comput. complex. (2019), 113 – 144

1016-3328/19/010113-32
published online November 30, 2018
https://doi.org/10.1007/s00037-019-0175-5

computational complexity

QUERY-TO-COMMUNICATION LIFTING FOR P^{NP}

Mika Göös, Pritish Kamath, Toniann Pitassi, and Thomas Watson

Abstract. We prove that the P^{NP}-type query complexity (alternatively, decision list width) of any Boolean function f is quadratically related to the P^{NP}-type communication complexity of a lifted version of f. As an application, we show that a certain "product" lower bound method of Impagliazzo and Williams (CCC 2010) fails to capture P^{NP} communication complexity up to polynomial factors, which answers a question of Papakonstantinou, Scheder, and Song (CCC 2014).

Keywords. Query, Communication, Lifting, P^{NP}

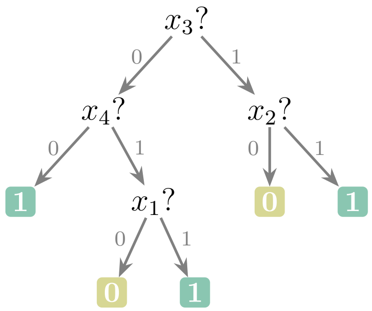
Subject classification. 68Q15

1. Introduction

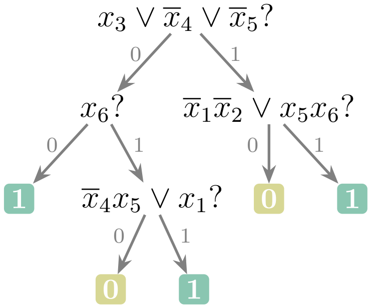
Broadly speaking, a ***query-to-communication lifting theorem*** (a.k.a. communication-to-query simulation theorem) translates, in a black-box fashion, lower bounds on some type of *query complexity* (a.k.a. decision tree complexity Buhrman & de Wolf (2002); Jukna (2012); Vereshchagin (1999)) of a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ into lower bounds on a corresponding type of *communication complexity* (Jukna (2012); Kushilevitz & Nisan (1997); Rao & Yehudayoff (2017)) of a two-party version of f. Table 1.1 lists several known results in this vein.

In this work, we provide a lifting theorem for P^{NP}-type query/communication complexity.

P^{NP} decision trees. Recall that a deterministic (i.e., P-type) decision tree computes an n-bit Boolean function f by repeatedly querying, at unit cost, individual bits $x_i \in \{0,1\}$ of the input x until the value $f(x)$ is output at a leaf of the tree. A P^{NP} decision tree is more powerful: in each step, it can query/evaluate a width-k DNF of its choice, at the cost of k. Here, k is simply the nondeterministic (i.e., NP-type) decision tree complexity of the predicate being evaluated at a node. The overall cost of a P^{NP} decision tree is the maximum over all inputs x of the sum of the costs of the individual queries that are made on input x. The P^{NP} query complexity of f, denoted $\mathsf{P}^{\mathsf{NPdt}}(f)$, is the least cost of a P^{NP} decision tree that computes f.



Deterministic decision tree of cost 3



P^{NP} *decision tree of cost 4*

EXAMPLE 1.1. Consider the fabled *odd-max-bit* function (Beigel (1994); Buhrman *et al.* (2007); Bun & Thaler (2018); Servedio *et al.* (2012); Thaler (2016)) defined by $\text{OMB}(x) \coloneqq 1$ iff $x \neq 0^n$ and the largest index $i \in [n]$ such that $x_i = 1$ is odd. This function admits an efficient $O(\log n)$-cost P^{NP} decision tree: we can *find* the largest i with $x_i = 1$ by using a binary search that queries 1-DNFs of the form $\bigvee_{a \leq j \leq n} x_j$ for different $a \in [n]$. $\Diamond$

P^{NP} communication protocols. Let $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a two-party function, i.e., Alice holds $x \in \mathcal{X}$, Bob holds $y \in \mathcal{Y}$. A deterministic communication protocol can be viewed as a decision tree where in each step, at unit cost, it evaluates either an arbitrary predicate of Alice's input x or an arbitrary predicate of

Query model	Communication model	References
Deterministic	Deterministic	Göös *et al.* (2018a); Raz & McKenzie (1999); Hatami *et al.* (2018); de Rezende *et al.* (2016)
Nondeterministic	Nondeterministic	Göös (2015); Göös *et al.* (2016)
Polynomial degree	Rank	Sherstov (2011); Shi & Zhu (2009); Razborov & Sherstov (2010); Robere *et al.* (2016)
Conical junta degree	Nonnegative rank	Göös *et al.* (2016); Kothari *et al.* (2017)
Sherali–Adams	LP extension complexity	Chan *et al.* (2016); Kothari *et al.* (2017)
Sum-of-squares	SDP extension complexity	Lee *et al.* (2015)

Table 1.1: Some query-to-communication lifting theorems. The first four are formulated in the language of boolean functions (as in this paper); the last two are formulated in the language of combinatorial optimization.

Bob's input y. A P^{NP} communication protocol (Babai *et al.* (1986); Göös *et al.* (2018b)) is more powerful: in each step, it can evaluate an arbitrary predicate of the form $(x, y) \in \bigcup_{i\in[2^k]} R_i$ ("oracle query") at the cost of k (We always assume $k \geq 1$, and k is an integer). Here, each R_i is a rectangle (i.e., $R_i = X_i \times Y_i$ for some $X_i \subseteq \mathcal{X}$, $Y_i \subseteq \mathcal{Y}$) and k is just the usual nondeterministic communication complexity of the predicate being evaluated. The overall cost of a P^{NP} protocol is the maximum over all inputs (x, y) of the sum of the costs of the individual oracle queries that are made on input (x, y). The P^{NP} communication complexity of F, denoted $\mathsf{P}^{\mathsf{NPcc}}(F)$, is the least cost of a P^{NP} protocol computing F.

Note that if $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ can be written as a k-DNF on $2n$ variables, then the nondeterministic communication complexity of F, denoted $\mathsf{NP}^{\mathsf{cc}}(F)$, is at most $O(k \log n)$ bits: we can guess one of the $\leq 2^k\binom{n}{k}$ many terms in the k-DNF and verify that the term evaluates to true. Consequently, any P^{NP} decision tree for a function f can be simulated efficiently by a P^{NP} protocol, regardless of how the input bits of f are split between Alice and Bob. That is, letting F be f equipped with any bipartition of the input bits, we have

$$\mathsf{P}^{\mathsf{NPcc}}(F) \;\leq\; \mathsf{P}^{\mathsf{NPdt}}(f) \cdot O(\log n). \tag{1.2}$$

1.1. Main result. Our main result establishes a rough converse to inequality (1.2) for a special class of *composed*, or *lifted*, functions. For an n-bit function f and a two-party function $g\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ (called a *gadget*), their composition $F \coloneqq f\circ g^n\colon \mathcal{X}^n\times\mathcal{Y}^n\to\{0,1\}$ is given by $F(x,y)\coloneqq f(g(x_1,y_1),\ldots,g(x_n,y_n))$. We use as a gadget the popular *index* function $\textsc{Ind}_m\colon [m]\times\{0,1\}^m$ defined by $\textsc{Ind}_m(x,y)\coloneqq y_x$.

THEOREM 1.3 (Lifting for P^{NP}). *Let $m = m(n) \coloneqq n^4$. For every $f\colon\{0,1\}^n\to\{0,1\}$,*

$$\mathsf{P}^{\mathsf{NPcc}}(f\circ\textsc{Ind}_m^n) \;\geq\; \sqrt{\mathsf{P}^{\mathsf{NPdt}}(f)\cdot\Omega(\log n)}.$$

The lower bound is tight up to the square root, since (1.2) can be adapted for composed functions to yield $\mathsf{P}^{\mathsf{NPcc}}(f\circ\textsc{Ind}_m^n)\leq\mathsf{P}^{\mathsf{NPdt}}(f)\cdot O(\log m+\log n)$. The reason we incur a quadratic loss is because we actually prove a *lossless* lifting theorem for a related complexity measure that is known to capture P^{NP} query/communication complexity up to a quadratic factor, namely *decision lists*, discussed shortly in Section 1.3.

1.2. Application. Impagliazzo & Williams (2010) gave the following criteria—we call it the *product method*—for a function F to have large P^{NP} communication complexity. Here, a *product* distribution μ over $\mathcal{X}\times\mathcal{Y}$ is such that $\mu(x,y)=\mu_{\mathcal{X}}(x)\cdot\mu_{\mathcal{Y}}(y)$ for some distributions $\mu_{\mathcal{X}},\mu_{\mathcal{Y}}$. A rectangle $R\subseteq\mathcal{X}\times\mathcal{Y}$ is *monochromatic* (relative to F) if F is constant on R.

> **Product method Impagliazzo & Williams (2010):** *Let $F\colon\mathcal{X}\times\mathcal{Y}\to\{0,1\}$ and suppose μ is a product distribution over $\mathcal{X}\times\mathcal{Y}$ such that $\mu(R)\leq\delta$ for every monochromatic rectangle R. Then,*
>
> $$\mathsf{P}^{\mathsf{NPcc}}(F) \;\geq\; \Omega(\log(1/\delta)).$$

This should be compared with the well-known *rectangle size method* (Karchmer *et al.* (1995); (Kushilevitz & Nisan 1997, §2.4)) (μ over $F^{-1}(1)$ such that $\mu(R)\leq\delta$ for all monochromatic R implies $\mathsf{NP}^{\mathsf{cc}}(F)\geq\Omega(\log(1/\delta))$), which is known to characterize nondeterministic communication complexity up to an additive $\Theta(\log n)$ term.

Papakonstantinou, Scheder, and Song (Papakonstantinou *et al.* 2014, Open Problem 1) asked whether the product method can yield a tight P^{NP} communication lower bound for every function. This is especially relevant in light of the fact that all existing lower bounds against $\mathsf{P}^{\mathsf{NPcc}}$ (proved in Impagliazzo & Williams (2010); Papakonstantinou *et al.* (2014)) have used the product method (except those lower bounds that hold against an even stronger model: unbounded error randomized communication complexity, $\mathsf{UPP}^{\mathsf{cc}}$ Paturi & Simon (1986)). We show that the product method can fail exponentially badly, even for total functions.

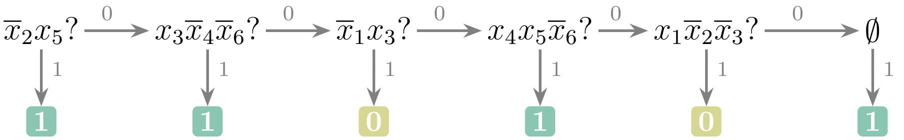
THEOREM 1.4. *There exists a total* $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *satisfying the following.*

- *F has large P^{NP} communication complexity:* $\mathsf{P}^{\mathsf{NPcc}}(F) \geq n^{\Omega(1)}$.
- *For any product distribution μ over $\{0,1\}^n \times \{0,1\}^n$, there exists a monochromatic rectangle R that is large:* $\log(1/\mu(R)) \leq \log^{O(1)} n$.

1.3. Decision lists (DLs).

Conjunction DLs. The following definition is due to Rivest (1987): a *conjunction decision list* of width k is a sequence $(C_1, \ell_1), \ldots, (C_L, \ell_L)$ where each C_i is a conjunction of $\leq k$ literals and $\ell_i \in \{0,1\}$ is a label. We assume for convenience that C_L is the empty conjunction (accepting every input). Given an input x, the conjunction decision list finds the least $i \in [L]$ such that $C_i(x) = 1$ and outputs ℓ_i. We define the conjunction decision list width of f, denoted $\mathsf{DL}^{\mathsf{dt}}(f)$, as the minimum k such that f can be computed by a width-k conjunction decision list. For example, $\mathsf{DL}^{\mathsf{dt}}(\textsc{Omb}) = 1$. This complexity measure is quadratically related to P^{NP} query complexity (see Appendix A).

FACT 1.5. *For all* $f\colon \{0,1\}^n \to \{0,1\}$, $\Omega(\mathsf{DL}^{\mathsf{dt}}(f)) \leq \mathsf{P}^{\mathsf{NPdt}}(f) \leq O(\mathsf{DL}^{\mathsf{dt}}(f)^2 \cdot \log n)$.



A conjunction decision list of width 3

Rectangle DLs. A communication complexity variant of decision lists was introduced by Papakonstantinou *et al.* (2014) (they called them *rectangle overlays*). A *rectangle decision list* of cost k is a sequence $(R_1, \ell_1), \ldots, (R_{2^k}, \ell_{2^k})$ where each R_i is a rectangle and $\ell_i \in \{0,1\}$ is a label. We assume for convenience that R_{2^k} contains every input. Given an input (x, y), the rectangle decision list finds the least $i \in [2^k]$ such that $(x, y) \in R_i$ and outputs ℓ_i. We define the rectangle decision list complexity of F, denoted $\mathsf{DL}^{\mathsf{cc}}(F)$, as the minimum k such that F can be computed by a cost-k rectangle decision list. We again have a quadratic relationship (Papakonstantinou *et al.* 2014, Theorem 3) (see Appendix A).

FACT 1.6. *For all $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $\Omega(\mathsf{DL}^{\mathsf{cc}}(F)) \leq \mathsf{P}^{\mathsf{NPcc}}(F) \leq O(\mathsf{DL}^{\mathsf{cc}}(F)^2)$.*

DLs are combinatorially slightly more comfortable to work with than P^{NP} decision trees/protocols. This is why our main lifting theorem (Theorem 1.3) is in fact derived as a corollary of a *lossless* lifting theorem for DLs.

THEOREM 1.7 (Lifting for DL). *Let $m = m(n) := n^4$. For every $f\colon \{0,1\}^n \to \{0,1\}$,*

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

Indeed, Theorem 1.3 follows because

$$\begin{aligned} \mathsf{P}^{\mathsf{NPcc}}(f \circ \textsc{Ind}_m^n) &\geq \Omega(\mathsf{DL}^{\mathsf{cc}}(f \circ \textsc{Ind}_m^n)) \\ &\geq \Omega(\mathsf{DL}^{\mathsf{dt}}(f) \cdot \log n) \\ &\geq \Omega((\mathsf{P}^{\mathsf{NPdt}}(f)/\log n)^{1/2} \cdot \log n) \\ &= (\mathsf{P}^{\mathsf{NPdt}}(f) \cdot \Omega(\log n))^{1/2}, \end{aligned}$$

where the first inequality is by Fact 1.6, the second is by Theorem 1.7, and the third is by Fact 1.5. We mention that Theorem 1.3 and Theorem 1.7, as well as Fact 1.5 and Fact 1.6, in fact hold for all partial functions.

As a curious aside, we mention that a time-bounded analogue of decision lists (capturing P^{NP}) has also been studied in a work of Williams (2001).

1.4. Separation between P^{NP} and DL. Fact 1.5 and Fact 1.6 show that decision lists can be converted to P^{NP} decision trees or protocols with a quadratic overhead. Is this conversion optimal? In other words, are there functions that witness a quadratic gap between P^{NP} and DL? We at least show that *if a lossless lifting theorem holds for* P^{NP}, then such a quadratic gap indeed exists for communication complexity.

CONJECTURE 1.8. *There is an $m = m(n) \coloneqq n^{\Theta(1)}$ such that for every $f\colon \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{P}^{\mathsf{NPcc}}(f \circ \textsc{Ind}_m^n) \;=\; \mathsf{P}^{\mathsf{NPdt}}(f) \cdot \Theta(\log n).$$

Our bonus contribution here (proven in Section 5) shows that the simple $O(\log n)$-cost P^{NP} decision tree for the odd-max-bit function is optimal:

THEOREM 1.9. $\mathsf{P}^{\mathsf{NPdt}}(\textsc{Omb}) \geq \Omega(\log n)$.

COROLLARY 1.10. *The second inequality of Fact 1.5 is tight (i.e., $\mathsf{P}^{\mathsf{NPdt}}(f) \geq \Omega(\mathsf{DL}^{\mathsf{dt}}(f)^2 \cdot \log n)$ for some f), and assuming Conjecture 1.8, the second inequality of Fact 1.6 is tight (i.e., $\mathsf{P}^{\mathsf{NPcc}}(F) \geq \Omega(\mathsf{DL}^{\mathsf{cc}}(F)^2)$ for some F).*

This corollary is witnessed by $f \coloneqq \textsc{Omb}$ (which has $\mathsf{DL}^{\mathsf{dt}}(f) \leq O(1)$ and $\mathsf{P}^{\mathsf{NPdt}}(f) \geq \Omega(\log n)$) and its lifted version $F \coloneqq \textsc{Omb} \circ \textsc{Ind}_m^n$ (which has $\mathsf{DL}^{\mathsf{cc}}(F) \leq O(\log n)$ and $\mathsf{P}^{\mathsf{NPcc}}(F) \geq \Omega(\log^2 n)$ under Conjecture 1.8). One caveat is that we have only shown the corollary for an extreme setting of parameters (constant $\mathsf{DL}^{\mathsf{dt}}(f)$ and logarithmic $\mathsf{DL}^{\mathsf{cc}}(F)$). It would be interesting to show a separation for functions of $n^{\Omega(1)}$ decision list complexity.

2. Preliminaries: decision list lower bound techniques

We present two basic lemmas in this section that allow one to prove lower bounds on conjunction/rectangle decision lists. First, we recall the proof of the product method, which will be important for us, as we will extend the proof technique in both Section 3 and Section 4.

LEMMA 2.1 (Product method for $\mathsf{DL}^{\mathsf{cc}}$). *Let $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and suppose μ is a product distribution over $\mathcal{X}\times\mathcal{Y}$. Then, F admits a monochromatic rectangle R with* $\log(1/\mu(R)) \leq O(\mathsf{DL}^{\mathsf{cc}}(F))$.

PROOF. (from Impagliazzo & Williams (2010);Papakonstantinou *et al.* (2014)) Let $(R_1, \ell_1), \ldots, (R_{2^k}, \ell_{2^k})$ be an optimal rectangle decision list of cost $k \coloneqq \mathsf{DL}^{\mathsf{cc}}(F)$ computing F. Recall we assume that $R_{2^k} = \mathcal{X} \times \mathcal{Y}$ contains every input. We find a monochromatic R with $\mu(R) \geq 2^{-2k}$ via the following process.

We initialize $X \coloneqq \mathcal{X}$ and $Y \coloneqq \mathcal{Y}$ and iterate the following for $i = 1, \ldots, 2^k$ rounds, shrinking the rectangle $X \times Y$ in each round.

(†) *Round i:* (loop invariant: $R_i \cap (X \times Y)$ is a monochromatic rectangle)

Write $R_i \cap (X \times Y) = X_i \times Y_i$ and test whether $\mu(X_i \times Y_i) = \mu_{\mathcal{X}}(X_i) \cdot \mu_{\mathcal{Y}}(Y_i)$ is at least 2^{-2k}. Suppose not, as otherwise we are successful. Then, either $\mu_{\mathcal{X}}(X_i) < 2^{-k}$ or $\mu_{\mathcal{Y}}(Y_i) < 2^{-k}$; say the former. We now "delete" the rows X_i from consideration by updating $X \leftarrow X \smallsetminus X_i$.

Note that since $R_i \cap (X \times Y)$ is removed from $X \times Y$ in each unsuccessful round, it must hold (inductively) that $\bigcup_{j<i} R_j$ is disjoint from $X \times Y$ at the start of the i-th round, and so $R_i \cap (X \times Y)$ is indeed monochromatic (since it only contains points for which R_i is the first rectangle in the decision list to contain them, which means F evaluates to ℓ_i). The process starts out with $\mu(X\times Y) = 1$ and in each unsuccessful round the quantity $\mu(X\times Y)$ decreases by $< 2^{-k}$. Some round must succeed, as otherwise the process would finish with $X \times Y = \emptyset$, and hence, $\mu(X \times Y) = 0$ in 2^k rounds, which is impossible. □

Recall that our Theorem 1.4 states that the product method is not complete for the measure $\mathsf{DL}^{\mathsf{cc}}$. By contrast, we are able to give an alternative characterization for the analogous query complexity measure $\mathsf{DL}^{\mathsf{dt}}$. We do not know if this characterization has been observed in the literature before.

LEMMA 2.2 (Characterization for $\mathsf{DL}^{\mathsf{dt}}$). *Let $f\colon \{0,1\}^n \to \{0,1\}$. Then, $\mathsf{DL}^{\mathsf{dt}}(f) \le k$ iff for every nonempty $Z \subseteq \{0,1\}^n$ there exists an $\ell \in \{0,1\}$ and a width-k conjunction that accepts an input in $Z_\ell \coloneqq Z \cap f^{-1}(\ell)$ but none in $Z_{1-\ell}$.*

PROOF. Suppose f has a width-k conjunction decision list $(C_1, \ell_1), (C_2, \ell_2), \ldots, (C_L, \ell_L)$. The first C_i that accepts an input in Z (such an i must exist since the last C_L accepts every input) must accept an input in Z_{ℓ_i} but none in $Z_{1-\ell_i}$ (since all inputs in $C_i^{-1}(1) \cap Z$ are such that C_i is the first conjunction in the decision list to accept them).

Conversely, assume the right side of the "iff" holds. Then, we can build a conjunction decision list for f iteratively as follows. Start with $Z = \{0,1\}^n$. Let C_1 be a width-k conjunction that accepts an input in some Z_{ℓ_1} but none in $Z_{1-\ell_1}$, and remove from Z all inputs accepted by C_1. Then continue with the new Z: let C_2 be a width-k conjunction that accepts an input in some Z_{ℓ_2} but none in $Z_{1-\ell_2}$, and further remove from Z all inputs accepted by C_2. Once Z becomes empty (this must happen since the right side of the iff holds for all nonempty Z), we have constructed a conjunction decision list $(C_1, \ell_1), (C_2, \ell_2), \ldots$ for f. □

3. Proof of the lifting theorem

In this section we prove Theorem 1.7, restated here for convenience.

THEOREM 1.7 (Lifting for DL). *Let $m = m(n) \coloneqq n^4$. For every $f\colon \{0,1\}^n \to \{0,1\}$,*

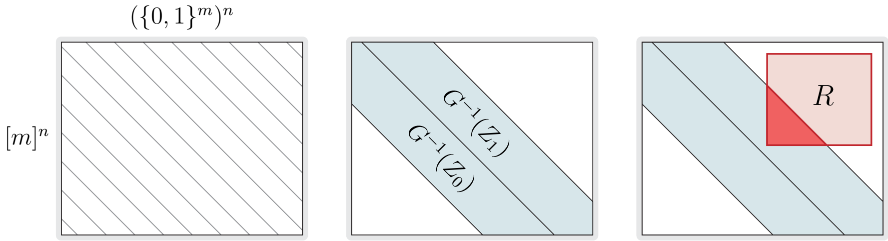
$$\mathsf{DL}^{\mathsf{cc}}(f \circ \textsc{Ind}_m^n) \;=\; \mathsf{DL}^{\mathsf{dt}}(f) \cdot \Theta(\log n).$$

We use the abbreviations $g \coloneqq \textsc{Ind}_m \colon [m] \times \{0,1\}^m \to \{0,1\}$ and $F \coloneqq f \circ g^n$.

The upper bound of Theorem 1.7 is straightforward: given a width-k conjunction decision list for f (which necessarily has length $\leq 2^k\binom{n}{k} \leq n^{O(k)}$), we can form a rectangle decision list for F by transforming each labeled conjunction into a set of same-labeled rectangles (which can be ordered arbitrarily among themselves), one for each of the m^k ways of choosing a row from each of the copies of g corresponding to bits read by the conjunction—for a total of $n^{O(k)} \cdot m^k \leq n^{O(k)}$ rectangles and hence a cost of $k \cdot O(\log n)$. For example, if $k = 2$ and the conjunction is $z_1\overline{z}_2$, then for each $x_1, x_2 \in [m]$ there would be a rectangle consisting of all inputs with that value of x_1, x_2 and with y_1, y_2 such that $g(x_1, y_1) = 1$ and $g(x_2, y_2) = 0$. For the rest of this section, we prove the matching lower bound.

3.1. Overview. Fix an optimal rectangle decision list $(R_1, \ell_1), \ldots, (R_{2^k}, \ell_{2^k})$ for F. By our characterization of $\mathsf{DL}^{\mathsf{dt}}$ (Lemma 2.2), it suffices to show that for every nonempty $Z \subseteq \{0,1\}^n$ there is a width-$O(k/\log n)$ conjunction that accepts an input in $Z_\ell \coloneqq Z \cap f^{-1}(\ell)$ for some $\ell \in \{0,1\}$, but none in $Z_{1-\ell}$. Thus fix some nonempty Z henceforth.

Write $G \coloneqq g^n$ for short. We view the communication matrix of F as being partitioned into *slices* $G^{-1}(z) = \{(x,y) : G(x,y) = z\}$, one for each $z \in \{0,1\}^n$; see (a) below. We focus naturally on the slices corresponding to Z, namely $G^{-1}(Z) = \bigcup_{z \in Z} G^{-1}(z)$, which is further partitioned into $G^{-1}(Z_0)$ and $G^{-1}(Z_1)$; see (b) below. Our goal is to find a rectangle R that touches $G^{-1}(Z_\ell)$ (for some ℓ) but not $G^{-1}(Z_{1-\ell})$, and such that $G(R) = C^{-1}(1)$ for a width-$O(k/\log n)$ conjunction C; see (c) below. Thus, $C^{-1}(1)$ touches Z_ℓ but not $Z_{1-\ell}$, as desired.



(a) **(b)** **(c)**

We find such an R as follows. We maintain a rectangle $X \times Y$, which is initially the whole domain of F and which we iteratively shrink. In each round, we consider the next rectangle R_i in the decision list, and one of two things happens. Either:

– The round is declared unsuccessful, in which case we remove from $X \times Y$ a small number of rows and columns that together cover all of $R_i \cap (X \times Y) \cap G^{-1}(Z)$. This guarantees that throughout the whole execution, by the i-th round, $\bigcup_{j<i}(R_j \cap G^{-1}(Z))$ has been removed from $X \times Y$—thus every input in $R_i \cap (X \times Y) \cap G^{-1}(Z)$ is such that R_i is the first rectangle in the decision list that contains it, so it is in $G^{-1}(Z_{\ell_i}) \subseteq F^{-1}(\ell_i)$ by the definition of decision lists.

Or,

– Success is declared, in which case it will hold that $R_i \cap (X \times Y)$ touches $G^{-1}(Z)$—in fact, it touches $G^{-1}(Z_{\ell_i})$ but not $G^{-1}(Z_{1-\ell_i})$, by the above—and we can restrict $R_i \cap (X \times Y)$ to a subrectangle R that still touches $G^{-1}(Z_{\ell_i})$ but is such that $G(R)$ is fixed on $O(k/\log n)$ coordinates and has full support on the remaining coordinates. In other words, $G(R) = C^{-1}(1)$ for a width-$O(k/\log n)$ conjunction C.

This process is a variation of the process (†) from the product method (Lemma 2.1). The difference is that the Z-slices, $G^{-1}(Z)$, now play the role of the product distribution, and we maintain the monochromatic property for $R_i \cap (X \times Y)$ only inside the Z-slices. Another difference is that in each unsuccessful round we remove *both* rows *and* columns from $X \times Y$ (not *either–or* as in (†)).

To flesh out this outline, we need to specify how to determine whether a round is successful, which rows and columns to remove if not, and how to restrict to the desired R if so, and we need to argue that the process will terminate with success.

3.2. Tools. We will need to find a rectangle R such that $G(R)$ is fixed on few coordinates and has full support on the remaining coordinates. We now describe some tools that help us achieve this. First of all, under what conditions on $R = A \times B$ can we guarantee that $G(R)$ has full support over all n coordinates?

DEFINITION 3.1 (Blockwise-density Göös *et al.* 2016). *$A \subseteq [m]^n$ is called δ-*dense *if the uniform random variable $\boldsymbol{x}$ over A satisfies the following: for every nonempty $I \subseteq [n]$, the blocks $\boldsymbol{x}_I$ have min-entropy rate at least δ, that is, $\mathbf{H}_\infty(\boldsymbol{x}_I) \geq \delta \cdot |I| \log m$. Here, $\boldsymbol{x}_I$ is marginally distributed over $[m]^I$, and $\mathbf{H}_\infty(\boldsymbol{x}) := \min_x \log(1/\mathbf{Pr}[\boldsymbol{x} = x])$ is the usual min-entropy of a random variable (see, e.g., Vadhan's monograph Vadhan (2012) for an introduction).*

DEFINITION 3.2 (Deficiency). *For $B \subseteq (\{0,1\}^m)^n$, we define $\mathbf{D}_\infty(B) := mn - \log|B|$ (equivalently, $|B| = 2^{mn-\mathbf{D}_\infty(B)}$), representing the log-size deficiency of B compared to the universe $(\{0,1\}^m)^n$. (The notation $\mathbf{D}_\infty$ was chosen partly because this corresponds to the Rényi max-divergence between the uniform distributions over B and over $(\{0,1\}^m)^n$.)*

LEMMA 3.3 (Full support). *If $A \subseteq [m]^n$ is 0.9-dense and $B \subseteq (\{0,1\}^m)^n$ satisfies $\mathbf{D}_\infty(B) \leq n^2$, then $G(A \times B) = \{0,1\}^n$ (i.e., for every $z \in \{0,1\}^n$ there are $x \in A$ and $y \in B$ with $G(x, y) = z$).*

We prove Lemma 3.3 in Section 3.4 using the probabilistic method: we show for a suitably randomly chosen rectangle $U \times V \subseteq G^{-1}(z)$, (i) U intersects A with high probability, and (ii) V intersects B with high probability. The proof of (i) uses the second moment method (which is different from how blockwise-density was employed in previous work Göös *et al.* (2016)). The proof of (ii), which is simpler than the one in the original version of this paper Göös *et al.* (2017), is inspired by arguments from Göös *et al.* (2018a); Raz & McKenzie (1999) (these papers proved the full support property under a different assumption on A, which they called "thickness") and a key suggestion from an anonymous reviewer.

Lemma 3.3 gives us the full support property assuming A is blockwise-dense and B has low deficiency. How can we get blockwise-density? Our tool for this is the following claim, which follows from Göös *et al.* (2016); we provide the simple argument.

CLAIM 3.4. *If $A \subseteq [m]^n$ satisfies $|A| \geq m^n/2^s$ then there exists an $I \subseteq [n]$ of size $|I| \leq 10s/\log m$ and an $A' \subseteq A$ such that A' is fixed on I and 0.9-dense on $\overline{I} := [n] \smallsetminus I$.*

PROOF. If A is 0.9-dense, then we can take $I = \emptyset$ and $A' = A$, so assume not. Letting $\boldsymbol{x}$ be the uniform random variable over A, take $I \subseteq [n]$ to be a maximal subset for which there is a violation of blockwise-density: $\mathbf{H}_\infty(\boldsymbol{x}_I) < 0.9 \cdot |I| \log m$. From $\mathbf{H}_\infty(\boldsymbol{x}) \geq n \log m - s$, we deduce $\mathbf{H}_\infty(\boldsymbol{x}_I) \geq |I| \log m - s$ since marginalizing out $|\overline{I}| \log m$ bits may only cause the min-entropy to go down by $|\overline{I}| \log m$. Combining these, we get $|I| \log m - s < 0.9 \cdot |I| \log m$, so $|I| \leq 10s/\log m$.

Let $\alpha \in [m]^I$ be an outcome for which $\mathbf{Pr}[\boldsymbol{x}_I = \alpha] > 2^{-0.9 \cdot |I| \log m}$, and take $A' := \{x \in A : x_I = \alpha\}$, which is fixed on I. To see that A' is 0.9-dense on $\overline{I}$, let $\boldsymbol{x}'$ be the uniform random variable over A' and note that if $\mathbf{H}_\infty(\boldsymbol{x}'_J) < 0.9 \cdot |J| \log m$ for some nonempty $J \subseteq \overline{I}$, a straightforward calculation shows that then $\boldsymbol{x}_{I \cup J}$ would also have min-entropy rate < 0.9, contradicting the maximality of I. □

3.3. Finding $\boldsymbol{R}$. We initialize $X := [m]^n$ and $Y := (\{0,1\}^m)^n$ and iterate the following for $i = 1, \ldots, 2^k$ rounds.

(‡) *Round i:* (loop invariant: $R_i \cap (X \times Y) \cap G^{-1}(Z)$ is monochromatic)

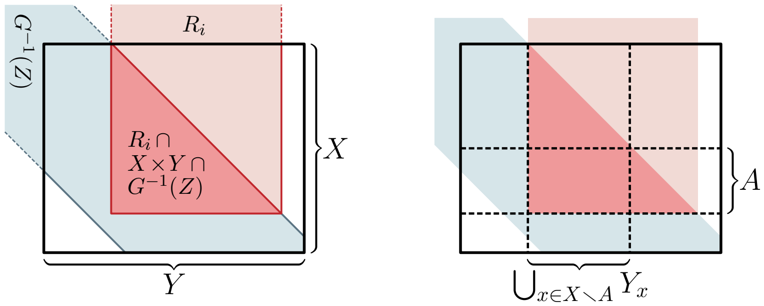
Define a set $A \subseteq X$ of *weighty rows* as

$$\begin{aligned} A &:= \left\{x \in X : |Y_x| \geq 2^{mn-n^2}\right\} \quad \text{where} \\ Y_x &:= \left\{y \in Y : (x,y) \in R_i \cap G^{-1}(Z)\right\}. \end{aligned}$$

Test whether there are many weighty rows: $|A| \geq m^n/2^{k+1}$?

- If *no*, we update $X \leftarrow X \smallsetminus A$ and $Y \leftarrow Y \smallsetminus \bigcup_{x \in X \smallsetminus A} Y_x$ and proceed to the next round. Since $R_i \cap G^{-1}(Z)$ has been removed from $X \times Y$, this ensures our loop invariant, as explained in Section 3.1.

– If *yes*, we declare this round a *success* and halt.



We shortly argue that the process halts with success. First, we show how to find a desired R assuming the process is successful in round i (with associated sets R_i, $X \times Y$, A, and Y_x for $x \in X$). Using Claim 3.4 with $s = k+1$, obtain $A' \subseteq A$ which is fixed to α on some $I \subseteq [n]$ of size $|I| \leq 10(k+1)/\log m \leq O(k/\log n)$ and is 0.9-dense on $\overline{I}$. Pick any $x' \in A'$, define $\beta \in (\{0,1\}^m)^I$ to maximize the size of $B \coloneqq \{y \in Y_{x'} : y_I = \beta\}$, and let $\gamma \coloneqq g^I(\alpha, \beta) \in \{0,1\}^I$. Note that $|B| \geq |Y_{x'}|/2^{m|I|} \geq 2^{mn-n^2-m|I|} = 2^{m|\overline{I}|-n^2}$ since $x' \in A$.

We claim that $R \coloneqq A' \times B$ can serve as our desired rectangle. Certainly, R touches $G^{-1}(Z_{\ell_i})$ (at (x', y) for any $y \in B$) but not $G^{-1}(Z_{1-\ell_i})$ by the loop invariant (since $R \subseteq R_i \cap (X \times Y)$). Also, $G(R)$ is fixed to γ on I. Defining

$$\begin{aligned} A'_{\overline{I}} &\coloneqq \left\{x_{\overline{I}} \in [m]^{\overline{I}} : \alpha x_{\overline{I}} \in A'\right\} \quad \text{and} \\ B_{\overline{I}} &\coloneqq \left\{y_{\overline{I}} \in (\{0,1\}^m)^{\overline{I}} : \beta y_{\overline{I}} \in B\right\} \end{aligned}$$

to be the projections of A' and B to the coordinates $\overline{I}$, we have that

$$A'_{\overline{I}} \text{ is 0.9-dense} \qquad \text{and} \qquad \mathbf{D}_\infty(B_{\overline{I}}) \leq n^2$$

(noting that $\mathbf{D}_\infty(B_{\overline{I}})$ is relative to $(\{0,1\}^m)^{\overline{I}}$). Applying Lemma 3.3 to $A'_{\overline{I}} \times B_{\overline{I}}$ shows[1] that $G(R)$ has full support on $\overline{I}$. In summary, "$z_I = \gamma$" is the conjunction we were looking for.

We now argue that the process halts with success. In each unsuccessful round, we remove $|A| < m^n/2^{k+1}$ rows from X and

---

[1] Technically, we need the result of Lemma 3.3 with $|I'|$ as the number of coordinates instead of n, but it still works.

at most $\sum_{x\in X\smallsetminus A}|Y_x| < m^n \cdot 2^{mn-n^2} \leq 2^{mn}/2^{k+1}$ columns[2] from Y (since $k+1 \leq n\log m+1 \leq n^2 - n\log m$). Suppose for contradiction that all 2^k rounds are unsuccessful, then at most half of the rows and half of the columns are removed altogether. Supposedly, the set $X \times Y$ we finish with is disjoint from $\bigcup_{i\in[2^k]}(R_i \cap G^{-1}(Z)) = G^{-1}(Z)$. But since Z is nonempty, this contradicts the fact that $G(X \times Y)$ has full support by Lemma 3.3 (as it is straightforward to check that since $X \times Y$ contains at least half the rows and half the columns, it also satisfies the assumptions of the lemma).

This concludes the proof of Theorem 1.7, except for the proof of Lemma 3.3.

3.4. Full support Lemma. In this section, we prove Lemma 3.3, restated here for convenience.

LEMMA 3.3 (Full support). *If $A \subseteq [m]^n$ is 0.9-dense and $B \subseteq (\{0,1\}^m)^n$ satisfies $\mathbf{D}_\infty(B) \leq n^2$, then $G(A \times B) = \{0,1\}^n$ (i.e., for every $z \in \{0,1\}^n$ there are $x \in A$ and $y \in B$ with $G(x,y) = z$).*

Fixing any $z \in \{0,1\}^n$, our goal is to show that $(A \times B) \cap G^{-1}(z) \neq \emptyset$. We write random variables as bold letters. For each $i \in [n]$ independently: Choose $\boldsymbol{U}_i \subseteq [m]$ by letting each $j \in [m]$ be in $\boldsymbol{U}_i$ independently with probability $m^{-0.64}$, and correspondingly define $\boldsymbol{V}_i \coloneqq \{y \in \{0,1\}^m : \forall j \in \boldsymbol{U}_i,\ y_j = z_i\}$. Then, let $\boldsymbol{U} \coloneqq \boldsymbol{U}_1 \times \cdots \times \boldsymbol{U}_n \subseteq [m]^n$ and $\boldsymbol{V} \coloneqq \boldsymbol{V}_1 \times \cdots \times \boldsymbol{V}_n \subseteq (\{0,1\}^m)^n$. We have $\boldsymbol{U} \times \boldsymbol{V} \subseteq G^{-1}(z)$ by construction, so it suffices to show that $(A \times B) \cap (\boldsymbol{U} \times \boldsymbol{V})$ is nonempty with positive probability. This holds by the following two claims and a union bound.

CLAIM 3.5 (Alice side). *$\mathbf{Pr}[A \cap \boldsymbol{U} \neq \emptyset] > 1/2$ if $A \subseteq [m]^n$ is 0.9-dense.*

[2] This is the main reason we need to use the index gadget in which Bob gets a polynomially long string, rather than, say, the inner product gadget on $O(\log n)$ bits, which has been used in other lifting theorems (Chattopadhyay *et al.* (2017); Göös *et al.* (2016); Wu *et al.* (2017)). Since we sum $|Y_x|$ over potentially m^n many x's, we need the threshold in the definition of weighty rows to be much less than an m^{-n} fraction of columns. For the inner product gadget, this would be less than one column, but for the index gadget it leaves a substantial number of columns—enough for the full support lemma.

Claim 3.6 (Bob side). *$\mathbf{Pr}[B \cap \boldsymbol{V} \neq \emptyset] > 1/2$ if $B \subseteq (\{0,1\}^m)^n$ satisfies $\mathbf{D}_\infty(B) \leq n^2$.*

Proof (of Claim 3.5). For each $x \in A$ consider the indicator random variable $\mathbf{1}_x \in \{0,1\}$ for whether $x \in \boldsymbol{U}$. Let $\boldsymbol{s} := \sum_{x \in A} \mathbf{1}_x$ so that $\boldsymbol{s} = |A \cap \boldsymbol{U}|$ and $\mathbf{E}[\boldsymbol{s}] = p|A|$, where $p := m^{-0.64n}$. By the second moment method, it will suffice to show that $\boldsymbol{s}$ has small variance. Blockwise-density implies that distinct elements of A are likely to disagree on most coordinates, in which case their contribution to the variance is small. We now carry out this argument. Since

$$\mathbf{Pr}[A \cap \boldsymbol{U} \neq \emptyset] \;=\; 1 - \mathbf{Pr}[\boldsymbol{s} = 0] \;\geq\; 1 - \frac{\mathbf{Var}[\boldsymbol{s}]}{\mathbf{E}[\boldsymbol{s}]^2},$$

to prove the claim it suffices to show that $\mathbf{Var}[\boldsymbol{s}] < \mathbf{E}[\boldsymbol{s}]^2/2 = p^2|A|^2/2$. Since

$$\mathbf{Var}[\boldsymbol{s}] \;=\; \textstyle\sum_{x,x'} \mathbf{Cov}[\mathbf{1}_x, \mathbf{1}_{x'}] \;=\; \textstyle\sum_{x,x'} \left(\mathbf{E}[\mathbf{1}_x \mathbf{1}_{x'}] - \mathbf{E}[\mathbf{1}_x]\mathbf{E}[\mathbf{1}_{x'}]\right),$$

it suffices to show that for each fixed $x^* \in A$,

$$\textstyle\sum_{x \in A} \mathbf{Cov}[\mathbf{1}_x, \mathbf{1}_{x^*}] \;<\; p^2|A|/2.$$

Fix $x^* \in A$. Let $I_x \subseteq [n]$ denote the set of all blocks i such that $x_i = x_i^*$. First note that if $I_x = \emptyset$ then $\mathbf{Cov}[\mathbf{1}_x, \mathbf{1}_{x^*}] = 0$, i.e., the events "$x \in \boldsymbol{U}$" and "$x^* \in \boldsymbol{U}$" are independent. The interesting case is thus $I_x \neq \emptyset$ when the two events are positively correlated. We note that

$$\text{(3.7)} \qquad \mathbf{Pr}[x \in \boldsymbol{U} \mid x^* \in \boldsymbol{U}] \;=\; (m^{-0.64})^{n-|I_x|} \;=\; m^{0.64|I_x|} \cdot p.$$

Let $\boldsymbol{I}$ be the distribution of $I_{\boldsymbol{x}}$ when $\boldsymbol{x} \in A$ is chosen uniformly at random. We have

$$\begin{aligned}
& \textstyle\sum_{x \in A} \mathbf{Cov}[\mathbf{1}_x, \mathbf{1}_{x^*}] \\
= \;& \textstyle\sum_{x : I_x \neq \emptyset} \mathbf{Cov}[\mathbf{1}_x, \mathbf{1}_{x^*}] \\
\leq \;& \textstyle\sum_{x : I_x \neq \emptyset} \mathbf{E}[\mathbf{1}_x \mathbf{1}_{x^*}] \\
= \;& \textstyle\sum_{x : I_x \neq \emptyset} \mathbf{Pr}[x \in \boldsymbol{U} \text{ and } x^* \in \boldsymbol{U}]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbf{Pr}[x^* \in \boldsymbol{U}] \cdot \textstyle\sum_{x: I_x \neq \emptyset} \mathbf{Pr}[x \in \boldsymbol{U} \mid x^* \in \boldsymbol{U}] \\
&= p \cdot \textstyle\sum_{x: I_x \neq \emptyset} \mathbf{Pr}[x \in \boldsymbol{U} \mid x^* \in \boldsymbol{U}] \\
&= p|A| \cdot \textstyle\sum_{\emptyset \neq I \subseteq [n]} \mathbf{Pr}[\boldsymbol{I} = I] \cdot \mathbf{E}_{\boldsymbol{x} \sim A | I_{\boldsymbol{x}} = I} \mathbf{Pr}[\boldsymbol{x} \in \boldsymbol{U} \mid x^* \in \boldsymbol{U}] \\
&\leq p|A| \cdot \textstyle\sum_{\emptyset \neq I \subseteq [n]} \mathbf{Pr}_{\boldsymbol{x} \sim A}[\boldsymbol{x}_I = x^*_I] \cdot \mathbf{E}_{\boldsymbol{x} \sim A | I_{\boldsymbol{x}} = I} \mathbf{Pr}[\boldsymbol{x} \in \boldsymbol{U} \mid x^* \in \boldsymbol{U}]
\end{aligned}
$$

(0.9-density and (3.7))

$$
\begin{aligned}
&\leq p|A| \cdot \textstyle\sum_{\emptyset \neq I \subseteq [n]} 2^{-0.9|I| \log m} \cdot m^{0.64|I|} \cdot p \\
&= p^2|A| \cdot \textstyle\sum_{\emptyset \neq I \subseteq [n]} 2^{-0.26|I| \log m} \\
&= p^2|A| \cdot \textstyle\sum_{k \in [n]} \binom{n}{k} 2^{-0.26k \log m} \\
&\leq p^2|A| \cdot \textstyle\sum_{k \in [n]} (m^{0.25})^k \cdot 2^{-0.26k \log m} \\
&\leq p^2|A| \cdot 2 \cdot 2^{-0.01 \log m} \\
&< p^2|A|/2.
\end{aligned}
$$

□

For the Bob side claim, we first reproduce the proof of a claim from Göös *et al.* (2018a). For any $W \subseteq \{0,1\}^\ell$ and $j \in [\ell]$ for some ℓ, we define $W_j \coloneqq \{w \in W : w_j = 1\}$ and $\mathrm{Bad}(W) \coloneqq \{j \in [\ell] : |W_j| < |W|/4\}$, and we recall that $\mathbf{D}_\infty(W) \coloneqq \ell - \log|W|$.

Claim 3.8. $|\mathrm{Bad}(W)| \leq 6\mathbf{D}_\infty(W)$ *for every* $W \subseteq \{0,1\}^\ell$.

Proof (of Claim 3.8). Let $\boldsymbol{w}$ be a random variable uniformly distributed over W, and let $\mathbf{H}(\cdot)$ denote the Shannon entropy. Note that $j \in \mathrm{Bad}(W)$ iff $\mathbf{Pr}[\boldsymbol{w}_j = 1] < 1/4$. There are at most $6\mathbf{D}_\infty(W)$ coordinates j such that $\mathbf{Pr}[\boldsymbol{w}_j = 1] < 1/4$, since otherwise $\mathbf{H}(\boldsymbol{w}) \leq \sum_{j=1}^{\ell} \mathbf{H}(\boldsymbol{w}_j) < 6\mathbf{D}_\infty(W) \cdot \mathbf{H}(1/4) + (\ell - 6\mathbf{D}_\infty(W)) \cdot 1 \leq \ell - 6\mathbf{D}_\infty(W) \cdot (1 - 0.82) \leq \ell - \mathbf{D}_\infty(W)$, contradicting the fact that $\mathbf{H}(\boldsymbol{w}) = \log|W| = \ell - \mathbf{D}_\infty(W)$. □

Proof (of Claim 3.6). We assume z is the all-1's string; the same argument works for any $z \in \{0,1\}^n$, but this assumption simplifies our notation somewhat.

Defining $\boldsymbol{T} \coloneqq \{(i,j) \in [n] \times [m] : j \in \boldsymbol{U}_i\}$, we have

$$\boldsymbol{V} \coloneqq \{y \in (\{0,1\}^m)^n : \forall (i,j) \in \boldsymbol{T},\ y_{i,j} = 1\}.$$

Now we may "flatten" things by forgetting the block structure: letting $\ell := mn$, we identify $[n] \times [m] = [\ell]$ and $(\{0,1\}^m)^n = \{0,1\}^\ell$, and we view $\boldsymbol{V} \subseteq \{0,1\}^\ell$ as the set of all strings with 1's in the positions indexed by $\boldsymbol{T} \subseteq [\ell]$.

The claim can be viewed as saying that if $B \subseteq \{0,1\}^\ell$ satisfies $\mathbf{D}_\infty(B) \leq n^2$ and we sample a random restriction $\boldsymbol{\rho} \in \{1,*\}^\ell$ by independently letting each $\boldsymbol{\rho}_j$ be 1 with probability $m^{-0.64}$ and be $*$ otherwise, then w.h.p. at least one string in B is consistent with the partial assignment $\boldsymbol{\rho}$ (it "survives" the random restriction). This corresponds to letting $\boldsymbol{T}$ be the positions fixed to 1 in $\boldsymbol{\rho}$ and $\boldsymbol{V}$ be all strings consistent with $\boldsymbol{\rho}$.

To analyze this, we introduce a procedure that samples $\boldsymbol{T}$ in an adaptive way that depends on B. We initialize $\boldsymbol{T}^0 := \emptyset$ and $\boldsymbol{B}^0 := B$. At the beginning of each step $i = 0, \ldots, \ell-1$, we will have a set $\boldsymbol{T}^i \subseteq [\ell]$ and correspondingly $\boldsymbol{B}^i := \{y \in B : \forall j \in \boldsymbol{T}^i,\ y_j = 1\}$. In step i, we select a $j \in [\ell]$ that has not been previously considered, and we update $\boldsymbol{T}^i$ to $\boldsymbol{T}^{i+1}$ by randomly deciding once and for all whether to include j (so $\boldsymbol{B}^{i+1} = \boldsymbol{B}^i_j$) or leave it out (so $\boldsymbol{B}^{i+1} = \boldsymbol{B}^i$). By the end, we will have sampled $\boldsymbol{T} := \boldsymbol{T}^\ell$ and $B \cap \boldsymbol{V} = \boldsymbol{B}^\ell$.

We let $\boldsymbol{J}^i \subseteq [\ell]$ be the set of coordinates that have not yet been considered by the beginning of step i. Thus, $\boldsymbol{T}^i \cap \boldsymbol{J}^i = \emptyset$ and the coordinates $[\ell] \setminus \boldsymbol{J}^i$ are "finalized" ($\boldsymbol{T}^i$ is guaranteed to agree with the final $\boldsymbol{T}$ on which of these are included). We have $\boldsymbol{J}^0 = [\ell]$ and $\boldsymbol{J}^\ell = \emptyset$ and in general $|\boldsymbol{J}^i| = \ell - i$ since we select one new coordinate to finalize in each step.

The procedure has two phases, and the random variable $\boldsymbol{i}^*$ records the step during which it switches from phase 1 to phase 2. Here is the procedure:

> Initialize $\boldsymbol{T}^0 := \emptyset$, $\boldsymbol{B}^0 := B$, and $\boldsymbol{J}^0 := [\ell]$.
> For $i = 0, 1, 2, \ldots, \ell - 1$:
> 1. If $\boldsymbol{i}^*$ is unassigned (phase 1):
>   1a. If $\boldsymbol{J}^i \subseteq \mathrm{Bad}(\boldsymbol{B}^i)$ then assign $\boldsymbol{i}^* := i$.
>   1b. Else non-randomly select any $j \in \boldsymbol{J}^i \setminus \mathrm{Bad}(\boldsymbol{B}^i)$.
> 2. If $\boldsymbol{i}^*$ is assigned (phase 2) then non-randomly select any $j \in \boldsymbol{J}^i$.
> 3. With probability $m^{-0.64}$ execute 3a; else execute 3b:
>   3a. Let $\boldsymbol{T}^{i+1} := \boldsymbol{T}^i \cup \{j\}$ and $\boldsymbol{B}^{i+1} := \boldsymbol{B}^i_j$.

3b. Let $\boldsymbol{T}^{i+1} \coloneqq \boldsymbol{T}^i$ and $\boldsymbol{B}^{i+1} \coloneqq \boldsymbol{B}^i$.

4. Let $\boldsymbol{J}^{i+1} \coloneqq \boldsymbol{J}^i \smallsetminus \{j\}$.

Let $\boldsymbol{T} \coloneqq \boldsymbol{T}^\ell$ and $\boldsymbol{V} \coloneqq \big\{y \in \{0,1\}^\ell : \forall j \in \boldsymbol{T},\, y_j = 1\big\}$, and if $\boldsymbol{i}^*$ is unassigned then let $\boldsymbol{i}^* \coloneqq \ell$.

This indeed generates a correctly distributed sample from $\boldsymbol{T}$ and $\boldsymbol{V}$, and we have $B \cap \boldsymbol{V} = \boldsymbol{B}^\ell$.

We have $\mathbf{E}[|\boldsymbol{T}|] = m^{-0.64}\ell = m^{0.61}$, and by $|\boldsymbol{T}^{\boldsymbol{i}^*}| \le |\boldsymbol{T}|$ and a standard concentration bound,

$$\mathbf{Pr}\big[|\boldsymbol{T}^{\boldsymbol{i}^*}| \le 2m^{0.61}\big] \;\ge\; \mathbf{Pr}\big[|\boldsymbol{T}| \le 2m^{0.61}\big] \;\ge\; 1 - e^{-m^{0.61}/3} \;\ge\; 3/4.$$

If i is a phase 1 step in which some j is added to $\boldsymbol{T}$, then $\mathbf{D}_\infty(\boldsymbol{B}^{i+1}) \le \mathbf{D}_\infty(\boldsymbol{B}^i) + 2$ since $j \notin \mathrm{Bad}(\boldsymbol{B}^i)$. Thus, conditioned on any outcome of phase 1 such that $|\boldsymbol{T}^{\boldsymbol{i}^*}| \le 2m^{0.61}$, by Claim 3.8 we have

$$\begin{aligned} |\boldsymbol{J}^{\boldsymbol{i}^*}| &\le |\mathrm{Bad}(\boldsymbol{B}^{\boldsymbol{i}^*})| \\ &\le 6\mathbf{D}_\infty(\boldsymbol{B}^{\boldsymbol{i}^*}) \\ &\le 6\big(\mathbf{D}_\infty(B) + 2|\boldsymbol{T}^{\boldsymbol{i}^*}|\big) \\ &\le 6(m^{0.5} + 4m^{0.61}) \\ &\le m^{0.62} \end{aligned}$$

in which case by a union bound, with probability at least $1 - m^{-0.64} \cdot m^{0.62} = 1 - m^{-0.02} \ge 3/4$, all of $\boldsymbol{J}^{\boldsymbol{i}^*}$ will remain excluded from $\boldsymbol{T}$, implying that $\boldsymbol{T} = \boldsymbol{T}^{\boldsymbol{i}^*}$ and $B \cap \boldsymbol{V} = \boldsymbol{B}^{\boldsymbol{i}^*}$, which is nonempty since $\mathbf{D}_\infty(\boldsymbol{B}^{\boldsymbol{i}^*}) < m^{0.62}$ is finite. In summary,

$$\begin{aligned} &\mathbf{Pr}[B \cap \boldsymbol{V} \ne \emptyset] \\ \ge\; &\mathbf{Pr}\big[|\boldsymbol{T}^{\boldsymbol{i}^*}| \le 2m^{0.61}\big] \cdot \mathbf{Pr}\big[B \cap \boldsymbol{V} \ne \emptyset \;\big|\; |\boldsymbol{T}^{\boldsymbol{i}^*}| \le 2m^{0.61}\big] \\ \ge\; &\tfrac{3}{4} \cdot \tfrac{3}{4} \\ >\; &\tfrac{1}{2}. \end{aligned}$$

□

4. Application

In this section, we prove Theorem 1.4, restated here for convenience.

THEOREM 1.4. *There exists a total $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ satisfying the following.*

- *F has large P^{NP} communication complexity:* $\mathsf{P}^{\mathsf{NPcc}}(F) \geq n^{\Omega(1)}$.
- *For any product distribution μ over $\{0,1\}^n \times \{0,1\}^n$, there exists a monochromatic rectangle R that is large:* $\log(1/\mu(R)) \leq \log^{O(1)} n$.

The function witnessing the separation is $F := f \circ g^n$ where $g := \mathrm{IND}_m$ is the index function with $m := n^4$ and $f\colon \{0,1\}^n \to \{0,1\}$ is defined as follows. We interpret the input M to f as a $\sqrt{n} \times \sqrt{n}$ Boolean matrix, and set

$$f(M) := 1 \qquad \text{iff} \qquad \text{every row of } M \text{ contains a unique 1-entry.}$$

Complexity class aficionados (Aaronson *et al.* (2017)) can recognize f as the canonical complete problem for the decision tree analogue of $\forall\cdot\mathsf{US}$ ($\subseteq \Pi_2\mathsf{P}$) where US is the class of functions whose 1-inputs admit a *unique* witness (Blass & Gurevich (1982)). We have $F\colon \{0,1\}^{n\log m} \times \{0,1\}^{nm} \to \{0,1\}$, but we can polynomially pad Alice's input length to match Bob's (as in the statement of Theorem 1.4).

4.1. Lower bound. It is proved in several sources (Håstad *et al.* (1995); Ko (1990); Santha (1989)) that f cannot be computed by an efficient $\Sigma_2\mathsf{P}$-type decision tree (i.e., quasi-polynomial-size depth-3 circuit with an OR-gate at the top and small bottom fan-in), let alone an efficient P^{NP} decision tree. However, for completeness, we might as well give a simple proof using our characterization (Lemma 2.2). Applying the lifting theorem to the following lemma yields the lower bound.

LEMMA 4.1. $\mathsf{DL}^{\mathsf{dt}}(f) \geq \sqrt{n}$.

PROOF. By Lemma 2.2, it is enough to exhibit a nonempty subset $Z \subseteq \{0,1\}^n$ of inputs such that each conjunction C of width $\sqrt{n}-1$ accepts an input in $Z_1 := Z \cap f^{-1}(1)$ iff it accepts an input in $Z_0 := Z \cap f^{-1}(0)$. We define Z as the set of $\sqrt{n} \times \sqrt{n}$ matrices with at most one 1-entry in each row. If C accepts an input

$M \in Z_1$, then there is some row of M none of whose entries are read by C; we may modify that row to all-0 and conclude that C accepts an input in Z_0. If C accepts an input $M \in Z_0$, then for each all-0 row of M there is some entry that is not read by C; we may modify each of those entries to a 1 and conclude that C accepts an input in Z_1. $\square$

4.2. Upper bound. Let μ be a product distribution over the domain of $F = f \circ g^n$. Call a matrix M *heavy* if it contains a row with at least two 1-entries. Hence, $f(M) = 0$ for every heavy matrix M. There is an efficient nondeterministic protocol of cost $k \leq O(\log n)$, call it Π, that checks whether a particular (x, y) describes a heavy matrix $M = g^n(x, y)$. Namely, Π guesses a row index $i \in [\sqrt{n}]$ and two column indices $1 \leq j < j' \leq \sqrt{n}$, and then communicates $2 \log m + 1 \leq O(\log n)$ bits to check that $M_{ij} = M_{ij'} = 1$. Thus, letting F' be such that $F'(x, y) \coloneqq 1$ iff $M = g^n(x, y)$ is heavy, we have

$$\mathsf{DL}^{\mathsf{cc}}(F') \;\leq\; O(\mathsf{P}^{\mathsf{NPcc}}(F')) \;\leq\; O(\mathsf{NP}^{\mathsf{cc}}(F')) \;\leq\; O(\log n).$$

Hence, we can apply the product method (Lemma 2.1) to find a rectangle S that is monochromatic for F' with $\log(1/\mu(S)) \leq O(\log n)$. If S is 1-monochromatic for F' then it is 0-monochromatic for F and we are done, so now assume S is 0-monochromatic for F'. We will complete the argument by showing that F_S (i.e., F restricted to the rectangle S) admits a large monochromatic rectangle relative to μ_S, the conditional distribution of μ given S (which is also product).

$$\begin{array}{l} \forall\cdot\mathsf{US} \\ \big\downarrow \ \text{restrict to } S \\ \forall\cdot\mathsf{UP} \\ \big\downarrow \ \text{Yannakakis} \\ \forall\cdot\mathsf{P} \\ \big\downarrow \ = \\ \mathsf{coNP} \\ \big\downarrow \ \text{product method} \\ \text{Large monochr. rectangle} \end{array}$$

All $(x, y) \in S$ are such that $M = g^n(x, y)$ is *not* heavy. This means that the function F_S is easier than the ($\forall\cdot\mathsf{US}$-complete) function F in the following sense: for each row $i \in [\sqrt{n}]$ there is an efficient $O(\log n)$-cost nondeterministic protocol, call it Π_i, to check whether the i-th row of $M = g^n(x, y)$ contains a 1-entry, and moreover, this protocol is *unambiguous* in that it has at most one accepting computation on any input. (In complexity lingo, F_S admits an efficient $\forall\cdot\mathsf{UP}$ protocol.) It is a well-known theorem of (Yannakakis 1991, Lemma 1) that any such unambiguous Π_i can be made deterministic with at most a quadratic blow-up in cost; let $\Pi_i^{\det}$ be that $O(\log^2 n)$-bit deterministic protocol. But now $\neg F_S$ (negation of F_S) is computed by the following $O(\log^2 n)$-bit nondeterministic protocol: on input (x, y) guess a row index $i \in [\sqrt{n}]$ and run $\Pi_i^{\det}$ accepting iff $\Pi_i^{\det}(x, y) = 0$. (That is, F_S admits an efficient $\forall\cdot\mathsf{P} = \mathsf{coNP}$ protocol.) We proved $\mathsf{NP}^{\mathsf{cc}}(\neg F_S) \leq O(\log^2 n)$; in particular,

$$\mathsf{DL}^{\mathsf{cc}}(F_S) \;\leq\; O(\mathsf{P}^{\mathsf{NP}\mathsf{cc}}(F_S)) \;\leq\; O(\mathsf{NP}^{\mathsf{cc}}(\neg F_S)) \;\leq\; O(\log^2 n).$$

Hence, we can apply the product method (Lemma 2.1) to find a monochromatic rectangle $R \subseteq S$ with $\log(1/\mu_S(R)) \leq O(\log^2 n)$ and hence $\log(1/\mu(R)) = \log(1/\mu_S(R)) + \log(1/\mu(S)) \leq O(\log^2 n)$. This completes the proof of Theorem 1.4.

In summary, the above proof finds a monochromatic rectangle S for F' using a nondeterministic protocol, then a monochromatic subrectangle of S for F using a conondeterministic protocol. These protocols cannot be "combined" into a P^{NP} protocol for F since the conondeterministic one only works on S, so there is no contradiction with the lower bound. We also mention that nondeterministic protocols yield large monochromatic rectangles even for non-product distributions (Karchmer *et al.* (1995); Kushilevitz & Nisan (1997)) but only for distributions over inputs accepted by the protocols. We need μ to be product so we can find monochromatic rectangles even though μ is distributed over potentially all inputs.

5. Odd-max-bit lower bound

PROOF (of Theorem 1.9: $\mathsf{P}^{\mathsf{NPdt}}(\textsc{Omb}) \geq \Omega(\log n)$).

Consider any P^{NP} decision tree of cost $o(\log n)$, i.e., on every root-to-leaf path, the sum of the widths of the DNFs queried is $o(\log n)$. We exhibit an adversary strategy that finds an input on which the decision tree fails to compute OMB. The adversary maintains a partial assignment (which fixes some of the input bits to 0 or to 1 and leaves others unfixed), starting with the empty assignment and fixing more bits in each round until a complete input has been specified at the end. The game between the decision tree and the adversary follows a root-to-leaf path (with one round per node on the path), and the adversary ensures that all inputs consistent with the current partial assignment indeed lead the decision tree to the current node. In other words, in each non-leaf round the adversary extends the partial assignment in a way that forces the current DNF query to evaluate to a particular value (0 or 1). In the leaf round, the adversary fixes all remaining bits to get an input x such that the output produced at the leaf disagrees with $\textsc{Omb}(x)$.

Our adversary strategy maintains a contiguous "`range`" of indices (with smaller indices being thought of as to the left, and larger indices to the right), with the following key invariants:

(I) All bits to the right of `range` are fixed to 0.
(II) No bit within `range` is fixed to 1.

Thus, an example picture to have in mind, showing a partial assignment x and `range` at some time during the execution, is as follows:

$$* \; * \; 1 \; * \; * \; 0 \; 1 \; * \left[0 \; * \; 0 \; * \right] 0 \; 0 \; 0 \; 0$$

Here is our adversary strategy:

1. Initialize x = empty partial assignment, `range` $= [n]$, and `node` = root of the decision tree.

2. While `node` is not a leaf:
   2a. If the DNF queried at `node` contains a term that (i) is not refuted by x *and* (ii) does not contain a positive literal whose index is in the right half of `range`, then:
      ▷ Extend x by fixing the bits appearing in the term in the unique way to satisfy it, thus ensuring that x forces the DNF to evaluate to 1.
      ▷ Restrict `range` to its right half.
      ▷ Update `node` by following the edge labeled 1.
   2b. Otherwise, if every term of the DNF either (i) is refuted by x *or* (ii) contains a positive literal whose index is in the right half of `range`, then:
      ▷ Extend x by fixing all remaining bits in the right half of `range` to 0, thus ensuring that x forces the DNF to evaluate to 0.
      ▷ Restrict `range` to its left half.
      ▷ Update `node` by following the edge labeled 0.
3. When `node` becomes a leaf:
   ▷ Find an index i in `range` such that x_i is unfixed and such that i is odd if the leaf's output is 0 and is even if the leaf's output is 1.
   ▷ Fix $x_i = 1$.
   ▷ Fix all remaining bits of x to the right of i to 0.
   ▷ Fix all remaining bits of x to the left of i arbitrarily.

It is straightforward to verify that this adversary indeed maintains invariants (I) and (II) and ensures that all inputs consistent with the current x lead to the current `node`. Furthermore, since in each round the adversary fixes bits according to one term and cuts `range` in half, the following properties are maintained throughout the execution:

(III) The number of bits in `range` that are fixed to 0 is at most the sum of the widths of the DNFs queried so far.
(IV) The size of `range` is $n/2^{\mathrm{depth}(\texttt{node})}$ where depth(`node`) is the distance of `node` from the root.

Assuming that in step 3 such an i does, in fact, exist, (I) and (II) guarantee that i is the maximum index of a 1 in the final x, so the output of the decision tree on x disagrees with $\textsc{Omb}(x)$. To see that such an i exists, note that (III) guarantees that the number of fixed bits in `range` is at most the cost of the decision tree, which is at most $\log n$, while (IV) guarantees that the size of `range` is at least $n/2^{\text{depth-of-tree}} \geq n/2^{o(\log n)} > 2\log n + 1$. Thus in step 3, `range` must contain both an odd unfixed index and an even unfixed index. $\square$

6. Conclusion

Let $\mathsf{PM}(F)$ denote the best lower bound on $\mathsf{DL}^{\mathsf{cc}}(F)$ that can be derived by the product method (Lemma 2.1). For any communication complexity measure $\mathcal{C}(F)$, we use the convention that $\mathcal{C}$ by itself refers to the class of (families of) functions $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ with $\mathcal{C}(F) \leq \text{polylog}(n)$. Then, our application (Theorem 1.4) shows that the inclusion $\mathsf{P}^{\mathsf{NPcc}} \subseteq \mathsf{PM}$ is strict: there is an $F \in \mathsf{PM} \smallsetminus \mathsf{P}^{\mathsf{NPcc}}$. Here are some open questions. (See Göös *et al.* (2018b) for definitions of classes such as $\mathsf{PP}^{\mathsf{cc}}$, $\mathsf{UPP}^{\mathsf{cc}}$, and $\mathsf{PSPACE}^{\mathsf{cc}}$.)

(1) Is there an $F \in \mathsf{PM} \smallsetminus \mathsf{UPP}^{\mathsf{cc}}$? This would be a stronger result since $\mathsf{P}^{\mathsf{NPcc}} \subseteq \mathsf{UPP}^{\mathsf{cc}}$. Note that our $\forall\cdot\mathsf{US}$-complete function does not witness this, since it is in $\mathsf{PP}^{\mathsf{cc}}$. One way to see this is to note that it is the intersection of a $\mathsf{coNP}^{\mathsf{cc}}$ function (does each row have at most one 1?) and a $\mathsf{PP}^{\mathsf{cc}}$ function (is the number of 1's at least the number of rows?), and use the closure of PP under intersection (Beigel *et al.* (1995)).

(2) Is there any reasonable upper bound for PM? For example, does $\mathsf{PM} \subseteq \mathsf{PSPACE}^{\mathsf{cc}}$ hold?

(3) Does $\mathsf{BPP}^{\mathsf{cc}} \subseteq \mathsf{PM}$ or even $\mathsf{BPP}^{\mathsf{cc}} \subseteq \mathsf{P}^{\mathsf{NPcc}}$ hold for total functions? The separation $\mathsf{BPP}^{\mathsf{cc}} \not\subseteq \mathsf{PM}$ was shown for partial functions implicitly in Papakonstantinou *et al.* (2014).

(4) Is there a lossless $\mathsf{P}^{\mathsf{NPdt}}$-to-$\mathsf{P}^{\mathsf{NPcc}}$ lifting theorem (Conjecture 1.8)?

(5) Can the quadratic upper bounds in Fact 1.5 and Fact 1.6 be shown tight for more general parameters (beyond constant $\mathsf{DL}^{\mathsf{dt}}(f)$ and logarithmic $\mathsf{DL}^{\mathsf{cc}}(F)$ as in Section 1.4)?

A. Appendix: Quadratic relationship between P^{NP} and DL

Proof (of Fact 1.5: $\Omega(\mathsf{DL}^{\mathsf{dt}}(f)) \leq \mathsf{P}^{\mathsf{NPdt}}(f) \leq O(\mathsf{DL}^{\mathsf{dt}}(f)^2 \cdot \log n)$).

For the first inequality, consider an optimal P^{NP} decision tree for f of cost k. Assume all the 0-edges point left and the 1-edges point right. We will generate a width-k conjunction decision list for f in phases, one phase for each leaf of the tree in right-to-left order (i.e., reverse lexicographic order of the bit strings formed by the edges along root-to-leaf paths). In each phase, say associated with some path $v_0, v_1, \ldots, v_h$ (where v_0 is the root and v_h is a leaf), we append to our decision list a set of conjunctions (ordered arbitrarily among themselves), each labeled with v_h's output. Specifically, the conjunctions associated with this path are each obtained by the following process: (i) for every v_i such that (v_i, v_{i+1}) is a 1-edge, choose a conjunction from the DNF queried by v_i, and (ii) if the conjunctions chosen in (i) are consistent with each other then form the conjunction of all of them and append it to the decision list. By the definition of $\mathsf{P}^{\mathsf{NPdt}}$ cost, each conjunction we append has width $\leq k$. If an input follows the path $v_0, v_1, \ldots, v_h$, then the first conjunction in the decision list that accepts it will indeed be from v_h's phase (hence have the correct label): the input is accepted by the DNFs queried by each v_i such that (v_i, v_{i+1}) is a 1-edge, and so is accepted by a conjunction in v_h's phase; furthermore, no conjunction from an earlier phase can accept the input since they would all include the literals of a conjunction from a DNF that rejects the input. Thus, the conjunction decision list we constructed is correct.

For the second inequality, consider an optimal conjunction decision list $(C_1, \ell_1), \ldots, (C_L, \ell_L)$ for f of width k (which necessarily has length $L \leq 2^k \binom{n}{k} \leq n^{O(k)}$). Our P^{NP} decision tree will perform a binary search to find the first conjunction C_i that accepts, then output ℓ_i. That is, the root will query the disjunction of the first half of the C_i's, $(C_1 \vee C_2 \vee \cdots \vee C_{L/2})$, the 1-child of the root will query the disjunction of the first quarter of the C_i's, the 0-child of the root will query the disjunction of the third quarter of the

C_i's, and so on. Since an execution consists of $O(k \cdot \log n)$ DNF queries, each of width $\leq k$, the cost of our P^{NP} decision tree for f is $O(k^2 \cdot \log n)$. $\square$

Proof (of Fact 1.6: $\Omega(\mathsf{DL}^{\mathsf{cc}}(F)) \leq \mathsf{P}^{\mathsf{NPcc}}(F) \leq O(\mathsf{DL}^{\mathsf{cc}}(F)^2)$).

The proof is very analogous to the proof of Fact 1.5 but with rectangles playing the role of conjunctions.

For the first inequality, consider an optimal P^{NP} protocol tree for F of cost k. Assume all the 0-edges point left and the 1-edges point right. We will generate a cost-$O(k)$ rectangle decision list for F in phases, one phase for each leaf of the tree in right-to-left order (i.e., reverse lexicographic order of the bit strings formed by the edges along root-to-leaf paths). In each phase, say associated with some path $v_0, v_1, \ldots, v_h$ (where v_0 is the root and v_h is a leaf), we append to our decision list a set of rectangles (ordered arbitrarily among themselves), each labeled with v_h's output. Specifically, the rectangles associated with this path are each obtained by the following process: (i) for every v_i such that (v_i, v_{i+1}) is a 1-edge, choose a rectangle from the union queried by v_i, and (ii) append the intersection of all the rectangles chosen in (i) to the decision list. (For the leftmost path, we take the "intersection of no rectangles" to be the whole domain of F.) By the definition of $\mathsf{P}^{\mathsf{NPcc}}$ cost, each phase contributes $\leq 2^k$ rectangles and there are $\leq 2^k$ phases, so the cost of the final rectangle decision list is $\leq 2k$. If an input follows the path $v_0, v_1, \ldots, v_h$, then the first rectangle in the decision list that contains it will indeed be from v_h's phase (hence have the correct label): the input is contained in the unions queried by each v_i such that (v_i, v_{i+1}) is a 1-edge, and so is contained in a rectangle in v_h's phase; furthermore, no rectangle from an earlier phase can contain the input since they would all be contained within a union that does not contain the input. Thus, the rectangle decision list we constructed is correct.

For the second inequality, consider an optimal rectangle decision list $(R_1, \ell_1), \ldots, (R_{2^k}, \ell_{2^k})$ for F. Our P^{NP} protocol tree will perform a binary search to find the first rectangle R_i that contains the input, then output ℓ_i. That is, the root will query the union of the first half of the R_i's, $(R_1 \cup R_2 \cup \cdots \cup R_{2^k/2})$, the 1-child of the root will query the union of the first quarter of the R_i's, the 0-child of the root will query the union of the third quarter of the R_i's, and so on. Since an execution consists of k oracle queries, each of cost $\leq k$, the cost of our P^{NP} protocol for F is $\leq k^2$. □

Acknowledgements

We thank anonymous reviewers for comments, especially for a suggestion that led to a simplified proof of Claim 3.6. We thank Paul Balister, Shalev Ben-David, Béla Bollobás, Robin Kothari, Nirman Kumar, Santosh Kumar, Govind Ramnarayan, Madhu Sudan, Li-Yang Tan, and Justin Thaler for discussions and correspondence. T.W. was supported by NSF grant CCF-1657377. A preliminary version of this work was published as Göös *et al.* (2017).

References

Scott Aaronson, Greg Kuperberg & Christopher Granade (2017). Complexity Zoo. Online. URL https://complexityzoo.uwaterloo.ca.

László Babai, Peter Frankl & Janos Simon (1986). Complexity Classes in Communication Complexity Theory. In *Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS)*, 337–347. IEEE.

Richard Beigel (1994). Perceptrons, PP, and the Polynomial Hierarchy. *Computational complexity* **4**(4), 339–349.

Richard Beigel, Nick Reingold & Daniel Spielman (1995). PP Is Closed under Intersection. *Journal of Computer and System Sciences* **50**(2), 191–202.

Andreas Blass & Yuri Gurevich (1982). On the Unique Satisfiability Problem. *Information and Control* **55**(1–3), 80–88.

Harry Buhrman, Nikolai Vereshchagin & Ronald de Wolf (2007). On Computation and Communication with Small Bias. In *Proceedings of the 22nd Conference on Computational Complexity (CCC)*, 24–32. IEEE.

HARRY BUHRMAN & RONALD DE WOLF (2002). Complexity Measures and Decision Tree Complexity: A Survey. *Theoretical Computer Science* **288**(1), 21–43.

MARK BUN & JUSTIN THALER (2018). Approximate Degree and the Complexity of Depth Three Circuits. In *Proceedings of the 22nd International Conference on Randomization and Computation (RANDOM)*. To appear.

SIU ON CHAN, JAMES LEE, PRASAD RAGHAVENDRA & DAVID STEURER (2016). Approximate Constraint Satisfaction Requires Large LP Relaxations. *Journal of the ACM* **63**(4), 34:1–34:22.

ARKADEV CHATTOPADHYAY, MICHAL KOUCKÝ, BRUNO LOFF & SAGNIK MUKHOPADHYAY (2017). Composition and Simulation Theorems via Pseudo-random Properties. Technical Report TR17-014, Electronic Colloquium on Computational Complexity (ECCC). URL https://eccc.weizmann.ac.il/report/2017/014/.

MIKA GÖÖS (2015). Lower Bounds for Clique vs. Independent Set. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, 1066–1076. IEEE.

MIKA GÖÖS, PRITISH KAMATH, TONIANN PITASSI & THOMAS WATSON (2017). Query-to-Communication Lifting for P^{NP}. In *Proceedings of the 32nd Computational Complexity Conference (CCC)*, 12:1–12:16. Schloss Dagstuhl.

MIKA GÖÖS, SHACHAR LOVETT, RAGHU MEKA, THOMAS WATSON & DAVID ZUCKERMAN (2016). Rectangles Are Nonnegative Juntas. *SIAM Journal on Computing* **45**(5), 1835–1869.

MIKA GÖÖS, TONIANN PITASSI & THOMAS WATSON (2018a). Deterministic Communication vs. Partition Number. *SIAM Journal on Computing* To appear.

MIKA GÖÖS, TONIANN PITASSI & THOMAS WATSON (2018b). The Landscape of Communication Complexity Classes. *Computational Complexity* **27**(2), 245–304.

JOHAN HÅSTAD, STASYS JUKNA & PAVEL PUDLÁK (1995). Top-Down Lower Bounds for Depth-Three Circuits. *Computational Complexity* **5**(2), 99–112. ISSN 1420-8954.

Hamed Hatami, Kaave Hosseini & Shachar Lovett (2018). Structure of Protocols for XOR Functions. *SIAM Journal on Computing* **47**(1), 208–217.

Russell Impagliazzo & Ryan Williams (2010). Communication Complexity with Synchronized Clocks. In *Proceedings of the 25th Conference on Computational Complexity (CCC)*, 259–269. IEEE.

Stasys Jukna (2012). *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer.

Mauricio Karchmer, Eyal Kushilevitz & Noam Nisan (1995). Fractional Covers and Communication Complexity. *SIAM Journal on Discrete Mathematics* **8**(1), 76–92.

Ker-I Ko (1990). Separating and Collapsing Results on the Relativized Probabilistic Polynomial-Time Hierarchy. *Journal of the ACM* **37**(2), 415–438.

Pravesh Kothari, Raghu Meka & Prasad Raghavendra (2017). Approximating Rectangles by Juntas and Weakly-Exponential Lower Bounds for LP Relaxations of CSPs. In *Proceedings of the 49th Symposium on Theory of Computing (STOC)*, 590–603. ACM.

Eyal Kushilevitz & Noam Nisan (1997). *Communication Complexity*. Cambridge University Press.

James Lee, Prasad Raghavendra & David Steurer (2015). Lower Bounds on the Size of Semidefinite Programming Relaxations. In *Proceedings of the 47th Symposium on Theory of Computing (STOC)*, 567–576. ACM.

Periklis Papakonstantinou, Dominik Scheder & Hao Song (2014). Overlays and Limited Memory Communication. In *Proceedings of the 29th Conference on Computational Complexity (CCC)*, 298–308. IEEE.

Ramamohan Paturi & Janos Simon (1986). Probabilistic Communication Complexity. *Journal of Computer and System Sciences* **33**(1), 106–123.

Anup Rao & Amir Yehudayoff (2017). *Communication Complexity*. In preparation.

Ran Raz & Pierre McKenzie (1999). Separation of the Monotone NC Hierarchy. *Combinatorica* **19**(3), 403–435.

Alexander Razborov & Alexander Sherstov (2010). The Sign-Rank of AC^0. *SIAM Journal on Computing* **39**(5), 1833–1855.

Susanna de Rezende, Jakob Nordström & Marc Vinyals (2016). 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)*, 295–304. IEEE.

Ronald Rivest (1987). Learning Decision Lists. *Machine Learning* **2**(3), 229–246.

Robert Robere, Toniann Pitassi, Benjamin Rossman & Stephen Cook (2016). Exponential Lower Bounds for Monotone Span Programs. In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, 406–415. IEEE.

Miklos Santha (1989). Relativized Arthur–Merlin versus Merlin–Arthur Games. *Information and Computation* **80**(1), 44–49.

Rocco Servedio, Li-Yang Tan & Justin Thaler (2012). Attribute-Efficient Learning and Weight-Degree Tradeoffs for Polynomial Threshold Functions. In *Proceedings of the 25th Conference on Learning Theory (COLT)*, 14.1–14.19. JMLR. URL http://www.jmlr.org/proceedings/papers/v23/servedio12/servedio12.pdf.

Alexander Sherstov (2011). The Pattern Matrix Method. *SIAM Journal on Computing* **40**(6), 1969–2000.

Yaoyun Shi & Yufan Zhu (2009). Quantum Communication Complexity of Block-Composed Functions. *Quantum Information and Computation* **9**(5–6), 444–460.

Justin Thaler (2016). Lower Bounds for the Approximate Degree of Block-Composed Functions. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, 17:1–17:15. Schloss Dagstuhl.

Salil Vadhan (2012). Pseudorandomness. *Foundations and Trends in Theoretical Computer Science* **7**(1–3), 1–336.

NIKOLAI VERESHCHAGIN (1999). Relativizability in Complexity Theory. In *Provability, Complexity, Grammars*, volume 192 of *AMS Translations, Series 2*, 87–172. American Mathematical Society.

RYAN WILLIAMS (2001). Brute Force Search and Oracle-Based Computation. Technical report, Cornell University. URL https://web.stanford.edu/~rrwill/bfsearch-rev.ps.

XIAODI WU, PENGHUI YAO & HENRY YUEN (2017). Raz–McKenzie Simulation with the Inner Product Gadget. Technical Report TR17-010, Electronic Colloquium on Computational Complexity (ECCC). URL https://eccc.weizmann.ac.il/report/2017/010/.

MIHALIS YANNAKAKIS (1991). Expressing Combinatorial Optimization Problems by Linear Programs. *Journal of Computer and System Sciences* **43**(3), 441–466.

Manuscript received 21 July 2017

MIKA GÖÖS
Institute for Advanced Study, Princeton, USA
mika@ias.edu

PRITISH KAMATH
Massachusetts Institute of Technology, Cambridge, USA
pritish@mit.edu

TONIANN PITASSI
University of Toronto, Toronto, Canada
toni@cs.toronto.edu

THOMAS WATSON
University of Memphis, Memphis, USA
Thomas.Watson@memphis.edu