_c}^{j_c,\beta_{j_c}}$ and $\rho \leftarrow \rho^{j_c,\beta_{j_c}}$;
end
Output the same value as v does;

□

Proof of Lemma 5.1. Our procedure for generating rectangles $X^j \times Y^{j,\beta}$ will be nearly the same as before, with the one small caveat that we run Phase I until $X^{\geq j}$ is empty instead of stopping after partitioning half of X. We first need to define the error rectangles X_{err} and Y_{err}. Intuitively every "structured" $X^j \times Y^{j,\beta}$ is $\rho^{j,\beta}$ structured for some $\rho^{j,\beta}$, and furthermore we want to ensure that the number of bits fixed in $\rho^{j,\beta}$ is at most $O(d)$. For X this means ensuring that I_j is small, while for Y this means ensuring that $Y^{j,\beta}$ is large. We initialize $J_{good} = [|\mathcal{F}|]$, and we repeatedly find "bad" $j \in J_{good}$ and add either X^j or $Y^{j,\beta}$ to X_{err} or Y_{err}.

– X_{err}: while there exists $j \in J_{good}$ such that $|I_j| > 40d$, update $X_{err} \leftarrow X_{err} \cup X^j$ and $J_{good} \leftarrow J_{good} \setminus \{j\}$

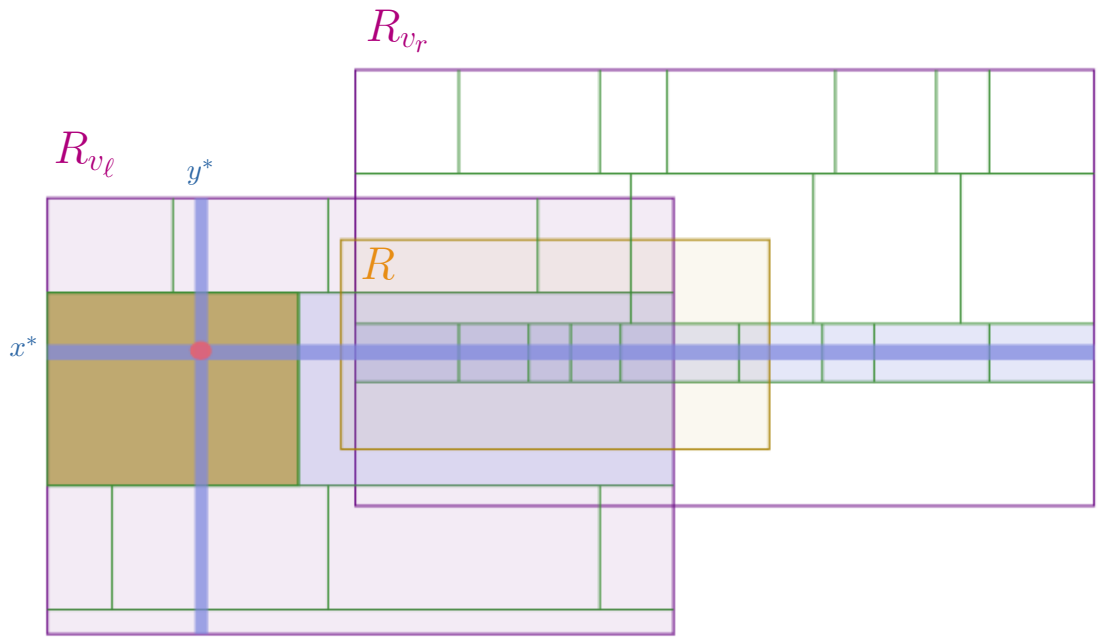

Figure 3: One iteration of Dag-like Simulation Protocol. We perform Rectangle Partition (green lines) on both R_{v_ℓ} and R_{v_r} separately, use Full Range Lemma find an $x^* \in R$ with full range, query all bits in the sets I_{j_ℓ} and I_{j_r} corresponding to $X^{j_\ell}, X^{j_r} \ni x^*$ (shaded in blue), find a y^* for which $\textsc{Ind}_m^n(x^*, y^*)$ matches the result, and set R to $X^{j_c} \times Y^{j_c,\beta_c} \ni (x^*, y^*)$ (shaded in brown) for $c \in \{\ell, r\}$ (shaded in purple).

– Y_{err}: while there exists $j \in J_{good}$ and β such that $|Y^{j,\beta} \setminus Y_{err}| < 2^{m\cdot|\overline{I_j}|-5d\log m}$, update $Y_{err} \leftarrow Y_{err} \cup Y^{j,\beta}$ for all such β and $J_{good} \leftarrow J_{good} \setminus \{j\}$

We prove a series of short claims about X_{err} and Y_{err}, most of which immediately follow in the same way as Claim 3.3, Claim 3.4, and Lemma 3.6. The first puts these claims together to show that all rectangles corresponding to $j \in J_{good}$ fulfill the "structured" case of Lemma 5.1.

Claim 5.3. *For all $j \in J_{good}$ and all $\beta \in \{0,1\}^{I_j}$, $X^j \times Y^{j,\beta}$ is $\rho^{j,\beta}$ structured for some $\rho^{j,\beta}$ which fixes at most $O(d)$ coordinates.*

Proof. As usual, for all j and for all $\beta \in \{0,1\}^{I_j}$, define $\rho^{j,\beta} \in \{0,1,*\}^n$ to be the restriction where $\text{fix}(\rho^{j,\beta}) = I_j$ and $\rho^{j,\beta}[I_j] = \beta$. Then

– by Claim 3.3, $X^j \times Y^{j,\beta}$ is fixed on I_j and outputs $\textsc{Ind}_m^{I_j}(X^j_{I_j}, Y^{j,\beta}_{I_j}) = \rho^{j,\beta}[I_j]$
– by Claim 3.4, $\boldsymbol{X}_{\overline{I_j}}$ has blockwise min-entropy $0.95\log m$
– since $j \in J_{good}$, it must be that $|Y^{j,\beta}| \geq 2^{m\cdot|\overline{I_j}|-5d\log m} \geq 2^{m\cdot|\overline{I_j}|-n\log m}$

and so $X^j \times Y^{j,\beta}$ is $\rho^{j,\beta}$-structured. Furthermore, since $j \in J_{good}$ it must be the case that $|\text{fix}(\rho^{j,\beta})| = |I_j| \leq 40d$. □

We briefly note that our error rectangle has succeeded in making sure any output row x^* from Full Range Lemma on $(X \setminus X_{err}) \times (Y \setminus Y_{err})$ lands in a good j (recall that we partitioned all of

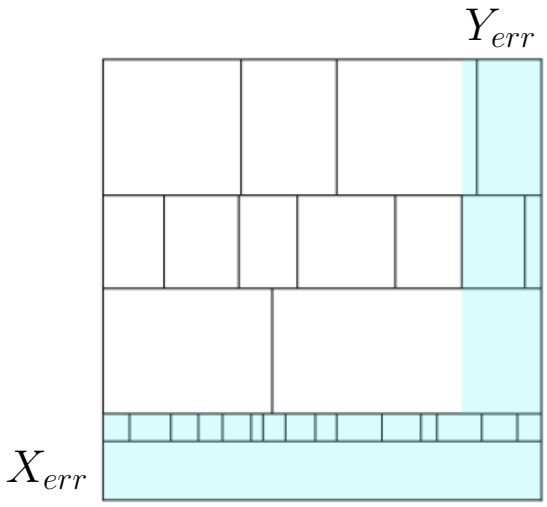

Figure 4: Error rectangles shaded in blue. X^j is added to X_{err} if I_j is too large (bottom), while $Y^{j,\beta}$ is added to Y_{err} if $Y^{j,\beta}$ is too small (right).

X in Phase I). This is not necessary to prove Lemma 5.1 but is needed to ensure that in Dag-like Simulation Protocol we can always query all bits of I_j for $X^j \ni x^*$ given by Full Range Lemma.

Claim 5.4. *Assume there exists an $x^* \in X \setminus X_{err}$ such that $\textsc{Ind}_m^n(x^*, Y \setminus Y_{err}) = \{0,1\}^n$, and let j be the unique index such that $x^* \in X^j$. Then $j \in J_{good}$.*

Proof. If $j \notin J_{good}$, then either $X^j \subseteq X_{err}$ or there exists a β such that $Y^{j,\beta} \subseteq Y^{err}$. The former case cannot happen as $x^* \in X \setminus X_{err}$, while in the latter case $\textsc{Ind}_m^n(x^*, y)$ is not consistent with $\rho^{j,\beta}$ for all $y \in Y \setminus Y_{err}$, which is a contradiction since $\textsc{Ind}_m^n(x^*, Y \setminus Y_{err}) = \{0,1\}^n$. □

Finally we handle the density of the error rectangles. In our simulation this will be used to ensure we can apply Full Range Lemma at every step.

Claim 5.5. $|X_{err}| \le m^n \cdot 2^{-2d \log m}$ *and* $|Y_{err}| \le 2^{mn} \cdot 2^{-2d \log m}$

Proof. For X_{err} we have two cases: either X_{err} is empty, in which case the claim is trivial, or X_{err} is not empty and there exists some largest j such that $X_{err} \subseteq X^{\ge j}$, and by extension $|I_j| > 40d$. Recall that we showed $|X^j| \le |X^{\ge j}| \cdot 2^{-0.95|I_j| \log m}$, and by extension $\mathbf{H}_\infty(\boldsymbol{X}^j) \ge \mathbf{H}_\infty(\boldsymbol{X}^{\ge j}) - |I_j| \cdot 0.95 \log m$. Then because X^j is a set in $[m]^n$ fixed on coordinates $I_j \subseteq n$, $\mathbf{H}_\infty(\boldsymbol{X}^j) \le (n - |I_j|) \log m$. Combining these two bounds gives us $\mathbf{H}_\infty(\boldsymbol{X}^{\ge j}) \le (n - 0.05|I_j|) \log m$, and by our choice of j we get that

$$|X_{err}| \le |X^{\ge j}| < 2^{(n - 0.05 \cdot 40d) \log m} < m^n \cdot 2^{-2d \log m}$$

For Y_{err}, as in the proof of Lemma 3.6 for all $k \in [40d]$ there are $\binom{n}{k} \cdot m^k \cdot 2^k < 2^{3k \log m}$ choices of (I_j, α_j, β_j) such that $|Y^{j,\beta_j}| < 2^{m \cdot (N-k) - 5d \log m}$, and taking a union bound we get that

$$|Y_{err}| \le \sum_{k=1}^{40d} 2^{3k \log m} \cdot 2^{m \cdot (N-k) - 5d \log m} \le 40d \cdot 2^{m(N-1) - 2d \log m} \ll 2^{mn} \cdot 2^{-2d \log m}$$

which completes the proof. □

The proof of Lemma 5.1 is now fairly immediate, and for completeness we state it explicitly. The density of X_{err} and Y_{err} follows from Claim 5.5. For any $X^j \times Y^{j,\beta}$, if $j \in J_{good}$ then by Claim 5.3 this fulfills the structured case, while if $j \notin J_{good}$ then either $X^j \subseteq X_{err}$ or $Y^{j,\beta} \subseteq Y_{err}$ by definition. The query alignment property holds by taking $I_x = I_j$ for all $x \notin X_{err}$. □

6 Lifting that scales with simulation length

In this section we prove a variant on Basic Lifting Theorem, which allows us to set the gadget size m in terms of the target decision tree depth d. This theorem was originally proven in [GKMP20] but here we get a simpler proof via the sunflower lemma together with simple counting.

Theorem 6.1 (Low-depth Lifting Theorem [GKMP20]). *Let f be a search problem over $\{0,1\}^n$, and let $m \geq (\mathbf{P}^{dt}(f))^{30}$. Then*

$$\mathbf{P}^{cc}(f \circ \textsc{Ind}_m) \;\geq\; \mathbf{P}^{dt}(f) \cdot \Omega(\log m)$$

Proof sketch. Again we start with a given real protocol Π of depth $d \log m$ for the composed problem $f \circ \textsc{Ind}_m^n$ and construct a decision-tree of depth $O(d)$ for f. Putting aside Full Range Lemma for the moment, we can apply Rectangle Partition as in Basic Lifting Theorem, and Claims 3.3, 3.4, and 3.5 hold with no change in proof. For Lemma 3.6 we impose the slightly stronger precondition that $|Y| \geq 2^{mN - O(d \log m)}$,[6] which will become useful when we go to prove Full Range Lemma. Our simulation proceeds according to to Simulation Protocol, and the same proofs hold to show our simulation's efficiency and correctness.

Finally we return to Full Range Lemma. To see why this is our main technical challenge, note that we can apply Claim 2.3 as before, but the rest of the argument no longer holds for $m \ll n$, as we cannot apply Blockwise Approximate Sunflower Lemma for $0.95 \log m < \log(K \log(N/\epsilon))$. Instead, it is enough to prove the following lemma, since we imposed the additional constraint $|Y| > 2^{mN - O(d \log m)}$ in the invariants and in Lemma 3.6. For convenience we reuse the shorthand $\gamma_j := (I_j, \alpha_j)$, and we round $0.95 \log m - O(1)$ down to $0.9 \log m$ for clarity.

Lemma 6.2. *Let $N \leq n$, let $d = o(n)$, and let $m > d^{30}$. Let X be such that $\boldsymbol{X}$ has blockwise min-entropy $0.9 \log m$, and let $\mathcal{F} = \{\gamma_j\}_j$ be a block-respecting set system over $[m]^N$ such that 1) for all $x \in X$ there exists a $\gamma_j \in \mathcal{F}$ consistent with x, and 2) $|\gamma_j| \leq O(d)$ for all j. Then*

$$\mathbf{Pr}_{\boldsymbol{y} \subseteq [mN]}(\forall j : \gamma_j \not\subseteq y) < 2^{-\Omega(d \log m)}$$

We prove this lemma in Section 6.1, as it is our main technical contribution. □

6.1 Proof of Lemma 6.2

Proof sketch. Our goal will be to show that $\mathcal{F}$ contains an $2^{-\Omega(d \log m)}$-approximate sunflower with an empty core. Note that in proving Full Range Lemma we invoked Blockwise Approximate Sunflower Lemma, which stated that X contained an ϵ-approximate sunflower with an empty core. However, while we still have blockwise min-entropy on X, for $m \ll N$ we cannot use Blockwise Approximate Sunflower Lemma on X.

Instead we turn to $\mathcal{F}$, and since we don't have a notion of blockwise min-entropy for $\mathcal{F}$ we switch to using the basic Approximate Sunflower Lemma, and instead use the blockwise min-entropy of X

[6] Since our invariants actually give $|Y^t| \geq 2^{m \cdot |\operatorname{free}(\rho^t)| - t - 2d \log m}$ for $t \leq d \log m$, we could have stated Lemma 3.6 this way originally, but to simplify the presentation we omitted any reference to d in Rectangle Partition

to ensure $\mathcal{F}$ is large. More specifically, because the blockwise min-entropy of X is at least $0.9 \log m$, for any non-empty set $\gamma_j \in \mathcal{F}$ the set of all $x \in X$ consistent with γ_j can only cover a $2^{-0.9|\gamma_j| \log m}$ fraction of X. Since each $x \in X$ must be consistent with some γ_j, there must be a huge number of sets γ_j in $\mathcal{F}$, and so by Approximate Sunflower Lemma $\mathcal{F}$ contains some ϵ-approximate sunflower $\mathcal{F}_{\bar{S}}$ even for very small ϵ.

If $|S| = 0$ then we are done, but unfortunately using Approximate Sunflower Lemma we have no control over $|S|$. Instead, we employ the strategy an iterative strategy where we drive down the size of the smallest core S for which $\mathcal{F}_{\bar{S}}$ is an ϵ-approximate sunflower. For simplicity assume there is some $s \leq 20d$ such that every set in $\mathcal{F}$ has size s, and so in the worst case we can assume that every core S for which $\mathcal{F}_{\bar{S}}$ is an ϵ-approximate sunflower has size $s-1$. We want to show now that there exist enough such cores S that the collection of these cores *itself* is an ϵ-approximate sunflower, and so it must have a core S' of size at most $s-2$. If this is true then it turns out $\mathcal{F}_{\bar{S}'}$ is an ϵ'-approximate sunflower for ϵ' only slightly larger than ϵ. From this we've made progress; by increasing ϵ slightly we've found a core of a smaller size.

Using this idea, at a high level we will perform an iterative procedure, where we repeat the following three steps until we find a sunflower with an empty core in $\mathcal{F}$: 1) repeatedly pluck ϵ-approximate sunflowers from $\mathcal{F}$; 2) when we have enough sunflowers, pluck an approximate sunflower from their cores; 3) increase ϵ enough so that the core of this new sunflower is the core of an ϵ-approximate sunflower in $\mathcal{F}$ as well. In our actual calculations we will need to keep track of the sets of cores of each size, as well as to focus only on the sets in $\mathcal{F}$ of a certain size. This will allow us to know when we should pluck a sunflower from the cores, and will give us a measure of progress towards finding an empty core, which will allow us to choose our ϵ small enough to get $2^{-\Omega(d \log m)}$ at the end.

The last remaining piece is showing that we can actually pluck enough sunflowers from $\mathcal{F}$ to repeat this procedure enough to get an empty core, without running out of sets in $\mathcal{F}$. Unfortunately when we find a core S and pluck the sunflower $\mathcal{F}_{\bar{S}}$, we have no control over how many sets are actually in $\mathcal{F}_{\bar{S}}$, and so it seems hopeless to control how many rounds we can run for. However, note that the $0.95 \log m$ lower bound on the blockwise min-entropy of X holds for *any* S over $[m]^N$, which applies to a) the original sets $\gamma_j \in \mathcal{F}$, and b) the cores S that we pluck. Thus instead of arguing that each $\mathcal{F}_{\bar{S}}$ we find is small, we instead argue that the fraction of X covered by sets remaining in $\mathcal{F}$ is large, using the blockwise min-entropy of X for all (non-empty) cores S we've found so far. Then, again using the blockwise min-entropy of X on $\mathcal{F}$, we know that $\mathcal{F}$ must still have many sets to cover the remaining fraction of X, as we did when showing that $\mathcal{F}$ was originally big enough to apply Approximate Sunflower Lemma. □

Proof. It is sufficient to show that $\mathcal{F}$ contains an ϵ-approximate sunflower with an empty core for some $\epsilon \leq 2^{-\Omega(d \log m)}$. Again we assume $\emptyset \notin \mathcal{F}$ as the lemma is trivial otherwise, and so for all $s \in [O(d)]$ let $\mathcal{F}(s)$ be the set of all sets in $\mathcal{F}$ of size exactly s, and let $X(s)$ be the set of all $x \in X$ consistent with a set in $\mathcal{F}(s)$. Since every x is consistent with some $\gamma \in \mathcal{F} = \cup_s \mathcal{F}(s)$, we know that $X = \cup_s X(s)$. Therefore by averaging there must exist some $s \in [O(d)]$ such that $|X(s)| \geq \frac{1}{O(d)}|X|$, and so we fix an arbitrary such s.

We define an iterative procedure to find an approximate sunflower with an empty core in $\mathcal{F}(s)$. Set $t = 0$, set $\epsilon_0 := 2^{-\Omega(d \log m) - s^2 \log m}$, and set $\mathcal{F}^0 := \mathcal{F}(s)$. For $k = 0 \ldots s-1$ set $\mathcal{S}^k := \emptyset$. We repeat the following until we ever add a set to $\mathcal{S}^0$:

0. **abort** *if the following invariants ever do not hold:*

(a) $|\mathcal{S}^k| \leq 2^{0.8k \log m}$
(b) $|\mathcal{F}^t| \geq 2^{0.8s \log m}$

(c) for every k and every $S \in \mathcal{S}^k$, $|S| = k$ and $\mathcal{F}(s)_{\bar{S}}$ is an ϵ_t-approximate sunflower
(d) $\epsilon_t < 2^{-\Omega(d \log m)}$

1. let $\mathcal{F}^t_{\bar{S}}$ be an ϵ_t-approximate sunflower in $\mathcal{F}^t$; if none exists, **abort**
2. increment t and set $\epsilon_t \leftarrow \epsilon_{t-1}$
3. set $\mathcal{S}^{|S|} \leftarrow \mathcal{S}^{|S|} \cup \{S\}$ and set $\mathcal{F}^t \leftarrow \mathcal{F}^{t-1} - \mathcal{F}^{t-1}_{\bar{S}}$
4. while there exists k such that $|\mathcal{S}^k| = 2^{0.8k \log m}$:
   (a) if $k = 0$, **exit** and return ϵ_t
   (b) let $\mathcal{S}^k_{\bar{S}}$ be an ϵ_0-approximate sunflower in $\mathcal{S}^k$; if none exists, **abort**
   (c) increment t and set $\epsilon_t \leftarrow \epsilon_{t-1} + \epsilon_0$
   (d) set $\mathcal{S}^{|S|} \leftarrow \mathcal{S}^{|S|} \cup \{S\}$, set $\mathcal{S}^{k'} \leftarrow \mathcal{S}^{k'} - \mathcal{S}^{k'}_{\bar{S}}$ for all $k' > |S|$, and set $\mathcal{F}^t \leftarrow \mathcal{F}^{t-1} - \mathcal{F}^{t-1}_{\bar{S}}$

If this process exits without aborting, clearly by invariants (c) and (d), $\mathcal{F}(s)$ is a $2^{-\Omega(d \log m)}$-approximate sunflower with an empty core as desired (note that when the procedure exits, $|\mathcal{S}^0| = 2^{0.8 \cdot 0 \log m} = 1$). Thus we prove that the process never aborts.

First we show that by Approximate Sunflower Lemma, in steps 1 and 4b we always find an approximate sunflower. Recall that $m > d^{30}$. For step 1, by invariant (b) and the fact that $1/\epsilon_t \leq 1/\epsilon_0 = 2^{\Omega(d \log m) + s^2 \log m}$ we have

$$|\mathcal{F}^t| \geq 2^{0.8s \log m} = (m^{0.8})^s \gg (\Omega(d^3 \log m))^s \geq (s \cdot \log \exp(\Omega(d \log m) + s^2 \log m))^s \geq (s \cdot \log 1/\epsilon_t)^s$$

and for step 4b by the inner loop condition the same calculation shows

$$|\mathcal{S}^k| = 2^{0.8k \log m} = (m^{0.8})^k \gg (\Omega(d^3 \log m))^k \geq (k \cdot \log \exp(\Omega(d \log m) + s^2 \log m))^k = (k \cdot \log 1/\epsilon_0)^k$$

We now prove that the invariants hold. For invariant (a), clearly after exiting the inner loop $|\mathcal{S}^k| < 2^{0.8k \log m}$ for all k. Before the inner loop runs we add at most one element to at most one set $\mathcal{S}^k$, and thus for that set $|\mathcal{S}^k| < 2^{0.8k \log m} + 1$, or in other words $|\mathcal{S}^k| \leq 2^{0.8k \log m}$. At the start of each iteration of the inner loop at most one set $\mathcal{S}^k$ has size $2^{0.8k \log m}$, and since we remove at least one element from it and add at most one element to at most one other set we maintain that invariant.

For invariant (b), assume for contradiction that $|\mathcal{F}^t| < 2^{0.8s \log m}$. Recall that $\boldsymbol{X}$ has blockwise min-entropy at least $0.9 \log m$, meaning that *every* set S over $[m]^N$ covers at most $2^{-0.9|S| \log m} \cdot |X|$ elements in X, and by extension in $X(s)$. In particular this applies to every set $\gamma_j \in \mathcal{F}^t$ as well as every set $S \in \mathcal{S}^k$. Lastly by assumption $|\mathcal{F}^t| < 2^{0.8s \log m}$, and likewise by invariant (a) we know that $|\mathcal{S}^k| < 2^{0.8k \log m}$ for every k. Therefore since $m > d^{30}$,

$$\begin{aligned}
|X(s)| &\leq |\mathcal{F}^t| \cdot (2^{-0.9s \log m} \cdot |X|) + \sum_{k=1}^{s-1} |\mathcal{S}^k| \cdot (2^{-0.9k \log m} \cdot |X|) \\
&< 2^{0.8s \log m} \cdot 2^{-0.9s \log m} \cdot |X| + \\
&\quad \sum_{k=1}^{s-1} 2^{0.8k \log m} \cdot 2^{-0.9k \log m} \cdot |X| \\
&= (\sum_{k=1}^{s} 2^{-0.1k \log m}) \cdot |X| \\
&\leq (s \cdot 2^{-0.1 \log m}) \cdot |X| \\
&\leq (s \cdot d^{-3}) \cdot |X| = \tfrac{1}{\omega(d)} |X|
\end{aligned}$$

which is a contradiction of our choice of s.

For invariant (c), we first note the following simple observation about sunflowers.

Fact 6.3. *Let $\mathcal{F}$ and $\mathcal{H}$ be any two set systems such that $\mathcal{H} \subseteq \mathcal{F}$, let $\epsilon, \epsilon' > 0$ be such that $\epsilon \leq \epsilon'$, and let S be any set. Then if $\mathcal{H}_{\bar{S}}$ is an ϵ-approximate sunflower, $\mathcal{F}_{\bar{S}}$ is also an ϵ' approximate sunflower.*

Consider $S \in \mathcal{S}^k$. Clearly $|S| = k$ by construction, and so we show that $\mathcal{F}(s)_{\bar{S}}$ is an ϵ_t-approximate sunflower. We consider only the value of t when S was added to $\mathcal{S}^k$, as ϵ_t only grows, and we do this by induction on t. First observe that for any t, if S was added to $\mathcal{S}^k$ in step 1 then the claim follows immediately since $\mathcal{F}^t \subseteq \mathcal{F}(s)$. This establishes the base case since at $t = 0$ we are at the start of the procedure, and so we consider $t > 0$. We show this with induction on k in reverse order from $s-1$ to 0. If $k = s-1$, since there is no $\mathcal{S}^{k'}$ for $k' > s-1$ it must have been added in step 1, and so again the claim follows immediately. Thus we consider $k < s-1$ and assume S was added in step 4b.

Let $k' > k$ be such that $\mathcal{S}^{k'}_S$ was the sunflower discovered in step 4b which made us add S to $\mathcal{S}^k$. We claim that $\mathcal{F}(s)_{\bar{S}}$ is an $(\epsilon_{t-1} + \epsilon_0)$-approximate sunflower, which completes the claim since $\epsilon_t = \epsilon_{t-1} + \epsilon_0$. Consider the probability that a random set $y \subseteq [mN] - S$ doesn't contain any set in $\mathcal{F}(s)_{\bar{S}}$. For this to happen, for every set $S' \in \mathcal{S}^{k'}_{\bar{S}}$ either y contains no sets in $\mathcal{F}(s)_{\bar{S}'}$ or it does not contain S' itself. If there is some S' such that $S' \subseteq y$, then by the inductive hypothesis on t and k we know that $\mathcal{F}^{t'}_{S'}$ is an $\epsilon_{t'}$-approximate sunflower, where $t' \leq t-1$ was the value of t when S' was added to $\mathcal{S}^k$. Since $\epsilon_{t'} \leq \epsilon_{t-1}$ and $\mathcal{F}^{t'} \subseteq \mathcal{F}(s)$, by extension y avoids every set in $\mathcal{F}(s)_{S'}$ with probability at most ϵ_{t-1}. In the other case where no such S' exists, then because $\mathcal{S}^{k'}_{\bar{S}}$ is an ϵ_0-approximate sunflower y avoids every set $S' \in \mathcal{S}^{k'}_{\bar{S}}$ with probability at most ϵ_0. Taking a union bound over these two events gives us our claim.

Finally for invariant (d), we claim that $t \leq 2^{s^2 \log m} - 1$ when the process ends. Putting this fact together with $\epsilon_0 := 2^{-\Omega(d \log m) - s^2 \log m}$ and $\epsilon_t \leq \epsilon_{t-1} + 2^{-\Omega(d \log m) - s^2 \log m}$ for all t gives us

$$\epsilon_t \leq \epsilon_0 + t \cdot \epsilon_0 \leq 2^{s^2 \log m} \cdot 2^{-\Omega(d \log m) - s^2 \log m} = 2^{-\Omega(d \log m)}$$

We associate each tuple $\mathcal{S} := (\mathcal{S}^k)_{k=1 \ldots s-1}$ with the string $\tau(\mathcal{S}) = |\mathcal{S}^1| \# |\mathcal{S}^2| \# \ldots \# |\mathcal{S}^{s-1}|$. We claim that for every t there is a unique string τ_t corresponding to $\tau(\mathcal{S})$ at the time t was incremented. This is simply because in every round of the outer loop we increase the size of at least one set $\mathcal{S}^k$, and in every round of the inner loop that we cause some $\mathcal{S}^k$ to shrink in some round of the inner loop, we also cause some set $\mathcal{S}^{k'}$ to grow where $k' < k$. By invariant (a) and the inner loop condition, $|\mathcal{S}^k| \leq 2^{0.8k \log m}$ for every k whenever we updated t, and so as long as $|\mathcal{S}^0| = 0$—in other words for all t except the very last one—we have

$$t \leq |\tau(\mathcal{S})| = \prod_{k=1}^{s-1} 2^{0.8k \log m} < (2^{0.8 \log m})^{\sum_{k \leq s} k} < 2^{s^2 \log m} - 2$$

and so at the end of the procedure $t \leq 2^{s^2 \log m} - 1$. □

7 Open problems

The next clear frontier for this counting style of lifting is the *BPP lifting* of [GPW17]. Two main improvements are needed to Lemma 3.6. First, we need to prove that for a *random* choice of $x \sim X$, the index j corresponding to $X^j \ni x$ is valid for Lemma 3.6. This is actually not very difficult; since our only restriction is on the blockwise min-entropy of $\boldsymbol{X}$, we can start by assuming for contradiction that a large fraction of X is bad, and remove those before applying Lemma 3.6 to the remainder. The much larger issue is that we need a *random* choice of y falls evenly in each $Y^{j,\beta}$. As noted before,

in Lemma 3.6 we only show that $|Y^{j,\beta}|$ is a multiplicative $2^{-|I_j|\log m}$ factor away from $|Y|/2^{-|I_j|}$, while [GPW17] show that $|Y^{j,\beta}|$ is an additive $1/\operatorname{poly}(n)$ factor away instead. Our argument comes from our upper bound on $|\mathcal{F}(k)|$ by counting the number of possible restrictions (I_j, α_j), and while this may seem coarse, even more subtle arguments—such as using the fact that each (I_j, α_j) violates $0.95 \log m$ blockwise min-entropy in $\boldsymbol{X}^{\geq j}$—fail to asymptotically improve this counting argument. More importantly, this is the also only barrier in our argument to getting a smaller gadget size in Improved Basic Lifting Theorem and Dag-like Lifting Theorem; if we could put the threshold for a bad $|Y^{j,\beta}$ at $|Y|/2^{-O(|I_j|)}$, it would immediately imply a gadget of size $m = \Theta(n)$.

Acknowledgements

The authors thank Paul Beame for comments, and Shachar Lovett, Raghu Meka, and Jiapeng Zhang for pointing to the goal of proving a quasilinear size gadget. Both authors was supported in part by NSERC, and the second author was supported in part by NSF grant No. CCF-1900460.

References

[ALWZ20] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 624–630. ACM, 2020. doi:10.1145/3357713.3384234.

[CFK+19] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. Technical Report TR19-103, Electronic Colloquium on Computational Complexity (ECCC), 2019. URL: https://eccc.weizmann.ac.il/report/2019/103/.

[CKLM17] Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudorandom properties. *CoRR*, abs/1704.06807, 2017. URL: http://arxiv.org/abs/1704.06807, arXiv:1704.06807.

[CLRS16] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. *J. ACM*, 63(4):34:1–34:22, 2016. doi:10.1145/2811255.

[dRMN+19] Susanna de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. Technical Report TR19-186, Electronic Colloquium on Computational Complexity (ECCC), 2019. URL: https://eccc.weizmann.ac.il/report/2019/186/.

[dRNV16] Susanna de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, pages 295–304. IEEE, 2016. doi:10.1109/FOCS.2016.40.

[ER60] Paul Erdös and R. Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 35(1):85–90, 1960.

[GGKS18] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In *Proceedings of the 50th Symposium on Theory of Computing (STOC)*, pages 902–911. ACM, 2018. doi:10.1145/3188745.3188838.

[GKMP20] Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is np-hard. In *Proceedings of the 52nd Symposium on Theory of Computing (STOC)*, 2020.

[GLM+16] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM J. Comput.*, 45(5):1835–1869, 2016. doi:10.1137/15M103145X.

[Göö15] Mika Göös. Lower bounds for clique vs. independent set. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1066–1076. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.69.

[GP18] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM Journal on Computing*, 47(5):1778–1806, 2018. doi:doi.org/10.1137/16M1082007.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1077–1088. IEEE, 2015. doi:10.1109/FOCS.2015.70.

[GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *Proceedings of the 58th Symposium on Foundations of Computer Science (FOCS)*, pages 132–143, 2017. doi:10.1109/FOCS.2017.21.

[HN12] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time–space trade-offs in proof complexity. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 233–248. ACM, 2012. doi:10.1145/2213977.2214000.

[KMR17] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 590–603. ACM, 2017. doi:10.1145/3055399.3055438.

[LLZ18] Xin Li, Shachar Lovett, and Jiapeng Zhang. Sunflowers and quasi-sunflowers from randomness extractors. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 51:1–51:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.51.

[LMZ20] Shachar Lovett, Raghu Meka, and Jiapeng Zhang. Improved lifting theorems via robust sunflowers. 2020.

[LRS15] James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In Rocco A. Servedio and Ronitt Rubinfeld,

editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 567–576. ACM, 2015. doi:10.1145/2746539.2746599.

[PR17] Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1246–1255. ACM, 2017. doi:10.1145/3055399.3055478.

[PR18] Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1207–1219. ACM, 2018. doi:10.1145/3188745.3188914.

[Rao19] Anup Rao. Coding for sunflowers. *CoRR*, abs/1909.04774, 2019.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi:10.1007/s004930050062.

[Ros10] Benjamin Rossman. Approximate sunflowers. *Manuscript*, 2010.

[RPRC16] Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 406–415. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.51.

[She11] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011. doi:10.1137/080733644.

[WYY17] Xiaodi Wu, Penghui Yao, and Henry S. Yuen. Raz-mckenzie simulation with the inner product gadget. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:10, 2017. URL: https://eccc.weizmann.ac.il/report/2017/010.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il