THEORY OF COMPUTING, Volume 19 (9), 2023, pp. 1–35
www.theoryofcomputing.org

Optimal Composition Theorem for Randomized Query Complexity

Dmytro Gavinsky*† Troy Lee‡ Miklos Santha
Swagato Sanyal§

Received January 11, 2021; Revised December 29, 2022; Published December 30, 2023

Abstract. For any set S, any relation $f \subseteq \{0,1\}^n \times S$, and any partial Boolean function g defined on a subset of $\{0,1\}^m$, we show that

$$\mathbb{R}_{1/3}(f \circ g^n) \in \Omega\left(\mathbb{R}_{4/9}(f) \cdot \sqrt{\mathbb{R}_{1/3}(g)}\right),$$

where $\mathbb{R}_\epsilon(\cdot)$ stands for the *bounded-error randomized query complexity* with error at most ϵ, and $f \circ g^n \subseteq (\{0,1\}^m)^n \times S$ denotes the composition of f with n instances of g. This result is new even in the special case when $S = \{0,1\}$ and g is a total function.

We show that the new composition theorem is *optimal* for the general case of relations: A relation f_0 and a partial Boolean function g_0 are constructed, such that $\mathbb{R}_{4/9}(f_0) \in \Theta(\sqrt{n})$, $\mathbb{R}_{1/3}(g_0) \in \Theta(n)$ and $\mathbb{R}_{1/3}(f_0 \circ g_0^n) \in \Theta(n)$.

A preliminary version of this paper appeared in the Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP'19) [10].

*In 2022 D. G. changed his first name, as explained on his Internet page.

†Partially funded by the grant 19-27871X of GA ČR.

‡Supported by the Australian Research Council (Grant No: DP200100950).

§Supported by an ISIRD Grant from Sponsored Research and Industrial Consultancy, IIT Kharagpur.

ACM Classification: F.1.2, F.2.3

AMS Classification: 68Q17, 68W20

Key words and phrases: query complexity, randomized decision tree, composed function, lower bound

© 2023 Dmytro Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal
Licensed under a Creative Commons Attribution License (CC-BY)
DOI: 10.4086/toc.2023.v019.a009

The theorem is proved via introducing a new complexity measure, *max-conflict complexity*, denoted by $\bar{\chi}(\cdot)$. Its investigation shows that $\bar{\chi}(g) \in \Omega(\sqrt{\mathbb{R}_{1/3}(g)})$ for any partial Boolean function g and $\mathbb{R}_{1/3}(f \circ g^n) \in \Omega(\mathbb{R}_{4/9}(f) \cdot \bar{\chi}(g))$ for any relation f, which readily implies the composition statement. It is further shown that $\bar{\chi}(g)$ is always at least as large as the *sabotage complexity* of g (introduced by Ben-David and Kothari in 2016).

1 Introduction

Let S be a finite set, $|S| \geq 2$. An *S-relation* in n Boolean variables is defined to be a subset $f \subseteq \{0,1\}^n \times S$ such that for every $x \in \{0,1\}^n$ there exists a $s \in S$ such that $(x,s) \in f$. (For motivation see Section 2.) If $S = \{0,1\}$, we call the S-relation a *Boolean relation.* A *partial Boolean function* in m variables is a function $g : T \to \{0,1\}$, where $T \subseteq \{0,1\}^m$. We will identify g with the Boolean relation $\{(x, g(x)) \mid x \in T\} \cup \{(x,b) \mid x \notin T, b \in \{0,1\}\} \subseteq \{0,1\}^m \times \{0,1\}$. g is called a *total Boolean function* if $T = \{0,1\}^m$. The *composition* of an S-relation f and a partial Boolean function g is the S-relation $f \circ g^n \subseteq (\{0,1\}^m)^n \times S$ defined as follows. A tuple $(x_1, \ldots, x_n, s)$ is in $f \circ g^n$ if and only if one of the following two conditions holds:

1. There exists an i such that x_i is not a valid input of g, i. e., $x_i \notin T$.

2. Each x_i is a valid input of g and $((g(x_1), \ldots, g(x_n)), s) \in f$.

Note that if f is a partial Boolean function in n Boolean variables with domain $R \subseteq \{0,1\}^n$ (i. e., $f : R \to \{0,1\}$), then $f \circ g^n$ is a partial Boolean function in mn Boolean variables with domain $\{(x_1, \ldots, x_n) \in T^n \mid (g(x_1), \ldots, g(x_n)) \in R\} \subseteq (\{0,1\}^m)^n$.

A partial Boolean function models a decision task in computer science, whose domain is the set of valid binary encodings of inputs to the task. An S-relation models a search problem, where each input may correspond to more than one items in a search space (e. g., a Boolean formula may have more than one satisfying assignments).

Relating the complexity of $f \circ g^n$ to the complexities of f and g is a natural research problem. A *query algorithm* for computing an S-relation h is allowed to query *individual bits* of the input x, with the goal of outputting an element s such that $(x,s) \in h$. The *query complexity of an algorithm* is the maximum possible number of queries that it makes.

Query algorithms can be *deterministic*, *randomized* or *quantum*, where the latter two classes allow for (bounded) errors. The corresponding *query complexity of an S-relation h* – denoted, respectively, by $\mathbb{D}(h)$, $\mathbb{R}(h)$ or $\mathbb{Q}(h)$ – is the minimal query complexity of an algorithm that belongs to the corresponding class and computes h with error $1/3$. [1] Section 2 contains formal definitions of various query complexity measures.

It is easy to see that $\mathbb{D}(f \circ g^n) \leq \mathbb{D}(f) \cdot \mathbb{D}(g)$. [2] For the cases of randomized and quantum

[1] $\mathbb{R}_\epsilon(h)$ stands for the ϵ-error randomized query complexity, and $\mathbb{Q}_\epsilon(h)$ denotes the ϵ-error quantum query complexity.

[2] To compute $f \circ g^n$, one can simulate an optimal query algorithm for f, serving every query of this algorithm by running an optimal query algorithm for g.

query complexity the argument is slightly more subtle, though very similar conceptually; in particular, both $\mathbb{R}(f \circ g^n) \in O(\mathbb{R}(f) \cdot \mathbb{R}(g) \cdot \log \mathbb{R}(f))$ and $\mathbb{Q}(f \circ g^n) \in O(\mathbb{Q}(f) \cdot \mathbb{Q}(g))$ hold. [3]

Showing strong *lower bounds* on the query complexity of $f \circ g^n$ (preferably, matching the trivial upper bound) is often more interesting, the corresponding statements are sometimes called *composition theorems*. Such results can lead to further theoretical developments (e. g., separating complexity measures, as well as different classes in structural complexity).

For deterministic query complexity it has been shown [14, 17] that

$$\mathbb{D}(f \circ g^n) = \mathbb{D}(f) \cdot \mathbb{D}(g),$$

which means that *the trivial query algorithm for $f \circ g^n$ described above is optimal*. Similarly, for bounded-error quantum query complexity it has been shown [12, 15] that

$$\mathbb{Q}(f \circ g^n) \in \Theta(\mathbb{Q}(f) \cdot \mathbb{Q}(g)).$$

Prior to this work, the randomized query complexity of composition has remained an open problem, even for the special case where $f \subseteq \{0,1\}^n \times \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ are total Boolean functions. We partially solve it for the most general case of composition, namely, letting f be an *S-relation* and g be a *partial Boolean function*.

Theorem 1.1. *For any S-relation $f \subseteq \{0,1\}^n \times S$ and any partial Boolean function $g \subseteq \{0,1\}^m \times \{0,1\}$,*

$$\mathbb{R}_{1/3}(f \circ g^n) \in \Omega\left(\mathbb{R}_{4/9}(f) \cdot \sqrt{\mathbb{R}_{1/3}(g)}\right).$$

Note that the above lower bound *does not match the trivial upper bound*, so we address its optimality separately. We do this by constructing an example where the above bound is tight. In other words, while some *incomparable* lower bounds on $\mathbb{R}_{1/3}(f \circ g^n)$ are conceivable, the statement of Theorem 1.1 is *a strongest possible statement in general*. [4]

Theorem 1.2. *There exists a relation $f_0 \subseteq \{0,1\}^n \times \{0,1\}^n$ and a partial Boolean function $g_0 \subseteq \{0,1\}^n \times \{0,1\}$, such that*

$$\mathbb{R}_{4/9}(f_0) \in \Theta(\sqrt{n}),\ \mathbb{R}_{1/3}(g_0) \in \Theta(n) \text{ and } \mathbb{R}_{1/3}(f_0 \circ g_0^n) \in \Theta(n).$$

Theorem 1.2 shows that a perfect composition theorem does not hold when f is an S-relation for an arbitrary finite set S and g is any partial Boolean function. After the conference publication of the current work, Ben-David and Blais [5] exhibited partial Boolean functions f and g such that $\mathbb{R}(f \circ g^n) \in \widetilde{O}(\mathbb{R}(f)^{2/3} \cdot \mathbb{R}(g)^{2/3})$; their example showed that a perfect composition theorem fails to hold even when both f and g are partial Boolean functions. However, in their example, the randomized query complexities grow logarithmically with the respective number of variables, which leaves the following possibility open (mentioned as a conjecture in [5]).

[3] The multiplicative factor of $\log \mathbb{R}(f)$ in the case of $\mathbb{R}(h)$ is due to the need to reduce the error in computing each instance of g to $O(\frac{1}{\mathbb{R}(f)})$; in the quantum case this can be handled in a more elegant, "lossless" way.

[4] The following construction also witnesses the possibility of $\mathbb{R}(f \circ g^n) \in O(\mathbb{R}(g))$ when $\mathbb{R}(f) \in \Omega(\sqrt{n})$ – in other words, *it is true in general that composition with a "hard" Boolean relation makes a Boolean function harder for randomized query algorithms*.

Conjecture 1.3 (Ben-David and Blais [5]). *Let f and g be any partial Boolean functions, and n be the number of variables of f. Then*

$$\mathbb{R}(f \circ g^n) = \frac{\Omega(\mathbb{R}(f) \cdot \mathbb{R}(g))}{\log n}.$$

1.1 Our approach

We introduce a new complexity measure of Boolean functions, the *max-conflict complexity*, denoted by $\bar{\chi}(g)$. We show that $\bar{\chi}(g)$ is a quadratically tight lower bound on the randomized query complexity of a (partial) function g.

Theorem 1.4. *For any partial Boolean function $g \subseteq \{0,1\}^m \times \{0,1\}$,*

$$\bar{\chi}(g) \in \Omega\left(\sqrt{\mathbb{R}_{1/3}(g)}\right).$$

Theorem 1.4 is tight for the function g_0 in Theorem 1.2. The main technical ingredient of this article is the following composition statement for the max-conflict complexity.

Theorem 1.5. *For any S-relation $f \subseteq \{0,1\}^n \times S$ and any partial Boolean function $g \subseteq \{0,1\}^m \times \{0,1\}$,*

$$\mathbb{R}_{1/3}(f \circ g^n) \in \Omega\left(\mathbb{R}_{4/9}(f) \cdot \bar{\chi}(g)\right).$$

Theorem 1.4 and Theorem 1.5 together imply Theorem 1.1.

1.2 Prior work

In the special case of f being a partial function and g being a total one, significant progress has been made by Ben-David and Kothari [7], who showed that

$$\mathbb{R}_{1/3}(f \circ g^n) \in \Omega\left(\mathbb{R}_{1/3}(f) \cdot \sqrt{\frac{\mathbb{R}_0(g)}{\log \mathbb{R}_0(g)}}\right). \tag{1.1}$$

To prove the above statement, the authors have introduced and investigated a new complexity measure of Boolean functions, *sabotage complexity*, denoted by $\mathbb{RS}(g)$. The notion has a very natural definition and is of independent interest. The authors have shown that for any partial functions f, g, $\mathbb{R}_{1/3}(f \circ g^n) = \Omega(\mathbb{R}_{1/3}(f) \cdot \mathbb{RS}(g))$. The authors have further shown that if g is a total Boolean function, then $\mathbb{RS}(g) = \Omega\left(\sqrt{\frac{\mathbb{R}_0(g)}{\log \mathbb{R}_0(g)}}\right)$. This implies (1.1). We note here that for partial functions g, $\mathbb{RS}(g)$ can be arbitrarily smaller than $\mathbb{R}(g)$; the authors have shown a family of partial functions (the collision problem) for which $\mathbb{RS}(g) = O(1)$, but $\mathbb{R}(g) = \Theta(\sqrt{n})$, where n is the size of the input.

In this article we show that max-conflict complexity is always lower-bounded by the sabotage complexity of the same function.

Theorem 1.6. *For any partial Boolean function $g \subseteq \{0,1\}^m \times \{0,1\}$,*

$$\bar{\chi}(g) \geq \mathbb{RS}(g).$$

Since f is a Boolean function, $\mathbb{R}_{1/3}(f) = \Theta(\mathbb{R}_{4/9}(f))$; hence Theorem 1.6 along with Theorem 1.5 imply (1.1).

1.3 Proof technique

At a high level, the proof of Theorem 1.5 follows the structure of the proof by Anshu et al. [3] and Ben-David and Kothari [7]. We show that for every probability distribution η over the input space $\{0,1\}^n$ of f, there exists a deterministic query algorithm $\mathcal{A}$ that makes $O(\mathbb{R}_{1/3}(f \circ g^n)/\sqrt{\mathbb{R}_{1/3}(g)})$ queries in the worst case, and computes f with high probability, $\Pr_{z\sim\eta}[(z, \mathcal{A}(z)) \in f] \geq 5/9$. By the minimax principle (Fact 2.4) this implies Theorem 1.5.

We do this by using a query algorithm for $f \circ g^n$ to construct a query algorithm for f. We define a sampling procedure that for any $z \in \{0,1\}^n$ samples $x = (x_1, \ldots, x_n)$ such that $(z, s) \in f$ if and only if $(x, s) \in f \circ g^n$. This procedure is defined in terms of Q, which is a probability distribution over pairs of distributions (μ_0, μ_1), where μ_0 is supported on $g^{-1}(0)$ and μ_1 is supported on $g^{-1}(1)$. We define a distribution γ_η over $(\{0,1\}^m)^n$ in terms of this sampling process as follows:

1. Sample $z = (z_1, \ldots, z_n)$ from $\{0,1\}^n$ according to η.
2. Independently sample $(\mu_0^{(i)}, \mu_1^{(i)})$ from Q for $i = 1, \ldots, n$.
3. Sample $x_i = (x_i^{(1)}, \ldots, x_i^{(m)})$ according to $\mu_{z_i}^{(i)}$ for $i = 1, \ldots, n$. Return $x = (x_1, \ldots, x_n)$.

Notice that steps (1) and (2) are independent and the order in which they are performed does not matter. For future reference, for a fixed z let $\gamma_z(Q)$ be the probability distribution defined by the last two steps.

Now γ_η is simply a probability distribution over $(\{0,1\}^m)^n$. Thus by the minimax principle (Fact 2.4 below), there is a deterministic query algorithm $\mathcal{A}'$ of worst-case complexity at most $\mathbb{R}_{1/3}(f \circ g^n)$ such that $\Pr_{x\sim\gamma_\eta}[(x, \mathcal{A}'(x)) \in f \circ g^n] \geq 2/3$. We first use $\mathcal{A}'$ to construct a randomized query algorithm T for f with bounded *expected* query complexity and error at most $1/3$. T is presented formally in Algorithm 3. The final algorithm $\mathcal{A}$ will be a truncation of T which has bounded worst-case complexity and error at most $4/9$.

On input z, the algorithm T seeks to sample a string x from $\gamma_z(Q)$, and run $\mathcal{A}'$ on x. Put another way, $\gamma_z(Q)$ induces a probability distribution over the leaves of $\mathcal{A}'$, and the goal of T is to sample a leaf of $\mathcal{A}'$ according to this distribution. Since for each $s \in S$, $(x, s) \in f \circ g^n$ if and only if $(z, s) \in f$, and $\Pr_{x\sim\gamma_\eta}[(x, \mathcal{A}'(x)) \in f \circ g^n] \geq 2/3$, we have that $\Pr_{z\sim\eta}[(z, T(z)) \in f] \geq 2/3$. Thus T meets the accuracy requirement.

The catch, of course, is to specify how T samples from $\gamma_z(Q)$ *without making too many queries to* z. To sample x_i from $\mu_{z_i}^{(i)}$ seems to require knowledge of z_i, and thus T would have to query all of z.

To bypass this problem, we remember that $\mathcal{A}'$, being an efficient algorithm, will query only a few bits of x. This allows us to sample x bit by bit as and when they are queried by $\mathcal{A}'$. To see this more clearly, consider a run of T where the pairs of distributions $(\mu_0^{(1)}, \mu_1^{(1)}), \ldots, (\mu_0^{(n)}, \mu_1^{(n)})$ were chosen in step (2) of the sampling procedure. Suppose that T is trying to simulate $\mathcal{A}'$ at a vertex v where $x_i^{(j)}$ is queried. To respond to this query, T will sample $x_i^{(j)}$ from its marginal distribution according to $\mu_{z_i}^{(i)}$ conditioned on the event $x \in v$. Let the following be the marginal distributions of $x_i^{(j)}$ for the two possible values of z_i.

	$\Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x \in v]$	$\Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 1 \mid x \in v]$
$z_i = 0$	p_0	$1 - p_0$
$z_i = 1$	p_1	$1 - p_1$

Without loss of generality, assume that $p_0 \leq p_1$. T answers the query by the procedure BITSAMPLER given in Algorithm 1. Note that the bit returned by BITSAMPLER has the desired distribution.

Algorithm 1: BITSAMPLER (suppose $p_0 \leq p_1$)

```
1  Sample r ~ [0,1] uniformly at random.
2  if r < p_0 then
3      return 0.
4
5  else if r > p_1 then
6      return 1.
7
8  else
9      query z_i.
10     if r ≤ p_{z_i} then
11         return 0.
12     else
13         return 1.
```

The step in which BITSAMPLER returns the bit depends on the value of r sampled in step 1. In particular, z_i is queried if and only if $r \in [p_0, p_1]$, and the bit is returned in step 11 or 13. Such a query to z_i contributes to the query complexity of T. Thus the probability that T makes a query when the underlying simulation of $\mathcal{A}'$ is at vertex v is $(p_1 - p_0)$. We refer to this quantity as $\Delta(v)$. It plays an important role in our analysis (see Section 6 and Appendix 6.1).

Our sampling procedure and the tools we use to bound its cost is reminiscent of work of Barak, Braverman, Chen and Rao [4] in communication complexity. They look at a communication analog of our setting where two players are trying to sample a leaf in a communication protocol while communicating as little as possible.

1.3.1 Conflict complexity and max-conflict complexity

Bounding the query complexity of T naturally suggests the quantities that we define in this article: the conflict complexity $\chi(g)$ and the max-conflict complexity $\bar{\chi}(g)$ of a partial Boolean function g. A formal definition can be found in Section 3; here we give the high-level idea and motivation behind these quantities.

Let us set aside T for a moment and just consider a deterministic query algorithm $\mathcal{B}$ computing the partial function $g \subseteq \{0,1\}^m \times \{0,1\}$. Let μ_0 and μ_1 be distributions with support on $g^{-1}(0)$ and $g^{-1}(1)$, respectively. For each vertex $v \in \mathcal{B}$ let $p_0(v)$ and $p_1(v)$ be the probability that the answer to the query at v is 0 on input $x \sim \mu_0$ and $x \sim \mu_1$, respectively, conditioned on x reaching v. Now we can imagine a process $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$ that runs BITSAMPLER on the tree $\mathcal{B}$: $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$ begins at the root, and at a vertex v in $\mathcal{B}$ it uniformly chooses a random real number $r \in [0,1]$. If $r < \min\{p_0(v), p_1(v)\}$ then the query is "answered" 0 and it moves to the left child. If $r > \max\{p_0(v), p_1(v)\}$ then the query is "answered" 1 and it moves to the right child. If $r \in [\min\{p_0(v), p_1(v)\}, \max\{p_0(v), p_1(v)\}]$ then the process halts. The conflict complexity $\chi(\mathcal{B}, (\mu_0, \mu_1))$ is the expected number of vertices this process visits before halting. The conflict complexity of g is defined to be

$$\chi(g) = \max_{(\mu_0,\mu_1)} \min_{T} \chi(T, (\mu_0, \mu_1)) \ ,$$

where the minimum is taken over trees T that compute g. For *max-conflict complexity* we enlarge the set over which we maximize. Let Q be a distribution with finite support over pairs of distributions (μ_0, μ_1), where $\mathrm{supp}(\mu_0) \subseteq g^{-1}(0), \mathrm{supp}(\mu_1) \subseteq g^{-1}(1)$ for each pair (μ_0, μ_1) in the support of Q. Let $\chi(\mathcal{B}, Q) = \mathbb{E}_{(\mu_0,\mu_1)\sim Q}\left[\chi(\mathcal{B}, (\mu_0, \mu_1))\right]$. The max-conflict complexity $\bar{\chi}(g)$ is defined as

$$\bar{\chi}(g) = \sup_{Q} \min_{T} \chi(T, Q) \ ,$$

where the minimum is taken over trees T that compute g. Clearly, the max-conflict complexity is at least as large as the conflict complexity.

To motivate the max-conflict complexity, note that the query complexity of T is the number of times step 9 in BITSAMPLER is executed, i. e., when the random number $r \in [p_0, p_1]$. In the definition of T we will choose Q to achieve close to the optimal value in the definition of $\bar{\chi}(g)$. Then intuitively one expects that for each i, T queries z_i only after $\mathcal{A}'$ makes about $\bar{\chi}(g)$ queries into x_i. By means of a direct sum theorem for max-conflict complexity we make this intuition rigorous and prove that the expected query complexity of T is at most $\mathbb{R}_{1/3}(f \circ g^n)/\bar{\chi}(g)$. We refer the reader to Section 5 for a formal proof.

1.3.2 $\bar{\chi}(g)$ and $\mathbb{R}(g)$

Note that applying Theorem 1.5 with the outer function $f(z) = z_1$ shows that $\mathbb{R}_{1/3}(g) \in \Omega(\bar{\chi}(g))$.[5] We complete the proof of Theorem 1.1 by showing that max-conflict complexity is a quadratically

[5] $\mathbb{R}_{1/3}(g) \in \Omega(\bar{\chi}(g))$ also follows fairly easily from the defintions of $\mathbb{R}_{1/3}(g)$ and $\bar{\chi}(g)$).

tight lower bound on randomized query complexity, even for partial functions g. In fact, we show the stronger result that this is true even for the conflict complexity.

Theorem 1.7. *For any partial Boolean function $g \subseteq \{0,1\}^m \times \{0,1\}$,*

$$\chi(g) \in \Omega\left(\sqrt{\mathbb{R}_{1/3}(g)}\right).$$

Theorem 1.7 is proved in Section 6. At a high level, our proof is reminiscent of the result of [4] on compressing communication protocols in that both look at a random sampling process to navigate a tree, and relate the probability of this process needing to query or communicate at a node to the amount of information that is learned at the node.

To prove $\mathbb{R}(g) \in O(\chi(g)^2)$, we again resort to the minimax principle; we show that for each probability distribution μ over the valid inputs to g, there is an accurate and efficient distributional query algorithm for g. For $b \in \{0,1\}$, let μ_b be the distribution obtained by conditioning μ on the event $g(x) = b$. By the definition of $\chi(g)$, there is a query algorithm $\mathcal{B}$ such that the following is true: if its queries are served by BITSAMPLER, step 9 is executed within expected $\chi(\mathcal{B}, \mu_0, \mu_1) \leq \chi(g)$ queries. Note that at a vertex v which queries i, the probability that step 9 is executed is $\Delta(v) = |\Pr_{\mu_0}[x_i = 0 \mid x \text{ at } v] - \Pr_{\mu_1}[x_i = 0 \mid x \text{ at } v]|$. This roughly implies that for a typical vertex v of $\mathcal{B}$, $\Delta(v)$ is at least about $\frac{1}{\chi(g)}$. By a technical claim that we prove (Claim 6.4) this implies that the query outcome at v carries about $\frac{1}{\chi(g)^2}$ bits of information about $g(x)$. Using the *chain rule of mutual information*, we can show that the mutual information between $g(x)$ and the outcomes of the first $O(\chi(g))^2$ queries by $\mathcal{B}$ is $\Omega(1)$. This enables us to conclude that we can infer the value of $g(x)$ with success probability $1/2 + \Omega(1)$ from the transcript of $\mathcal{B}$ restricted to the first $O(\chi(g)^2)$ queries. The distributional algorithm of g for μ is simply the algorithm $\mathcal{B}$ terminated after $O(\chi(g)^2)$ queries.

1.3.3 $\bar{\chi}(g)$ and $\mathbb{RS}(g)$

To see why $\bar{\chi}(g) \geq \mathbb{RS}(g)$, we first give an alternative characterization of $\mathbb{RS}(g)$. For a deterministic tree T computing g and strings x, y such that $g(x) \neq g(y)$, let $\mathrm{sep}_T(x,y)$ be the depth of the node v in T such that x and y both reach v yet $x_{q(v)} \neq y_{q(v)}$, where $q(v)$ is the index queried at v. Let $\mathcal{T}$ be a zero-error randomized protocol for g, i. e., $\mathcal{T}$ is a probability distribution supported on deterministic trees that compute g. Then we have (for a proof see Appendix B)

$$\mathbb{RS}(g) = \min_{\mathcal{T}} \max_{\substack{x,y \\ g(x)\neq g(y)}} \mathbb{E}_{T\sim\mathcal{T}}[\mathrm{sep}_T(x,y)] \ .$$

By von Neumann's minimax theorem [18], this is equal to

$$\mathbb{RS}(g) = \max_{p} \min_{T} \mathbb{E}_{(x,y)\sim p}[\mathrm{sep}_T(x,y)] \ .$$

Here, the max is taken over distributions p on pairs (x,y) where $g(x) \neq g(y)$, and the min is taken over deterministic trees T computing g.

We have seen that the definition of $\bar{\chi}(g)$ is

$$\bar{\chi}(g) = \sup_{Q} \min_{T} \mathbb{E}_{(\mu_0,\mu_1)\sim Q} \left[\chi(T,(\mu_0,\mu_1))\right] \ ,$$

where Q is a distribution with finite support over pairs (μ_0, μ_1) and T is a deterministic tree computing g. When (μ_0, μ_1) are taken to be singleton distributions, i. e., μ_0 puts all its weight on a single x with $g(x) = 0$, and μ_1 puts all its weight on a single y with $g(y) = 1$, it is easy to see that $\chi(T, (\mu_0, \mu_1)) = \mathrm{sep}_T(x, y)$ (see Claim 3.4). Since there are only finitely many such pairs (x, y), we have that $\bar{\chi}(g)$ is at least as large as the sabotage complexity of g.

1.4 The conference version

A preliminary version of this paper appeared in the proceedings of ICALP 2019 [10]. Among other revisions, some terminological confusion in the conference version is cleared up in this paper.

1.5 Follow-up work

As mentioned before, after the conference publication of the present work [10], Ben-David and Blais [5] exhibited partial Boolean functions f and g such that $\mathbb{R}(f \circ g^n) \in \widetilde{O}(\mathbb{R}(f)^{2/3} \cdot \mathbb{R}(g)^{2/3})$, ruling out the possibility of a perfect composition theorem even when both f and g are partial Boolean functions. Later, Ben-David, Blais, Göös and Maystre [6] introduced the *linearized complexity measure* $\mathsf{LR}(\cdot)$. They showed the composition theorem $\mathbb{R}(f \circ g^n) = \Omega(\mathbb{R}(f) \cdot \mathsf{LR}(g))$ for all partial Boolean functions f and g. They further showed that $\mathsf{LR}(g)$ is the asymptotically largest measure for which such a composition statement holds. Thus their composition theorem is an improvement on Theorem 1.5.

2 Preliminaries

For $T \subseteq \{0,1\}^m$, let $g : T \to \{0,1\}$ be a partial Boolean function. For $b \in \{0,1\}$, $g^{-1}(b)$ is defined to be the set of strings $\{x \in T \mid g(x) = b\}$. We refer to T as the set of valid inputs to g. For all strings $y \notin T$, we say that $g(y) = *$. All probability distributions μ over $\{0,1\}^m$ considered in connection with g are assumed to have support on T. Thus $g(x)$ is well-defined for any x in the support of μ.

Let S be any finite set. Let $h \subseteq \{0,1\}^k \times S$ be any S-relation (which could also be a partial Boolean function). For the sake of simplicity throughout the paper we will assume that for every $x \in \{0,1\}^k$ there exists an $s \in S$ such that $(x, s) \in h$; the case in which there exists an $x \in \{0,1\}^k$ such that for all $s \in S$ $(x, s) \notin h$, can be easily handled by including all pairs in $\{(x, s) \mid s \in S\}$ in h. Consider query algorithms $\mathcal{A}$ that accept a string $x \in \{0,1\}^k$ as input, query various bits of x, and produce an element of S as output. We denote the output by $\mathcal{A}(x)$.

Definition 2.1 (Deterministic query complexity)**.** A deterministic query algorithm $\mathcal{A}$ is said to compute an S-relation h if $(x, \mathcal{A}(x)) \in h$ for all $x \in \{0,1\}^k$. The deterministic query complexity

$\mathbb{D}(h)$ of h is the minimum over all deterministic query algorithms $\mathcal{A}$ computing h of the maximum number of queries made by $\mathcal{A}$ over $x \in \{0,1\}^k$.

Definition 2.2 (Bounded-error randomized query complexity)**.** Let $\epsilon \in [0, 1/2)$. We say that a randomized query algorithm $\mathcal{A}$ computes an S-relation h with error ϵ if $\Pr[(x, \mathcal{A}(x)) \in h] \geq 1-\epsilon$ for all $x \in \{0,1\}^k$. The bounded-error randomized query complexity $\mathbb{R}_\epsilon(h)$ of h is the minimum over all randomized query algorithms $\mathcal{A}$ computing h with error ϵ of the maximum number of queries made by $\mathcal{A}$ over all $x \in \{0,1\}^k$ and the internal randomness of $\mathcal{A}$.

Definition 2.3 (Distributional query complexity)**.** Let μ a distribution on the input space $\{0,1\}^k$ of h, and $\epsilon \in [0, 1/2)$. We say that a deterministic query algorithm $\mathcal{A}$ computes an S-relation h with distributional error ϵ on μ if $\Pr_{x\sim\mu}[(x, \mathcal{A}(x)) \in h] \geq 1 - \epsilon$. The distributional query complexity $\mathbb{D}^\mu_\epsilon(h)$ of h is the minimum over deterministic algorithms $\mathcal{A}$ computing h with distributional error ϵ on μ of the maximum over $x \in \{0,1\}^k$ of the number of queries made by $\mathcal{A}$ on x.

We will use the minimax principle in our proofs to go between distributional and randomized query complexity.

Fact 2.4 (Minimax principle)**.** *For any integer $k > 0$, finite set S, and S-relation $h \subseteq \{0,1\}^k \times \mathcal{S}$,*

$$\mathbb{R}_\epsilon(h) = \max_\mu \mathbb{D}^\mu_\epsilon(h).$$

We present a proof of Fact 2.4 in Appendix A.

Let μ be a probability distribution over $\{0,1\}^k$. We use $\mathrm{supp}(\mu)$ to denote the support of μ. By $x \sim \mu$ we mean that x is a random string drawn from μ. Let $C \subseteq \{0,1\}^k$ be an arbitrary set such that $\Pr_{x\sim\mu}[x \in C] = \sum_{y\in C} \mu(y) > 0$. Then $\mu \mid C$ is defined to be the probability distribution obtained by conditioning μ on the event that the sampled string belongs to C, i. e.,

$$(\mu \mid C)(x) = \begin{cases} 0 & \text{if } x \notin C \\ \frac{\mu(x)}{\sum_{y\in C}\mu(y)} & \text{if } x \in C \end{cases}$$

For a distribution Q over pairs of distributions (μ_0, μ_1), let $\mathrm{supp}_0(Q) = \cup\{\mathrm{supp}(\mu_0) : \exists \mu_1, (\mu_0, \mu_1) \in \mathrm{supp}(Q)\}$. Similarly let $\mathrm{supp}_1(Q) = \cup\{\mathrm{supp}(\mu_1) : \exists \mu_0, (\mu_0, \mu_1) \in \mathrm{supp}(Q)\}$. We say that Q is *consistent* if $\mathrm{supp}_0(Q)$ and $\mathrm{supp}_1(Q)$ are disjoint sets. We say that Q is consistent with a (partial) function g if $\mathrm{supp}_0(Q) \subseteq g^{-1}(0)$ and $\mathrm{supp}_1(Q) \subseteq g^{-1}(1)$. All such distributions Q considerd in this paper will be assumed to have finite support.

Definition 2.5 (Subcube, codimension)**.** A subset $\mathbb{C} \subseteq \{0,1\}^m$ is called a subcube if there exists a set $S \subseteq \{1, \ldots, m\}$ of indices and an *assignment function* $\sigma : S \to \{0,1\}$ such that $\mathbb{C} = \{x \in \{0,1\}^m : \forall i \in S, x_i = \sigma(i)\}$. The codimension $\mathrm{codim}(\mathbb{C})$ of $\mathbb{C}$ is defined to be $|S|$.

The composition of an S-relation and a partial Boolean function plays a central role in this paper. See Section 1 for a definition.

We will often view a deterministic query algorithm as a binary decision tree. In each vertex v of the tree, an input variable is queried. Depending on the outcome of the query, the computation goes to a child of v. The child of v corresponding to outcome b of the query is denoted by v_b.

The set of inputs that lead the computation of a decision tree to a certain vertex is a subcube. We will use the same symbol (e. g., v) to refer to a vertex as well as the subcube associated with it.

The execution of a decision tree terminates at some leaf. If the tree computes some S-relation $h \subseteq \{0,1\}^k \times S$, the leaves are labelled by elements of S, and the tree outputs the label of the leaf at which it terminates. We will also consider decision tree with unlabelled leaves (see Section 4).

3 Conflict complexity

In this section, we define the conflict complexity and max-conflict complexity of a partial Boolean function g on m bits. For this, we will need to introduce some notation related to a deterministic decision tree T. For a node $v \in T$, let $\pi(v) = \perp$ if v is the root and $\pi(v)$ be the parent of v otherwise. Let $q(v)$ be the index that is queried at v in T, and let $d_T(v)$ be the number of vertices on the unique path in T from the root to v (i. e., the depth of v). The depth of the root is 1.

Now fix a partial function $g \subseteq \{0,1\}^m \times \{0,1\}$ and probability distributions μ_0, μ_1 over $g^{-1}(0), g^{-1}(1)$, respectively. Let T be a tree that computes g. For a node $v \in T$ let $p_0(v) = \Pr_{\mu_0}[x_{q(v)} = 0 | x \text{ at } v]$ and $p_1(v) = \Pr_{\mu_1}[x_{q(v)} = 0 | x \text{ at } v]$, and

$$R(v) = \begin{cases} 1 & \text{if } v \text{ is the root,} \\ R(\pi(v)) \cdot \min\{\Pr_{\mu_0}[x \in v | x \in \pi(v)], \Pr_{\mu_1}[x \in v | x \in \pi(v)]\} & \\ & \text{otherwise} \ . \end{cases}$$

Also define

$$\Delta(v) = |p_0(v) - p_1(v)| \ .$$

To gather intuition about these quantities, imagine a random walk on T that begins at the root. At a node v, this walk moves to the left child with probability $\min\{p_0(v), p_1(v)\}$, and it moves to the right child with probability $1 - \max\{p_0(v), p_1(v)\}$. With the remaining probability, $\Delta(v)$, it terminates at v. $R(v)$ is the probability that the walk reaches v. The walk always terminates before it reaches a leaf of T. To see why, note that if the walk reaches a leaf before terminating, then there are two inputs $x \in g^{-1}(0), y \in g^{-1}(1)$ both of which lead the computation of T to the same leaf, which contradicts the assumption that T computes g. Hence for any tree T computing g we have $\sum_{v \in T} \Delta(v) R(v) = 1$. In particular, this means that $\sum_{v \in T} d_T(v) \Delta(v) R(v)$—the expected number of steps the walk takes before it terminates—is always at most the depth of the tree T.

Definition 3.1 (Conflict complexity and max-conflict complexity)**.** Let g be a partial function. For distributions μ_0, μ_1 with $\mathrm{supp}(\mu_b) \subseteq g^{-1}(b)$ for $b \in \{0,1\}$, and a deterministic decision tree T computing g, define

$$\chi(T, (\mu_0, \mu_1)) = \sum_{v \in T} d_T(v) \Delta(v) R(v) \ .$$

The conflict complexity of g is

$$\chi(g) = \max_{\mu_0,\mu_1} \min_T \chi(T, (\mu_0, \mu_1)) \ ,$$

where the maximum is over all pairs of distributions (μ_0, μ_1), where μ_0 and μ_1 are supported on $g^{-1}(0)$ and $g^{-1}(1)$, respectively, and the minimum is taken over all deterministic trees T computing g. For Q a distribution with finite support over pairs of distributions satisfying $\mathrm{supp}_b(Q) \subseteq g^{-1}(b)$ for $b \in \{0,1\}$, and T a deterministic tree computing g, let $\chi(T, Q) = \mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\chi(T, (\mu_0, \mu_1))]$. Finally, the max-conflict complexity of g is

$$\bar{\chi}(g) = \sup_Q \min_T \chi(T, Q) \ ,$$

where the supremum is taken over Q with finite support such that $\mathrm{supp}_b(Q) \subseteq g^{-1}(b)$ for $b \in \{0,1\}$, and the minimum is taken over deterministic trees T computing g.

We can extend the definition of conflict complexity and max-conflict complexity to more general query processes that do not necessarily compute a function. We first need the notion of FULL.

Definition 3.2. For a deterministic tree T and pair of distributions (μ_0, μ_1) with disjoint support, we say that $(T, (\mu_0, \mu_1))$ is FULL if $\sum_{v\in T} \Delta(v)R(v) = 1$, i. e., if the random walk described above terminates with probability 1. We say that (T, Q) is FULL if $(T, (\mu_0, \mu_1))$ is FULL for each $(\mu_0, \mu_1) \in \mathrm{supp}(Q)$.

Definition 3.3. For a deterministic tree T and pair of distributions (μ_0, μ_1) such that $(T, (\mu_0, \mu_1))$ is FULL, define $\chi(T, (\mu_0, \mu_1)) = \sum_{v\in T} d_T(v)\Delta(v)R(v)$. For a distribution Q such that (T, Q) is FULL, define $\chi(T, Q) = \mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\chi(T, (\mu_0, \mu_1))]$.

3.1 Comparison with other query measures

The definition of conflict complexity appeared for the first time in [16], which is one of the two papers the current paper is a merger of (the other paper being [9]). Li [13] analyzed this definition and showed that the conflict complexity of a total Boolean function g is at least the block sensitivity of g. Here we show that the max-conflict complexity of a (partial) function g is at least as large as the sabotage complexity of g. For a total Boolean function g, Ben-David and Kothari [7] show that the sabotage complexity of g is at least as large as the fractional block sensitivity of g [1, 17, 11], which in turn is at least as large as the block sensitivity. They also show examples where the sabotage complexity is much larger than the partition bound, quantum query complexity and approximate polynomial degree, thus the same holds for max-conflict complexity as well.

We first need the following simple claim. Let δ_x be the probability distribution that puts weight 1 on the string x.

Claim 3.4. *Let T be a deterministic tree computing the partial function g and let x, y be such that $g(x) = 0, g(y) = 1$. Then*

$$\chi(T, (\delta_x, \delta_y)) = \mathrm{sep}_T(x, y) \ .$$

Proof. Let v_1 be the root of T, and $v_1, v_2, \ldots, v_t$ be the longest sequence of vertices in T that are visited both by x and y, i. e., $x_{q(v_t)} \neq y_{q(v_t)}$. Since T computes g, v_t is not a leaf. For each $i = 1, \ldots, t-1$ we see that $\Delta(v_i) = 0$, while $\Delta(v_t) = 1$. Also $R(v_i) = 1$ for each $i = 1, \ldots, t$, while $R(v) = 0$ for any other vertex. Thus $\sum_{v \in T} d(v)\Delta(v)R(v) = t = \mathrm{sep}_T(x, y)$. □

Theorem 3.5. *Let $g \subseteq \{0,1\}^m \times \{0,1\}$ be a partial Boolean function. Then $\bar{\chi}(g) \geq \mathbb{RS}(g)$.*

Proof. By Theorem B.1 (Appendix B),

$$\mathbb{RS}(g) = \max_p \min_T \mathbb{E}_{(x,y)\sim p}[\mathrm{sep}_T(x, y)] \ .$$

By definition of max-conflict complexity we have

$$\bar{\chi}(g) = \sup_Q \min_T \mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\chi(T, (\mu_0, \mu_1))] \ .$$

The distribution p in sabotage complexity is a special case of Q where all the pairs of distributions in the support are singleton distributions. Note that p has finite support. The theorem now follows from Claim 3.4. □

4 Query process

We now come to the most important definition of the paper, that of the query process $\mathcal{P}(\mathcal{B}, Q)$. Let $t > 0$ be any integer and $\mathcal{B}$ be any deterministic query algorithm that runs on inputs in $(\{0,1\}^m)^t$. Let $x = (x_i^{(j)})_{\substack{i=1,\ldots,t \\ j=1,\ldots,m}}$ be a generic input to $\mathcal{B}$, and let x_i stand for $(x_i^{(j)})_{j=1,\ldots,m}$. For a vertex v of $\mathcal{B}$, let $v^{(i)}$ denote the subcube in v projected to x_i, i. e., $v = v^{(1)} \times \ldots \times v^{(t)}$. Recall from Section 2 that v_b stands for the child of v corresponding to the query outcome being b, for $b \in \{0,1\}$.

The query process $\mathcal{P}(\mathcal{B}, Q)$ runs on an input $z \in \{0,1\}^t$ and uses the BITSAMPLER (Algorithm 1) routine to simulate the queries of $\mathcal{B}$ to x when it can. This process is the heart of how we will transform an algorithm for $f \circ g^n$ into a query efficient algorithm for f.

Definition 4.1 (Query process $\mathcal{P}(\mathcal{B}, Q)$)**.** Let $\mathcal{B}$ be a decision tree that runs on inputs from $(\{0,1\}^m)^t$. Let Q be a consistent probability distribution with finite support over pairs of distributions (μ_0, μ_1). The query process $\mathcal{P}(\mathcal{B}, Q)$ is run on an input $z \in \{0,1\}^t$ and is defined by Algorithm 2.

A few comments about Definition 4.1. First, we think of $\mathcal{B}$ and $\mathcal{P}$ as query procedures that query input variables and terminate. In particular, they do not have to produce outputs, i. e., their leaves do not have to be labeled. Second, note that in Algorithm 2 the segment from line 9 to line 19 corresponds to the BITSAMPLER procedure in Algorithm 1. Queries to the input bits z_i are made in line 15, which corresponds to step 9 of BITSAMPLER. Finally, the variables $\mathbb{N}_i$ are not directly used by the algorithm, but are used in its analysis.

Algorithm 2: $\mathcal{P}(\mathcal{B}, \mathcal{Q})$

Input: $z = (z_1, \ldots, z_t) \in \{0,1\}^t$.

for $1 \le k \le t$ **do**
  $\mathrm{QUERY}_k \leftarrow 0$. `// Indicates if` z_k `is queried.`
  $\mathbb{N}_k \leftarrow 0$. `// Counts references to` x_k `till` z_k `is queried.`
  Sample $(\mu_0^{(k)}, \mu_1^{(k)})$ from $\mathcal{Q}$.
$v \leftarrow$ Root of $\mathcal{B}$ `// Corresponds to` $(\{0,1\}^m)^t$.
while *v is not a leaf of $\mathcal{B}$* **do**
  Let $q(v) = (i, j)$, the jth coordinate of x_i
  if $QUERY_i = 0$ **then**
    Sample a fresh real number $r \sim [0,1]$ uniformly at random.
    if $r < \min_b \Pr_{x_i \sim \mu_b^{(i)}}[x_i^{(j)} = 0 \mid x_i \in v^{(i)}]$ **then**
      $v \leftarrow v_0$.
    else if $r > \max_b \Pr_{x_i \sim \mu_b^{(i)}}[x_i^{(j)} = 0 \mid x_i \in v^{(i)}]$ **then**
      $v \leftarrow v_1$.
    else
      Query z_i. $\mathrm{QUERY}_i \leftarrow 1$.
      if $r \le \Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in v^{(i)}]$ **then**
        $v \leftarrow v_0$.
      else
        $v \leftarrow v_1$.
    $\mathbb{N}_i \leftarrow \mathbb{N}_i + 1$.
  else
    $$b \leftarrow \begin{cases} 1 & \text{with probability } \Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 1 \mid x_i \in v^{(i)}] \\ 0 & \text{with probability } \Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in v^{(i)}] \end{cases}$$
    $v \leftarrow v_b$

We now present an important structural result about $\mathcal{P}(\mathcal{B}, \mathcal{Q})$. In particular, this formally proves that the procedure BITSAMPLER given in Algorithm 1 samples the bits from the right distribution. The theorem should be intuitively clear, but we present a formal proof for completeness. Recall the definition of $\gamma_z(\mathcal{Q})$ from Section 1.3.

Theorem 4.2 (Simulation Theorem)**.** *Let $\mathcal{B}$ be a deterministic decision tree running on inputs from $(\{0,1\}^m)^t$, and let v be a vertex in $\mathcal{B}$. Let $A_z(v, \mathcal{Q})$ be the event that $\mathcal{P}(\mathcal{B}, \mathcal{Q})$, when run on z, reaches node v. Let $B_z(v, \mathcal{Q})$ be the event that for a random input x sampled from $\gamma_z(\mathcal{Q})$, the computation of $\mathcal{B}$*

reaches v. Then for every $z \in \{0,1\}^t$ and each vertex v of $\mathcal{B}$,

$$\Pr[A_z(v, \mathcal{Q})] = \Pr[B_z(v, \mathcal{Q})] \ .$$

Proof. To save writing, we fix $z \in \{0,1\}^t$ and $\mathcal{Q}$ and let $A(v) := A_z(v, \mathcal{Q})$ be the event that $\mathcal{P}(\mathcal{B}, Q)$ reaches node v on input z, and $B(v) := B_z(v, \mathcal{Q})$ be the event that $\mathcal{B}$ reaches node v under the distribution $\gamma_z(\mathcal{Q})$. Additionally, we write $\overline{(\mu_0, \mu_1)} = ((\mu_0^{(1)}, \mu_1^{(1)}), \ldots, (\mu_0^{(t)}, \mu_1^{(t)}))$ for a t-tuple of pairs of distributions. In the following when we write $\mathbb{E}_{\overline{(\mu_0,\mu_1)} \sim \mathcal{Q}^t}$ this expectation is taken with respect to drawing each $(\mu_0^{(i)}, \mu_1^{(i)})$ independently from $\mathcal{Q}$.

Now notice that $\Pr[A(v)] = \mathbb{E}_{\overline{(\mu_0,\mu_1)} \sim \mathcal{Q}^t} \Pr[A(v) \mid \overline{(\mu_0, \mu_1)}]$ and $\Pr[B(v)] = \mathbb{E}_{\overline{(\mu_0,\mu_1)} \sim \mathcal{Q}^t}$ $\Pr[B(v) \mid \overline{(\mu_0, \mu_1)}]$. We prove by induction on $d(v)$, the depth of a node v, that

$$\Pr[A(v) \mid \overline{(\mu_0, \mu_1)}] = \Pr[B(v) \mid \overline{(\mu_0, \mu_1)}] \tag{4.1}$$

for any $\overline{(\mu_0, \mu_1)}$. This will give the claim.

Towards the aim of showing (4.1), fix an arbitrary $\overline{(\mu_0, \mu_1)}$.

Base case: $d(v) = 1$, i. e., v is the root of $\mathcal{B}$. Thus $\Pr[A(v) \mid \overline{(\mu_0, \mu_1)}] = \Pr[B(v) \mid \overline{(\mu_0, \mu_1)}] = 1$.

Inductive step: Assume that $d(v) \geq 2$, and that the statement is true for all vertices of depth at most $d(v) - 1$. Since $d(v) \geq 2$, v is not the root of $\mathcal{B}$. Let $u = u^{(1)} \times \ldots \times u^{(t)}$ be the parent of v, and say variable $x_i^{(j)}$ is queried at u. Without loss of generality we assume that $v = u_0$. We split the proof into the following two cases.

- **Case 1:** $\Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}] \leq \Pr_{x_i \sim \mu_{\overline{z}_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]$.

  Conditioned on $A(u)$, $\overline{(\mu_0, \mu_1)}$ and $\mathrm{QUERY}_i = 1$, the probability that $\mathcal{P}$ reaches v is $\Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]$. Also, conditioned on $A(u)$, $\overline{(\mu_0, \mu_1)}$ and $\mathrm{QUERY}_i = 0$ the probability that $\mathcal{P}$ reaches v is exactly equal to the probability that the real number r sampled at u lies in $[0, \Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]]$, which is equal to $\Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]$. Thus,

$$\begin{aligned} \Pr[A(v) \mid \overline{(\mu_0, \mu_1)}] &= \Pr[A(u) \mid \overline{(\mu_0, \mu_1)}] \cdot \Pr[A(v) \mid A(u), \overline{(\mu_0, \mu_1)}] \\ &= \Pr[A(u) \mid \overline{(\mu_0, \mu_1)}] \cdot \Pr_{x_i \sim \mu_{z_i}^{(i)}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]. \end{aligned} \tag{4.2}$$

  Now condition on $B(u)$ and $\overline{(\mu_0, \mu_1)}$. The probability that $\mathcal{B}$ reaches v is exactly equal to the probability that $x_i^{(j)} = 0$ when x is sampled according to the distribution

$\gamma_z(\overline{(\mu_0,\mu_1)})$ conditioned on the event that $x \in u$. Note that in the distribution $\gamma_z(\overline{(\mu_0,\mu_1)})$, the x_k's are independently distributed. Thus,

$$\begin{aligned}\Pr[B(v) \mid \overline{(\mu_0,\mu_1)}] &= \Pr[B(u) \mid \overline{(\mu_0,\mu_1)}] \cdot \Pr[B(v) \mid B(u), \overline{(\mu_0,\mu_1)}] \\ &= \Pr[B(u) \mid \overline{(\mu_0,\mu_1)}] \cdot \Pr_{x_i \sim \mu^i_{z_i}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]. \end{aligned} \tag{4.3}$$

By the inductive hypothesis, $\Pr[A(u) \mid \overline{(\mu_0,\mu_1)}] = \Pr[B(u) \mid \overline{(\mu_0,\mu_1)}]$. It follows from (4.2) and (4.3) that $\Pr[A(v) \mid \overline{(\mu_0,\mu_1)}] = \Pr[B(v) \mid \overline{(\mu_0,\mu_1)}]$.

- **Case 2:** $\Pr_{x_i \sim \mu^{(i)}_{z_i}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}] > \Pr_{x_i \sim \mu_{\overline{z}_i^{(i)}}}[x_i^{(j)} = 0 \mid x_i \in u^{(i)}]$. Let $v' = u_1$. By an argument similar to Case 1, we have that

$$\Pr[A(v') \mid \overline{(\mu_0,\mu_1)}] = \Pr[B(v')\overline{(\mu_0,\mu_1)}]. \tag{4.4}$$

Now,

$$\begin{aligned}\Pr[A(v) \mid \overline{(\mu_0,\mu_1)}] &= \Pr[A(u) \mid \overline{(\mu_0,\mu_1)}] - \Pr[A(v') \mid \overline{(\mu_0,\mu_1)}] \\ &= \Pr[B(u) \mid \overline{(\mu_0,\mu_1)}] - \Pr[A(v') \mid \overline{(\mu_0,\mu_1)}] \\ &\qquad \text{(By inductive hypothesis)} \\ &= \Pr[B(u) \mid \overline{(\mu_0,\mu_1)}] - \Pr[B(v') \mid \overline{(\mu_0,\mu_1)}] \\ &\qquad \text{(By (4.4))} \\ &= \Pr[B(v) \mid \overline{(\mu_0,\mu_1)}] \ . \end{aligned}$$

□

We will be interested in the number of queries $\mathcal{P}(\mathcal{B},\mathcal{Q})$ is able to simulate before making a query to z_i. To this end, let the random variable $\mathcal{N}_i(\mathcal{B},z,\mathcal{Q})$ stand for the value of the variable $\mathbb{N}_i$ in Algorithm 2 after the termination of $\mathcal{P}(\mathcal{B},\mathcal{Q})$ on input z. Note that $\mathcal{N}_i$ depends on the randomness in the choices of r (step 9) and also on the randomness in $\mathcal{Q}$ in the choice of distributions $(\mu_0^{(k)},\mu_1^{(k)})$ (step 4).

4.1 Relating $\mathcal{P}(\mathcal{B},\mathcal{Q})$ to max-conflict complexity

A key to our composition theorem will be relating the number of simulated queries made by $\mathcal{P}(\mathcal{B},\mathcal{Q})$ to max-conflict complexity, which we do in this section. Let $\mathcal{B}$ be a query algorithm taking inputs from $\{0,1\}^m$. In this case, $\mathcal{N}_1(\mathcal{B},1,\mathcal{Q}) = \mathcal{N}_1(\mathcal{B},0,\mathcal{Q})$. This is because the behavior of $\mathcal{P}(\mathcal{B},\mathcal{Q})$ on input 0 is exactly the same as the behavior on input 1 before a query to z is made, and after z is queried the value of $\mathbb{N}_i$ does not change.

Claim 4.3. *Let $\mathcal{B}$ be an algorithm taking inputs from $\{0,1\}^m$. Then $(\mathcal{B},\mathcal{Q})$ is FULL if and only if $\mathcal{P}(\mathcal{B},\mathcal{Q})$ queries z with probability 1. If $(\mathcal{B},\mathcal{Q})$ is FULL then*

$$\chi(\mathcal{B},\mathcal{Q}) = \mathbb{E}[\mathcal{N}_1(\mathcal{B},1,\mathcal{Q})]$$

Proof. Note that until z is queried, $\mathcal{P}(\mathcal{B},(\mu_0,\mu_1))$ exactly executes the random walk described in Section 3, and querying z in $\mathcal{P}(\mathcal{B},(\mu_0,\mu_1))$ corresponds to this random walk terminating. The first part of the claim then follows as $\mathcal{P}(\mathcal{B},\mathcal{Q})$ queries z with probability 1 if and only if $\mathcal{P}(\mathcal{B},(\mu_0,\mu_1))$ queries z with probability 1 for every $(\mu_0,\mu_1)\in\operatorname{supp}(\mathcal{Q})$.

Also, because $\mathcal{P}(\mathcal{B},(\mu_0,\mu_1))$ exactly executes the random walk described in Section 3, we see that $\chi(\mathcal{B},(\mu_0,\mu_1)) = \mathbb{E}[\mathcal{N}_1(\mathcal{B},1,(\mu_0,\mu_1))]$. The second part of the claim follows by taking the expectation of this equality over $(\mu_0,\mu_1)\sim\mathcal{Q}$. □

The correspondence of Claim 4.3 prompts us to define FULL in a more general setting.

Definition 4.4 (FULL)**.** Let $\mathcal{B}$ be a query algorithm taking inputs from $(\{0,1\}^m)^t$. The pair $(\mathcal{B},\mathcal{Q})$ is said to be FULL if for every $z\in\{0,1\}^t$ it holds that $\mathcal{P}(\mathcal{B},\mathcal{Q})$ queries z_i with probability 1, for every $i=1,\dots,t$.

5 The Composition Theorem

In this section we prove Theorem 1.5 (restated below).

Theorem 5.1. *For any S-relation $f\subseteq\{0,1\}^n\times S$ and any partial Boolean function $g\subseteq\{0,1\}^m\times\{0,1\}$,*

$$\mathbb{R}_{1/3}(f\circ g^n)\in\Omega\left(\mathbb{R}_{4/9}(f)\cdot\bar{\chi}(g)\right).$$

Our proof will make use of the following *direct sum theorem*.

Theorem 5.2. *Let $\mathcal{B}$ be a query algorithm acting on inputs from $(\{0,1\}^m)^t$. Let $\mathcal{Q}$ be a consistent distribution with finite support over pairs of distributions (μ_0,μ_1) on m-bit strings. If $(\mathcal{B},\mathcal{Q})$ is FULL then for any $z\in\{0,1\}^t$*

$$\sum_{i=1}^{t}\mathbb{E}[\mathcal{N}_i(\mathcal{B},z,\mathcal{Q})]\geq t\cdot\min_{C}\chi(C,\mathcal{Q})\ ,$$

where the minimum is taken over deterministic trees C acting on inputs from $\{0,1\}^m$ such that $(C,\mathcal{Q})$ is FULL.

Proof of Theorem 5.2. Towards a contradiction, assume that

$$\sum_{i=1}^{t}\mathbb{E}[\mathcal{N}_i(\mathcal{B},z,\mathcal{Q})]<t\cdot\min_{C}\mathbb{E}_{(\mu_0,\mu_1)\sim\mathcal{Q}}[\chi(C,(\mu_0,\mu_1))]\ . \tag{5.1}$$

By averaging, there exists a k such that $\mathbb{E}[\mathcal{N}_k(\mathcal{B},z,\mathcal{Q})]]<\min_C\mathbb{E}_{(\mu_0,\mu_1)\sim\mathcal{Q}}$ $[\chi(C,(\mu_0,\mu_1))]$. Let us focus on the expression on the left hand side. Recall that there are two kinds of randomness in this expectation, the choice of the random numbers r in $\mathcal{P}(\mathcal{B},\mathcal{Q})$ and the choice of $\overline{(\mu_0,\mu_1)}\sim\mathcal{Q}^t$. We separate out these two as follows:

$$\begin{aligned}\mathbb{E}[\mathcal{N}_k(\mathcal{B},z,\mathcal{Q})] &= \mathbb{E}_{\overline{(\mu_0,\mu_1)}\sim\mathcal{Q}^t}\mathbb{E}_r[\mathcal{N}_k(\mathcal{B},z,\overline{(\mu_0,\mu_1)}]\\ &= \mathbb{E}_r\mathbb{E}_{\overline{(\mu_0,\mu_1)}\sim\mathcal{Q}^t}[\mathcal{N}_k(\mathcal{B},z,\overline{(\mu_0,\mu_1)}]\\ &= \mathbb{E}_r\mathbb{E}_{\overline{(\mu_0,\mu_1)}^{-(k)}\sim\mathcal{Q}^{t-1}}\mathbb{E}_{(\mu_0^{(k)},\mu_1^{(k)})\sim\mathcal{Q}}[\mathcal{N}_k(\mathcal{B},z,\overline{(\mu_0,\mu_1)}]\ ,\end{aligned}$$

where $\overline{(\mu_0,\mu_1)}^{-(k)}$ is a $(t-1)$-tuple of pairs of distributions without the kth coordinate. This further means that there is a fixing of the randomness r and the $(t-1)$-tuple of pairs distributions $\overline{(\mu_0,\mu_1)}^{-(k)}$ such that $\mathbb{E}_{(\mu_0^{(k)},\mu_1^{(k)})\sim Q}[\mathcal{N}_k(\mathcal{B},z,\overline{(\mu_0,\mu_1)})] < \min_C \mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\chi(C,(\mu_0,\mu_1))]$. With such a fixed setting, however, $\mathcal{P}(\mathcal{B},Q)$ creates a query process equivalent to $\mathcal{P}(\mathcal{B}',\mathbb{Q})$ run on $z_i \in \{0,1\}$ for a deterministic query algorithm $\mathcal{B}'$ running on inputs from $\{0,1\}^m$ and such that $(\mathcal{B}',Q)$ is FULL. The distribution $\mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\mathcal{N}_1(\mathcal{B}',1,\mu_0,\mu_1)]$ is the same as that as $\mathbb{E}_{(\mu_0^{(k)},\mu_1^{(k)})\sim Q}[\mathcal{N}_k(\mathcal{B},z,(\overline{(\mu_0,\mu_1)}^{-(k)},(\mu_0^{(k)},\mu_1^{(k)}))]$ conditioned on the earlier fixing of $\overline{(\mu_0,\mu_1)}^{-(k)}$ and the randomness r. Thus $\mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\chi(\mathcal{B}',(\mu_0,\mu_1))] < \min_C \mathbb{E}_{(\mu_0,\mu_1)\sim Q}[\chi(C,(\mu_0,\mu_1))]$, a contradiction. □

Proof of Theorem 1.5. We shall prove that for each distribution η on the inputs to f, there is a randomized query algorithm $\mathcal{A}$ making at most $18\mathbb{R}_{1/3}(f\circ g^n)/\bar{\chi}(g)$ queries in the worst case, for which $\Pr_{z\in\eta}[(z,\mathcal{A}(z))\in f] \geq \frac{5}{9}$ holds. $\mathcal{A}$ can be made deterministic with the same complexity and accuracy guarantees by appropriately fixing its randomness. This will imply the theorem by the *minmax principle* (Fact 2.4). To this end let us fix a distribution η over $\{0,1\}^n$.

Let Q be a distribution with finite support which is consistent with g such that for any deterministic decision tree C computing g we have $\chi(C,Q) \geq \bar{\chi}(g)-\epsilon$, where ϵ is to be set later. We will use distributions η and Q to set up a distribution γ_η over the input space of $f\circ g^n$. This distribution is defined as follows:

1. Sample $z=(z_1,\ldots,z_n)$ from η.
2. Sample $(\mu_0^{(i)},\mu_1^{(i)})$ independently from Q for $i=1,\ldots,t$.
3. Sample x_i from $\mu_{z_i}^{(i)}$ for $i=1,\ldots,t$. Return $x=(x_1,\ldots,x_t)$.

Recall from Section 1.3 the observation that for each z,x sampled as above, for each $s\in\mathcal{S}$, $(z,s)\in f$ *if and only if* $(x,s)\in f\circ g^n$.

Assume that $\mathbb{R}_{1/3}(f\circ g^n)=c$. The minimax principle (Fact 2.4) implies that there is a deterministic query algorithm $\mathcal{A}'$ for inputs from $(\{0,1\}^m)^n$, that makes at most c queries in the worst case, such that $\Pr_{x\in\gamma_\eta}[(x,\mathcal{A}'(x))\in f\circ g^n] \geq \frac{2}{3}$. We will first use $\mathcal{A}'$ to construct a randomized algorithm T for f whose accuracy under the distribution η is as desired and which, for every input z, makes few queries in expectation. T is described in Algorithm 3.

Algorithm 3: T

Input: $z\in\{0,1\}^n$
1 Run $\mathcal{P}(\mathcal{A}',Q)$ on z.
2 Return the output of $\mathcal{A}'$.

First we bound the probability of error by T. By Theorem 4.2, we have that $\Pr[(z,T(z))\in f] = \Pr_{x\sim\gamma_z(Q)}[(x,\mathcal{A}'(x))\in f\circ g^n]$ for each $z\in\{0,1\}^n$. Thus, $\Pr_{z\sim\eta}[(z,T(z))\in f] = \Pr_{x\sim\gamma_\eta}[(x,\mathcal{A}'(x))\in f\circ g^n] \geq \frac{2}{3}$.

Next, we bound the expected number of queries made by T in the worst-case.

Claim 5.3. *The expected number of queries made by T on each input z is at most $\frac{2c}{\bar{\chi}(g)}$.*

Proof. Fix an input $z \in \{0,1\}^n$. For each leaf $\ell = \ell^{(1)} \times \ldots \times \ell^{(n)}$ of $\mathcal{A}'$ and for each $i = 1, \ldots, n$ define $\mathcal{E}'_{i,\ell}$ to be the event that the computation of $\mathcal{P}(\mathcal{A}', \mathcal{Q})$ finishes at ℓ with $\mathsf{QUERY_i} = 0$. For $i = 1, \ldots, n$ define $\mathcal{F}'_i$ to be the event that $\mathsf{QUERY_i}$ is set to 1 in $\mathcal{P}(\mathcal{A}', \mathcal{Q})$. Let $\mathcal{Q}^n$ stand for the (product) distribution of n pairs of probability distributions each independently sampled from $\mathcal{Q}$. For i, ℓ such that g is not constant on $\ell^{(i)}$, let $\mathcal{D}_{i,\ell}$ be the distribution given by the following sampling procedure:

1. Sample $(\mu_0^{(1)}, \mu_1^{(1)}), \ldots, (\mu_0^{(n)}, \mu_1^{(n)})$ from $\mathcal{Q}^n$ conditioned on $\mathcal{E}'_{i,\ell}$,
2. return $(\mu_0^{(i)} \mid \ell^{(i)}, \mu_1^{(i)} \mid \ell^{(i)})$.

Let $\mathcal{B}_{i,\ell}$ be an optimal tree for $\mathcal{D}_{i,\ell}$, i. e., $\chi(\mathcal{B}_{i,\ell}, \mathcal{D}_{i,\ell}) = \min_C \chi(C, \mathcal{D}_{i,\ell})$, where the minimization is over all algorithms C that output 0 on $\mathrm{supp}_0(\mathcal{D}_{i,\ell})$ and output 1 on $\mathrm{supp}_1(\mathcal{D}_{i,\ell})$. Now, consider the query algorithm H defined in Algorithm 4. Note that $(H, \mathcal{Q})$ is FULL. Now consider a

Algorithm 4: H

Input: $x \in (\{0,1\}^m)^n$
Run $\mathcal{A}'$ on x.
Let $\mathcal{A}'$ terminate at leaf $\ell = \ell^{(1)} \times \ldots \times \ell^{(n)}$.
for $1 \le i \le n$ **do**
  if *g is not constant on $\ell^{(i)}$* **then**
    Run $\mathcal{B}_{i,\ell}$ on x_i.

run of the query process $\mathcal{P}(H, \mathcal{Q})$ on input z. Theorem 5.2 implies that $\sum_{i=1}^n \mathbb{E}[\mathcal{N}_i(H, z, \mathcal{Q})] \ge n \cdot \min_C \chi(C, \mathcal{Q}) = n \cdot (\bar{\chi}(g) - \epsilon)$, by the choice of $\mathcal{Q}$.

Let $\mathcal{F}_i$ to be the event that $\mathsf{QUERY_i}$ is set to 1 in $\mathcal{P}(H, \mathcal{Q})$ when it reaches a leaf of $\mathcal{A}'$, and for each leaf ℓ of $\mathcal{A}'$ let $\mathcal{E}_{i,\ell}$ be the event that $\mathcal{P}(H, \mathcal{Q})$ reaches ℓ and $\mathsf{QUERY_i} = 0$ when it does. Observe that for each $i = 1, \ldots, n$, the events $\{\mathcal{F}_i, (\mathcal{E}_{i,\ell})_\ell\}$ are mutually exclusive and exhaustive.

We have that

$$\begin{aligned} n \cdot (\bar{\chi}(g) - \epsilon) &\le \sum_{i=1}^n \mathbb{E}[\mathcal{N}_i(H, z, \mathcal{Q})] \\ &= \sum_{i=1}^n \sum_\ell \Pr[\mathcal{E}_{i,\ell}] \cdot \mathbb{E}[\mathcal{N}_i(H, z, \mathcal{Q}) \mid \mathcal{E}_{i,\ell}] + \sum_{i=1}^n \Pr[\mathcal{F}_i] \cdot \mathbb{E}[\mathcal{N}_i(H, z, \mathcal{Q}) \mid \mathcal{F}_i] \qquad (5.2) \end{aligned}$$

Let $d_i(\ell)$ be the number of queries into x_i made in the unique path from the root of $\mathcal{A}'$ to ℓ. Now, condition on the n pairs of distributions $(\mu_0^{(j)}, \mu_1^{(j)})_{j=1,\ldots,n}$ that are used in $\mathcal{P}(H, \mathcal{Q})$. We have that,

$$\mathbb{E}[\mathcal{N}_i(H, z, \mathcal{Q}) \mid \mathcal{E}_{i,\ell}, (\mu_0^{(j)}, \mu_1^{(j)})_{j=1,\ldots,n}] = d_i(\ell^{(i)}) + \mathbb{E}[\mathcal{N}_1(\mathcal{B}_{i,\ell}, z_i, (\mu_0^{(i)} \mid \ell^{(i)}, \mu_1^{(i)} \mid \ell^{(i)}))]. \qquad (5.3)$$

Averaging over $(\mu_0^{(j)}, \mu_1^{(j)})_{j=1,\dots,n}$ we have from (5.3) that

$$\begin{aligned}\mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{E}_{i,\ell}] &= d_i(\ell^{(i)}) + \mathbb{E}[\mathcal{N}_1(\mathcal{B}_{i,\ell}, z_i, \mathcal{D}_{i,\ell})] \\ &= d_i(\ell^{(i)}) + \min_{C} \chi(C, \mathcal{D}_{i,\ell}) \quad \text{(By the choice of } \mathcal{B}_{i,\ell}). \\ &\le d_i(\ell^{(i)}) + \bar{\chi}(g). \end{aligned} \tag{5.4}$$

Observing that $\sum_\ell \Pr[\mathcal{E}_{i,\ell}] = 1 - \Pr[\mathcal{F}_i]$, we have from (5.2) and (5.4) that

$$\begin{aligned} n \cdot (\bar{\chi}(g) - \epsilon) &\le \sum_{i=1}^{n} \sum_{\ell} \Pr[\mathcal{E}_{i,\ell}] \cdot (d_i(\ell^{(i)}) + \bar{\chi}(g)) + \sum_{i=1}^{n} \Pr[\mathcal{F}_i] \cdot \mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{F}_i] \\ &= \sum_{i=1}^{n} (1 - \Pr[\mathcal{F}_i]) \cdot \bar{\chi}(g) \\ &\qquad + \sum_{i=1}^{n} \left(\sum_{\ell} (\Pr[\mathcal{E}_{i,\ell}] \cdot d_i(\ell^{(i)}) + \Pr[\mathcal{F}_i] \cdot \mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{F}_i]) \right) \\ \Rightarrow \sum_{i=1}^{n} \Pr[\mathcal{F}_i] &\le \frac{1}{\bar{\chi}(g)} \cdot \sum_{i=1}^{n} \left(\sum_{\ell} (\Pr[\mathcal{E}_{i,\ell}] \cdot d_i(\ell^{(i)}) + \Pr[\mathcal{F}_i] \cdot \mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{F}_i]) \right) \\ &\qquad + \frac{n\epsilon}{\bar{\chi}(g)}. \end{aligned} \tag{5.5}$$

We will show that $\sum_{i=1}^{n} \left(\sum_\ell (\Pr[\mathcal{E}_{i,\ell}] \cdot d_i(\ell^{(i)}) + \Pr[\mathcal{F}_i] \cdot \mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{F}_i]) \right) \le c$. We set $\epsilon \le \frac{c}{n}$. Since $\sum_{i=1}^{n} \Pr[\mathcal{F}_i]$ is exactly the expected number of queries made by T, the claim will follow from (5.5).

Consider a run of $\mathcal{P}(H,Q)$ on input z, and let c_i be a random variable denoting the number of times step 7 of Algorithm 2 (with $\mathcal{B} = H$) is a query into x_i before a leaf of $\mathcal{A}'$ is reached, for $i = 1, \dots, n$. Thus $\sum_{i=1}^{n} \mathbb{E}[c_i] \le c$. Further, for each i, ℓ we have $d_i(\ell^{(i)}) = \mathbb{E}[c_i \mid \mathcal{E}_{i,\ell}]$ and $\mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{F}_i] = \mathbb{E}[\mathcal{N}_i(\mathcal{A}',z,Q) \mid \mathcal{F}_i] \le \mathbb{E}[c_i \mid \mathcal{F}_i]$. Thus,

$$\begin{aligned} &\sum_{i=1}^{n} \left(\sum_{\ell} (\Pr[\mathcal{E}_{i,\ell}] \cdot d_i(\ell^{(i)}) + \Pr[\mathcal{F}_i] \cdot \mathbb{E}[\mathcal{N}_i(H,z,Q) \mid \mathcal{F}_i]) \right) \\ &\le \sum_{i=1}^{n} \left(\sum_{\ell} (\Pr[\mathcal{E}_{i,\ell}] \cdot \mathbb{E}[c_i \mid \mathcal{E}_{i,\ell}]) + \Pr[\mathcal{F}_i] \cdot \mathbb{E}[c_i \mid \mathcal{F}_i] \right) \\ &= \sum_{i=1}^{n} \mathbb{E}[c_i] \le c. \end{aligned}$$

□

Now we finish the proof of Theorem 1.5 by constructing the query algorithm $\mathcal{A}$. Let $\mathcal{A}''$ be the algorithm obtained by terminating T after $18c/\bar{\chi}(g)$ queries. By Markov's inequality, for each

z, the probability that T makes more than $18c/\bar{\chi}(g)$ queries is at most $1/9$. Thus $\mathcal{A}''$ computes f with probability at least $2/3 - 1/9 = 5/9$ on a random input from η. Finally, $\mathcal{A}$ is obtained by fixing the randomness of $\mathcal{A}''$ appropriately so that the above probabilistic guarantee holds. □

6 Conflict complexity and randomized query complexity

In this section, we will prove Theorem 1.7 (restated below). Our proof relates the conflict complexity to the expected amount of information that is learned about the function value through each query via Pinsker's Inequality. At a high level, our proof is reminiscent of the result of [4] on compressing communication protocols in that both look at a random sampling process to navigate a tree, and relate the probability of this process needing to query or communicate at a node to the amount of information that is learned at the node.

Theorem 6.1 (Restatement of Theorem 1.7)**.** *For any partial Boolean function $g \subseteq \{0,1\}^m \times \{0,1\}$,*

$$\chi(g) \in \Omega\left(\sqrt{\mathbb{R}_{1/3}(g)}\right).$$

Proof. We will show that there exists a constant $\epsilon < 1/2$ such that for each input distribution μ, $\mathbb{D}^{\mu}_{\epsilon}(g) \leq 10\chi(g)^2$. Theorem 1.7 will follow from the *minimax principle* (Fact 2.4), and the observation that the error can be brought down to $1/3$ by constantly many independent repetitions followed by a selection of the majority of the answers. It is enough to consider distributions μ supported on valid inputs of g. To this end, fix a distribution μ supported on $g^{-1}(0) \cup g^{-1}(1)$. Define $\mu_0 := \mu \mid g^{-1}(0), \mu_1 := \mu \mid g^{-1}(1)$.

Let $\chi(g) = d$. Let $\mathcal{B}$ be a deterministic query algorithm for inputs in $\{0,1\}^m$ such that $(\mathcal{B}, \mu_0, \mu_1)$ is FULL and $\chi(\mu_0, \mu_1) = \chi(\mathcal{B}, \mu_0, \mu_1)$. We call such a decision tree an *optimal* decision tree for μ_0, μ_1. Thus in $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$, $\mathbb{E}[\mathcal{N}_1] = \chi(\mu_0, \mu_1) \leq d$. Recall from Section 4 that the leaves of $\mathcal{B}$ can be labelled by bits such that $\mathcal{B}$ computes g on the supports of μ_0 and μ_1. We assume $\mathcal{B}$'s leaves to be labelled as such.

Consider the following query algorithm $\mathcal{B}'$: Start simulating $\mathcal{B}$. Terminate the simulation if one of the following events occurs. The output in each case is specified below.

1. If $10d^2$ queries have been made and $v_{10d^2+1} \neq \perp$, terminate and output $\arg\max_b \Pr_{x\sim\mu}[g(x) = b \mid x \in v_{10d^2+1}]$.

2. If $\mathcal{B}$ terminates, terminate and output what $\mathcal{B}$ outputs.

By construction, $\mathcal{B}'$ makes at most $10d^2$ queries in the worst case. The following claim bounds the error of $\mathcal{B}'$, and completes the proof of Theorem 1.7.

Claim 6.2. *There exists a constant $\epsilon < 1/2$ such that $\Pr_{x\sim\mu}[\mathcal{B}'(x) \neq g(x)] \leq \epsilon$. Furthermore, the constant ϵ is independent of μ.*

Proof of Claim 6.2. Let v_k be the random vertex at which the $\mathcal{B}$ makes its k-th query when it is run on x; If $\mathcal{B}$ terminates before making k queries, define $v_k := \perp$. Let $\mathcal{E}$ denote the

event that in at most $10d^2$ queries, the computation of $\mathcal{B}$ does not reach a vertex v such that $\Pr_{x\sim\mu}[g(x)=0 \mid x\in v]\cdot \Pr_{x\sim\mu}[g(x)=1 \mid x\in v] \le \frac{1}{9}$. Since $\mathcal{B}$ computes g on the supports of μ_0 and μ_1, therefore if $\mathcal{E}$ happens then the computation of $\mathcal{B}$ does not reach a leaf within $10d^2$ queries. We split the proof into the following two cases.

Case 1: $\Pr[\mathcal{E}] < \frac{3}{4}$.

Condition on the event that the computation reaches a vertex v of $\mathcal{B}$ for which $\Pr_{x\sim\mu}[g(x)=0 \mid x\in v]\cdot \Pr_{x\sim\mu}[g(x)=1 \mid x\in v] \le \frac{1}{9}$ holds. In this case, one of $\Pr_{x\sim\mu}[g(x)=0 \mid x\in v]$ and $\Pr_{x\sim\mu}[g(x)=1 \mid x\in v]$ is at most $1/3$. Hence, $|\Pr_{x\sim\mu}[g(x)=0 \mid x\in v] - \Pr_{x\sim\mu}[g(x)=1 \mid x\in v]| \ge 1/3$. Let w be the random leaf of the subtree of $\mathcal{B}'$ rooted at v at which the computation ends. The probability that $\mathcal{B}'$ errs is at most

$$\begin{aligned}
&\mathbb{E}_w\left[\frac{1}{2}-\frac{1}{2}\left|\Pr_{x\sim\mu}[g(x)=0 \mid x\in w]-\Pr_{x\sim\mu}[g(x)=1 \mid x\in w]\right|\right]\\
&\le \frac{1}{2}-\frac{1}{2}\left|\mathbb{E}_w\left[\Pr_{x\sim\mu}[g(x)=0 \mid x\in w]\right]-\mathbb{E}_w\left[\Pr_{x\sim\mu}[g(x)=1 \mid x\in w]\right]\right|\\
&\qquad\qquad\text{(By Jensen's inequality and linearity of expectation)}\\
&= \frac{1}{2}-\frac{1}{2}\left|\Pr_{x\sim\mu}[g(x)=0 \mid x\in v]-\Pr_{x\sim\mu}[g(x)=1 \mid x\in v]\right| \le \frac{1}{3}.
\end{aligned}$$

Thus we have shown that conditioned on $\overline{\mathcal{E}}$ the probability that $\mathcal{B}'$ errs is at most $\frac{1}{3}$. Since $\mathcal{B}'$ errs with probability at most $1/2$ when $\mathcal{E}$ happens due to the decision in step 1, therefore the probability that $\mathcal{B}'$ errs is at most $\frac{1}{4}\cdot\frac{1}{3}+\frac{3}{4}\cdot\frac{1}{2}=\frac{11}{24}<\frac{1}{2}$.

Case 2: $\Pr[\mathcal{E}] \ge \frac{3}{4}$.

Let $a_j := (i_j, x_{i_j})$ be the tuple formed by the index and value of the random input variable queried at the j-th step by $\mathcal{B}'$; if $\mathcal{B}'$ terminates before making j queries (i. e., $v_j=\perp$) or v_j is a leaf of $\mathcal{B}$, then define $i_j, x_{i_j} := \perp$. Note that the sequence $(a_1,\dots,a_{10d^2})$ uniquely specifies a leaf of $\mathcal{B}'$, and vice versa. Let $\mathsf{I}(\cdot,\cdot)$ denote the *mutual information*. (See Appendix C for the definitions and results from information theory used in this paper.) We prove the following claim in Section 6.1.

Claim 6.3. *If* $\Pr[\mathcal{E}] \ge \frac{3}{4}$, *then* $\mathsf{I}(a_1,\dots,a_{10d^2} : g(x)) \ge \frac{1}{40}$.

Thus if $\Pr[\mathcal{E}] \ge \frac{3}{4}$, Claim 6.3 implies that

$$\mathsf{H}(g(x) \mid a_1,\dots a_{10d^2}) \le 1-\frac{1}{40}=\frac{39}{40}. \tag{6.1}$$

Let $\mathcal{L}$ be the set of leaves ℓ of $\mathcal{B}'$ such that $\mathsf{H}(g(x) \mid x\in\ell) \le \frac{79}{80}$. For each $\ell\in\mathcal{L}$, $\min_b \Pr_{x\sim\mu}[g(x)=b \mid x\in\ell] \le \frac{9}{20}$. Conditioned on $(a_1,\dots,a_{10d^2})\in\mathcal{L}$, the probability that $\mathcal{B}'$ errs is at most $\frac{9}{20}$. By *Markov's inequality* and (6.1), it follows that $\Pr[(a_1,\dots,a_{10d^2})\in\mathcal{L}] \ge \frac{1}{79}$. Thus $\mathcal{B}'$ errs with probability at most $\frac{1}{79}\cdot\frac{9}{20}+\frac{78}{79}\cdot\frac{1}{2}<\frac{1}{2}$.

□

This completes the proof of Theorem 1.7. □

6.1 Proof of Claim 6.3

Let v be a vertex in $\mathcal{B}$. Define $\Delta(v)$ as follows.[6]

$$\Delta(v) := \begin{cases} |\Pr_{x\sim\mu_0}[x_i = 0 \mid x \in v] - \Pr_{x\sim\mu_1}[x_i = 0 \mid x \in v]| \\ \qquad\qquad \text{if } v \neq \perp \text{ and } \Pr_{x\sim\mu_b}[x \in v] > 0 \text{ for } b \in \{0,1\}, \\ 1 \qquad\qquad \text{otherwise.} \end{cases}$$

The following claim shows that if $\Delta(v)$ is large, the query outcome of v contains significant information about $g(x)$.

Claim 6.4. *Let v be a vertex in $\mathcal{B}$. Let variable x_i be queried at v. Then,*

$$\mathsf{I}(g(x) : x_i \mid x \in v) \geq 8\left(\Pr_{x\sim\mu}[g(x) = 0 \mid x \in v] \cdot \Pr_{x\sim\mu}[g(x) = 1 \mid x \in v] \cdot \Delta(v)\right)^2.$$

Proof. Define $b := g(x)$. Condition on the event $x \in v$. Recall from Appendix C that $(b \otimes x_i)$ is the distribution over pairs of bits, where the the first and the second bit are distributed independently according to the distributions of b and x_i, respectively. Fact C.7 implies that $\mathsf{I}(b : x_i) = \mathrm{D}((b, x_i)||(b \otimes x_i))$[7]. Now, *Pinsker's inequality* (Theorem C.9) implies that

$$\mathrm{D}((b, x_i)||(b \otimes x_i)) \geq \frac{1}{2}||(b, x_i) - (b \otimes x_i)||_1^2. \tag{6.2}$$

Next, we bound $||(b, x_i) - (b \otimes x_i)||_1$. To this end, we fix bits $z_1, z_2 \in \{0,1\}$, and bound $|\Pr[(b, x_i) = (z_1, z_2)] - \Pr[(b \otimes x_i) = (z_1, z_2)]|$. We have that,

$$\Pr[(b, x_i) = (z_1, z_2)] = \Pr[b = z_1]\Pr[x_i = z_2 \mid b = z_1]. \tag{6.3}$$

Now,

$$\begin{aligned} \Pr[(b \otimes x_i) = (z_1, z_2)] &= \Pr[b = z_1]\Pr[x_i = z_2] \\ &= \Pr[b = z_1](\Pr[b = z_1]\Pr[x_i = z_2 \mid b = z_1] + \\ &\qquad \Pr[b = \overline{z_1}]\Pr[x_i = z_2 \mid b = \overline{z_1}]). \end{aligned} \tag{6.4}$$

Taking the absolute difference of (6.4) and (6.3) we have that,

$$\begin{aligned} &|\Pr[(b, x_i) = (z_1, z_2)] - \Pr[(b \otimes x_i) = (z_1, z_2)]| \\ &= \Pr[b = z_1] \cdot \Pr[b = \overline{z_1}] \cdot \Delta(v) = \Pr[b = 0] \cdot \Pr[b = 1] \cdot \Delta(v) \end{aligned} \tag{6.5}$$

The Claim follows by adding (6.5) over z_1, z_2 and using (6.2). □

[6] Recall that we mentioned $\Delta(v)$ in Section 1.3.

[7] See Appendix C for definition of Kullback-Leibler divergence and L_1-distance.

Let $\mathcal{B}$ be run on a random input x sampled from μ. The next claim proves a lower bound on the expected sum of $\Delta(v)$ for the random vertices v in the transcript of $\mathcal{B}$. Recall from the proof of Claim 6.2 that v_k is the random vertex at which the k-th query is made; If $\mathcal{B}$ terminates before making k queries, define $v_k := \perp$. Note that if $\mathcal{B}$ terminates before making t queries, $v_t = \perp$ and $\Delta(v_t) = 1$.

Claim 6.5. *Let c be any positive integer. Then,*

$$\sum_{k=1}^{10dc} \mathbb{E}[\Delta(v_k) \mid \mathcal{E}] \geq \frac{13c}{20}.$$

To prove Claim 6.5 we need the following claim.

Claim 6.6.

$$\sum_{k=1}^{10d} \mathbb{E}[\Delta(v_k) \mid \mathcal{E}] \geq \frac{13}{20}.$$

Proof of Claim 6.6. Let us sample vertices u_k of $\mathcal{B}$ as follows:

1. Set $z = \begin{cases} 1 & \text{with probability } \Pr_{x\sim\mu}[g(x) = 1], \\ 0 & \text{with probability } \Pr_{x\sim\mu}[g(x) = 0] \end{cases}$
2. Run $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$ on the 1-bit input z.
3. Let u_k be the vertex v of $\mathcal{B}$ in the beginning of the k-th iteration of the *while* loop of Algorithm 2. If the simulation stops before k iterations, set $u_k := \perp$. Return $(u_k)_{k=1,\ldots}$.

By Theorem 4.2, the transcripts $(u_k)_{k=1,2,\ldots}$ and $(v_k)_{k=1,2,\ldots}$ have the same distribution.

Now, since $\mathbb{E}[\mathcal{N}_1] \leq \chi(g) = d$, we have by *Markov's inequality* that the probability that $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$ sets QUERY_1 to 1 within first $10d$ iterations of the *while* loop, is at least 9/10. Note that conditioned on the event that the computation of $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$ is at vertex v of $\mathcal{B}$ that queries the input bit x_i, the probability that the random real number r generated in the same iteration lies in the interval $[\min_b \Pr_{x_i\sim\mu_b}[x_i = 0 \mid x \in v], \max_b \Pr_{x_i\sim\mu_b}[x_i = 0 \mid x \in v]]$ is exactly $\Delta(v)$. We have,

$$\begin{aligned}
\sum_{k=1}^{10d} \mathbb{E}[\Delta(v_k) \mid \mathcal{E}] &= \sum_{k=1}^{10d} \mathbb{E}[\Delta(u_k) \mid \mathcal{E}] \\
&\geq \Pr[\text{QUERY}_1 \text{ is set to to 1 within first } 10d \text{ iterations} \mid \mathcal{E}] \\
&\qquad \text{(by union bound)} \\
&\geq \Pr[\text{QUERY}_1 \text{ is set to to 1 within first } 10d \text{ iterations}] - \Pr[\overline{\mathcal{E}}] \\
&\geq \frac{9}{10} - \frac{1}{4} = \frac{13}{20}.
\end{aligned}$$

□

The following observation will be useful in the proof of Claim 6.5.

Observation 6.7. Let v be any node of $\mathcal{B}$, such that the associated subcube has non-empty intersections with the supports of both μ_0 and μ_1. Let $\mu_0' := \mu_0 \mid v$ and $\mu_1' := \mu_1 \mid v$. Let $\mathcal{B}_v$ denote the subtree of $\mathcal{B}$ rooted at v. Then $\mathcal{B}_v$ is an optimal decision tree for μ_0' and μ_1'.

Proof. If $\mathcal{B}_v$ is not an optimal decision tree for μ_0' and μ_1' then we could replace it by an optimal decision tree for μ_0' and μ_1', and for the resultant decision tree $\mathcal{B}'$, the expected value of $\mathcal{N}_1$ in $\mathcal{P}(\mathcal{B}', \mu_0, \mu_1)$ will be smaller than that in $\mathcal{P}(\mathcal{B}, \mu_0, \mu_1)$. This will contradict the optimality of $\mathcal{B}$. □

Proof of Claim 6.5. For $i = 0, \ldots, c-1$, let w be any vertex at depth $10id + 1$ consistent with $\mathcal{E}$, such that $\Pr_{x\sim\mu_0}[x \in w], \Pr_{x\sim\mu_1}[x \in w] \neq 0$. Consider the subtree T of $\mathcal{B}$ rooted at w. Let $w_1 := w$ and w_ℓ be the random vertex at depth ℓ of T, when T is run on a random input from $\mu \mid w$, or $\perp$ if T terminates before ℓ queries. By Observation 6.7, T is an optimal decision tree for distributions $\mu_0' := \mu_0 \mid w, \mu_1' := \mu_1 \mid w$. From Claim 6.6 we have that,

$$\sum_{\ell=1}^{10d} \mathbb{E}[\Delta(w_\ell) \mid \mathcal{E}] \geq \frac{13}{20}, \tag{6.6}$$

where $\Delta(w_\ell)$ is with respect to distributions μ_0' and μ_1'. Since $\mu_0' \mid w_\ell = \mu_0 \mid w_\ell$ and $\mu_1' \mid w_\ell = \mu_1 \mid w_\ell$, $\Delta(w_\ell)$ in (6.6) is also with respect to distributions μ_0 and μ_1. Now, when w is the random vertex v_{10id+1}, w_ℓ is the random vertex $v_{10id+\ell}$. Thus from (6.6) we have that,

$$\sum_{k=10id+1}^{10(i+1)d} \mathbb{E}[\Delta(v_k) \mid \mathcal{E}] \geq \frac{13}{20}. \tag{6.7}$$

The claim follows by adding (6.7) over $i = 0, \ldots, c-1$. □

Now we are ready to prove Claim 6.3. By setting $c = d$ and invoking Claim 6.5 we have,

$$\sum_{t=1}^{10d^2} \mathbb{E}[\Delta(v_t) \mid \mathcal{E}] \geq \frac{13d}{20}. \tag{6.8}$$

Let E_j be the event $\Pr_{x\sim\mu_0}[x \in v_j] \neq 0 \wedge \Pr_{x\sim\mu_1}[x \in v_j] \neq 0 \wedge v_j \neq \perp$ (i. e., v_j is a vertex of $\mathcal{B}$ and is not a leaf). Note that if $v_j \neq \perp$ and v_j is not a leaf of $\mathcal{B}$, v_j is determined by $(a_1, \ldots, a_{j-1})$ and vice versa, and hence $\mathsf{I}(a_j : g(x) \mid a_1, \ldots, a_{j-1}) = \mathsf{I}(x_{i_j} : g(x) \mid v_j)$. If $v_j = \perp$ or v_j is a leaf of $\mathcal{B}$, then $g(x)$ is determined by $(a_1, \ldots, a_{i-1})$, and $x_{i_j} = \perp$; thus, $\mathsf{I}(a_j : g(x) \mid a_1, \ldots, a_{j-1}) = \mathsf{I}(x_{i_j} : g(x) \mid v_j) = 0$.

Thus we have,

$$
\begin{aligned}
&\mathsf{I}(a_1, \ldots, a_{10d^2} : g(x)) \\
&= \sum_{j=1}^{10d^2} \mathsf{I}(a_j : g(x) \mid a_1, \ldots, a_{j-1}) \quad \text{(By the chain rule of mutual information (Theorem C.5))} \\
&= \sum_{j=1}^{10d^2} \mathsf{I}(x_{i_j} : g(x) \mid v_j) \quad \text{(From the discussion above)} \\
&\geq 8 \sum_{j=1}^{10d^2} \mathbb{E}\left[\mathbf{1}_{\mathsf{E}_j} \cdot \left[\Pr[g(x) = 0 \mid x \in v_j] \cdot \Pr[g(x) = 1 \mid x \in v_j] \cdot \Delta(v_j)\right]^2\right] \\
&\qquad \text{(From Claim 6.4)} \qquad (6.9) \\
&\geq 8 \sum_{j=1}^{10d^2} \Pr[\mathcal{E}] \cdot \mathbb{E}\left[\left[\Pr[g(x) = 0 \mid x \in v_j] \cdot \Pr[g(x) = 1 \mid x \in v_j] \cdot \Delta(v_j)\right]^2 \mid \mathcal{E}\right] \\
&\qquad \text{(Conditioned on } \mathcal{E}, \mathsf{E}_j \text{ happens with probability 1 for each } j \leq 10d^2) \\
&\geq 8 \sum_{j=1}^{10d^2} \frac{3}{4} \cdot \frac{1}{9} \cdot \mathbb{E}[\Delta(v_j)^2 \mid \mathcal{E}] \quad \text{(By the assumption } \Pr[\mathcal{E}] \geq \tfrac{3}{4}\text{)} \\
&= \frac{2}{3} \sum_{j=1}^{10d^2} \mathbb{E}[\Delta(v_j)^2 \mid \mathcal{E}] \\
&\geq \frac{2}{3} \sum_{j=1}^{10d^2} \left(\mathbb{E}[\Delta(v_j) \mid \mathcal{E}]\right)^2 \quad \text{(By Jensen's inequality)} \\
&\geq \frac{2}{3} \cdot \frac{1}{10d^2} \left(\sum_{j=1}^{10d^2} \mathbb{E}[\Delta(v_j) \mid \mathcal{E}]\right)^2 \quad \text{(By Cauchy-Schwarz inequality)} \\
&\geq \frac{1}{40}. \quad \text{(From (6.8))} \qquad (6.10)
\end{aligned}
$$

7 Tightness

In this section we prove Theorem 1.2. We construct a Boolean relation $f_0 \subseteq \{0,1\}^n \times \{0,1\}^n$ (i. e., $\mathcal{S} = \{0,1\}^n$) and a promise function $g_0 \subseteq \{0,1\}^n \times \{0,1\}$ (i. e., $m = n$), such that $\mathbb{R}_{4/9}(f_0) \in \Theta(\sqrt{n})$, $\mathbb{R}_{1/3}(g_0) \in \Theta(n)$ and $\mathbb{R}_{1/3}(f_0 \circ g_0^n) \in \Theta(n)$.

For strings $x = (x_1, \dots, x_n), z = (z_1, \dots, x_n)$ in $\{0,1\}^n$, let $x \oplus z$ be the string $(x_1 \oplus z_1, \dots, x_n \oplus z_n)$ obtained by taking their bitwise XOR. Let $|x|$ stand for the *Hamming weight* $|\{i \in [n] : x_i = 1\}|$ of x. We define f_0 as follows:

$$f_0 \stackrel{\text{def}}{=} \left\{(a,z) \in \{0,1\}^n \times \{0,1\}^n \,\middle|\, |a \oplus z| \le \frac{n}{2} - \sqrt{n}\right\}$$

Now we define g_0 by specifying $g_0^{-1}(0)$ and $g_0^{-1}(1)$.

$$g_0^{-1}(0) \stackrel{\text{def}}{=} \left\{(x,0) \,\middle|\, x \in \{0,1\}^n, |x| \le \frac{n}{2} - \sqrt{n}\right\},$$
$$g_0^{-1}(1) \stackrel{\text{def}}{=} \left\{(x,1) \,\middle|\, x \in \{0,1\}^n, |x| \ge \frac{n}{2} + \sqrt{n}\right\}.$$

g_0 is a gap-majority function with specific parameters. The two-party version of g_0 is the *gap-Hamming-Distance* problem. It has been studied before in the context of communication complexity [8]. We now determine the randomized query complexities of f_0, g_0 and $f_0 \circ g_0^n$.

Claim 7.1. *(i)* $\mathbb{R}_{4/9}(f_0) \in \Omega(\sqrt{n})$.

(ii) $\mathbb{R}_{1/3}(g_0) \in \Omega(n)$.

(iii) $\mathbb{R}_\epsilon(f_0 \circ g_0^n) \in O(n)$ *for any* $\epsilon \in \Omega(1)$.

Theorem 1.2 follows from Theorem 1.5 and Claim 7.1 with ϵ set to $\frac{1}{3}$.

Proof of Claim 7.1. (i) Assume that a deterministic protocol of cost k solves f_0 with respect to the uniform input distribution with error at most $4/9$. Such a protocol partitions $\{0,1\}^n$ into (at most) 2^k subcubes, each marked by some "answer" (an element from $\{0,1\}^n$). In particular, more than $2^n - 2^{n-4}$ points belong to subcubes of size at least 2^{n-k-4} – in other words, to subcubes of codimension at most $k+4$. As more than $\frac{15}{16}$ fraction of all points belong to such subcubes and the total protocol error is at most $4/9$, there exists at least one subcube of codimension $k+4$, on which the protocol errs with probability less than $\frac{4}{9} \cdot \frac{16}{15} < 1/2$.

The symmetry in the definition of f_0 allows us to assume without loss of generality that the subcube is the set $\tau \stackrel{\text{def}}{=} 0^{k+4} \circ \{0,1\}^{n-k-4}$, where "$\circ$" denotes string concatenation. It is easy to see that the "answer" that would minimize the error probability with respect to this subcube can be any binary string starting with "0^{k+4}", so let us assume that the actual label is 0^n. Let $\mathcal{U}_\ell$ denote uniform distribution on $\{0,1\}^n$. Then

$$\mathbf{Pr}\left[\mathit{error} \mid Z \in \tau\right] = \mathop{\mathbf{Pr}}_{Z' \sim \mathcal{U}_{n-k-4}}\left[|Z'| \le \frac{n}{2} - \sqrt{n}\right] < 1/2,$$

which implies that $k + 4 \ge 2\sqrt{n}$, as a uniformly-random binary string of length more than $n - 2\sqrt{n}$ would have more than $\frac{n}{2} - \sqrt{n}$ "ones" with probability at least $1/2$.

(ii) A randomized query protocol of cost k and error $1/3$ for g_0 would trivially imply existence of a randomized communication protocol of cost at most $2k$ and error $1/3$ for the bipartite problem *Gap-Hamming-Distance*:

$$GHD(X,Y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } |X \oplus Y| \le \frac{n}{2} - \sqrt{n}; \\ 1 & \text{if } |X \oplus Y| \ge \frac{n}{2} + \sqrt{n}; \\ * & \text{otherwise,} \end{cases}$$

and it has been demonstrated by Chakrabarti and Regev [8] that the complexity of this problem for any constant error is $\Omega(n)$.

(iii) Consider the following protocol for computing $f_0(g_0(x_1), \dots, g_0(x_n))$, where $x_i \in \{0,1\}^n$: For every $i \in [n]$, let $a_i = x_i(j_i)$, where $j_i \in [n]$ – that is, a_i is a uniformly-random bit of x_i. Then $\left|\{i | a_i = g_0(x_i)\}\right|$ – the expected number of "correctly guessed" values of a_i – is at least $\frac{n}{2} + \sqrt{n}$; intuitively, this means that the probability that $a_1, \dots, a_n$ is a right answer to $f_0(g_0(x_1), \dots, g_0(x_n))$ is "non-trivially high" – to "boost" this probability, we will use several "probes" from every x_i.

For m a power of 3, let

$$\tau_m : \{0,1\}^m \to \{0,1\}$$

be the Boolean function represented by a complete ternary tree of depth $\log_3 m$ with leaves labelled by the m input variables naturally ordered and vertices computing the majority:

$$\mathrm{Maj}_3(z_1, z_2, z_3) \stackrel{\text{def}}{=} (z_1 \wedge z_2) \vee (z_1 \wedge z_3) \vee (z_2 \wedge z_3).$$

Protocol: For an integer d_ϵ (to be fixed later), independently choose $j_{i,k} \in [n]$ for $i \in [n]$ and $k \in [3^{d_\epsilon}]$. Let $a_i \stackrel{\text{def}}{=} \tau_{3^{d_\epsilon}}\left(x_i(j_{i,1}), \dots, x_i(j_{i,3^{d_\epsilon}})\right)$ and output "$a_1, \dots, a_n$".[8]

The behaviour of a single $\mathrm{Maj}_3(x_i(j_{i,1}), x_i(j_{i,2}), x_i(j_{i,3}))$ is as follows. For $\delta \in (0, 1/2]$:

$$\begin{aligned} &\mathbf{Pr}\left[\mathrm{Maj}_3(x_i(j_{i,1}), x_i(j_{i,2}), x_i(j_{i,3})) = g_0(x_i) \,\middle|\, \left||x_i| - \frac{n}{2}\right| = \delta \cdot n\right] \\ &= (\frac{1}{2} + \delta)^3 + 3 \cdot (\frac{1}{2} + \delta)^2 \cdot (\frac{1}{2} - \delta) = \frac{1}{2} + \frac{3\delta}{2} - 2\delta^3 \\ &> \min\left\{\frac{1}{2} + \frac{5\delta}{4}, \frac{3}{4}\right\}. \end{aligned}$$

The same analysis applies to every node in the tree representation of the function τ, so:

$$\mathbf{Pr}\left[a_i = g_0(x_i) \,\middle|\, \left||x_i| - \frac{n}{2}\right| \ge \sqrt{n}\right] \;>\; \min\left\{\frac{1}{2} + \frac{(5/4)^{d_\epsilon}}{\sqrt{n}}, \frac{3}{4}\right\}.$$

[8] Here we are using $\tau_{3^{d_\epsilon}}(\cdot)$ for approximating the value of $g_0(x_i)$, instead of the majority function, which would look more natural here (moreover, it is easy to see that the function τ is somewhat less efficient for the task). The symmetry appearing in the tree representation of the function τ simplifies its analysis considerably, while the resulting complexity is sufficient for our needs.

As long as $d_\epsilon \in o(\log n)$, it holds that $(5/4)^{d_\epsilon} \leq \frac{\sqrt{n}}{4}$ for sufficiently large n, and so:

$$\mathbf{Pr}\left[a_i = g_0(x_i) \mid g_0(x_i) \neq *\right] \;>\; \frac{1}{2} + \frac{(5/4)^{d_\epsilon}}{\sqrt{n}}.$$

Note that $a_1, \ldots, a_n$ is a wrong answer to $f_0\left(g_0(x_1), \ldots, g_0(x_n)\right)$ only if $\left|\{i | a_i = g_0(x_i)\}\right| < \frac{n}{2} + \sqrt{n}$, so by a Bernstein-type tail bound, as given in [2], it follows that

$$\Pr[\text{The protocol errs}] < \exp\left(-1/2 \cdot \left((5/4)^{d_\epsilon} - 1\right)^2\right).$$

Accordingly, $d_\epsilon \in O(1)$ suffices for any $\epsilon \in \Omega(1)$ and the result follows.

□

A Minimax principle: proof of Fact 2.4

Fix an integer ℓ. Let $\mathcal{D}_\ell$ be the finite set of all deterministic query algorithms on k bits with worst-case complexity at most ℓ. Let $\mathcal{H}_k := \{0,1\}^k$. For algorithm $\mathrm{A} \in \mathcal{D}_\ell$ and input $x \in \mathcal{H}_k$, let $\mathbb{E}(\mathrm{A}, x) = 1$ if $(x, \mathrm{A}(x)) \notin h$, and 0 otherwise. By von Neumann's minimax principle,

$$\min_\sigma \max_\mu \sum_{\mathrm{A} \in \mathcal{D}_\ell, x \in \mathcal{H}_k} \sigma(\mathrm{A}) \mathbb{E}(\mathrm{A}, x) \mu(x) = \max_\mu \min_\sigma \sum_{\mathrm{A} \in \mathcal{D}_\ell, x \in \mathcal{H}_k} \sigma(\mathrm{A}) \mathbb{E}(\mathrm{A}, x) \mu(x), \tag{A.1}$$

where σ and μ range over probability distributions over $\mathcal{D}_\ell$ and $\mathcal{H}_k$, respectively. Note that in equation (A.1), we can assume that the maximum in the left hand side is over point distributions on $\mathcal{H}_k$, i. e., distributions that assign weight 1 to some input $x \in \mathcal{H}_k$. Similarly we can assume that the minimum in the right hand side is over point distributions on $\mathcal{D}_\ell$. Thus we have that,

$$\min_\sigma \max_{x \in \mathcal{H}_k} \sum_{\mathrm{A} \in \mathcal{D}_\ell} \sigma(\mathrm{A}) \mathbb{E}(\mathrm{A}, x) = \max_\mu \min_{\mathrm{A} \in \mathcal{D}_\ell} \sum_{x \in \mathcal{H}_k} \mathbb{E}(\mathrm{A}, x) \mu(x). \tag{A.2}$$

From equation (A.2) it follows that

$$\begin{aligned}
\mathbb{R}_\epsilon(h) &= \min\left\{\ell \,\middle|\, \min_\sigma \max_{x \in \mathcal{H}_k} \sum_{\mathrm{A} \in \mathcal{D}_\ell} \sigma(\mathrm{A}) \mathbb{E}(\mathrm{A}, x) \leq \epsilon\right\} \\
&\qquad\qquad \text{(where } \sigma \text{ ranges over all probability distributions on } \mathcal{D}_\ell\text{)} \\
&= \min\left\{\ell \,\middle|\, \max_\mu \min_{\mathrm{A} \in \mathcal{D}_\ell} \sum_{x \in \mathcal{H}_k} \mathbb{E}(\mathrm{A}, x) \mu(x) \leq \epsilon\right\} \\
&\qquad\qquad \text{(where } \mu \text{ ranges over all probability distributions on } \mathcal{H}_k\text{)} \\
&= \max_\mu \min\left\{\ell \,\middle|\, \min_{\mathrm{A} \in \mathcal{D}_\ell} \sum_{x \in \mathcal{H}_k} \mathbb{E}(\mathrm{A}, x) \mu(x) \leq \epsilon\right\} \\
&= \mathbb{D}^\mu_\epsilon(h).
\end{aligned}$$

B Alternative characterization of sabotage complexity

We first go over the standard definition of sabotage complexity from [7]. Let $g \subseteq \{0,1\}^m \times \{0,1\}$ be a partial function. From g, define a partial function $g_{\text{sab}} : P \to \{*, \dagger\}$, where now $P \subseteq \{0,1,*,\dagger\}^n$ is defined in the following way. Let $P^* \subseteq \{0,1,*\}$ be the largest set such that for all $z \in P^*$ there exist x, y with $g(x) \neq g(y)$ and both x and y are consistent with the non-star coordinates of z. Define $P^\dagger \subseteq \{0,1,\dagger\}$ analogously with $\dagger$ instead of $*$. Then $P = P^* \cup P^\dagger$. Finally, define $g_{\text{sab}}(z) = *$ if $z \in P^*$ and $g_{\text{sab}}(z) = \dagger$ if $z \in P^\dagger$. The sabotage complexity of g is defined as $\mathbb{RS}(g) = R_0(g_{\text{sab}})$.

For a tree T computing g, and strings x, y such that $g(x) \neq g(y)$, let $\text{sep}_T(x,y)$ denote the depth of the node v in T such that x and y both reach v yet $x_{q(v)} \neq y_{q(v)}$ where $q(v)$ is the index queried at node v. We have the following alternative characterization of sabotage complexity.

Theorem B.1. *Let $g \subseteq \{0,1\}^m \times \{0,1\}$ be a partial function. Then*

$$\begin{aligned}\mathbb{RS}(g) &= \min_{\mathcal{T}} \max_{\substack{x,y \\ g(x)\neq g(y)}} \mathbb{E}_{T\sim\mathcal{T}}[\text{sep}_T(x,y)] \\ &= \max_p \min_T \mathbb{E}_{(x,y)\sim p}[\text{sep}_T(x,y)] \ .\end{aligned}$$

In the first equation the minimum is taken over zero-error randomized algorithms $\mathcal{T}$ for g. In the second equation, the maximum is taken over distributions over pairs (x,y) where $g(x) = 0, g(y) = 1$, and the minimum is taken over deterministic trees T computing g.

Proof. That the right hand side of the first line is equal to the second line follows by von Neumann's minimax theorem [18].

Now we focus on establishing the first line. We first show that $\mathbb{RS}(g)$ is at most the right hand side of the first line. Let $\mathcal{T}^*$ achieve the minimum of the expression on the right hand side. Let $z \in P$ be any sabotaged input. Then there are x^*, y^* with $g(x^*) \neq g(y^*)$ such that x^* and y^* only differ where z has special symbols. Thus any query that separates x^* and y^* will also find a special symbol. The expected number of queries to separate x^* and y^* is at most $\max_{x,y} \mathbb{E}_{T\sim\mathcal{T}^*}[\text{sep}_T(x,y)]$, thus the left hand side is at most the right hand side.

For the other direction, let $\mathcal{T}^*$ be an optimal zero-error randomized algorithm computing g_{sab}. For any x, y with $g(x) \neq g(y)$ we can create $z^* \in P^*$ such that z^* has $*$ in those positions where x, y disagree, and z^* agrees with x, y in those positions where they agree with each other.. Let $z^\dagger$ equal z^* with $*$ replaced by $\dagger$. Now $\mathcal{T}^*$ is able to distinguish between z^* and $z^\dagger$ using an expected number of queries that is at most $\mathbb{RS}(g)$. Any query that distinguishes z^* and $z^\dagger$ is also a query that separates x and y, as z_* and $z_\dagger$ only differ where x and y do. This means

$$\mathbb{E}_{T\sim\mathcal{T}^*}[\text{sep}_T(x,y)] \leq \mathbb{RS}(g) \ ,$$

showing that the right hand side is at most the left hand side. □

C Information Theory

Let X be a random variable supported on a finite set $\{x_1, \ldots, x_s\}$. Let $\mathcal{E}$ be any event in the same probability space. Let $\mathbb{P}[\cdot]$ denote the probability of any event. The *conditional entropy* $\mathsf{H}(X \mid \mathcal{E})$ of X conditioned on $\mathcal{E}$ is defined as follows.

Definition C.1 (Conditional entropy)**.**

$$\mathsf{H}(X \mid \mathcal{E}) := \sum_{i=1}^{s} \mathbb{P}[X = x_i \mid \mathcal{E}] \log_2 \frac{1}{\mathbb{P}[X = x_i \mid \mathcal{E}]}.$$

An important special case is when $\mathcal{E}$ is the entire sample space. In that case the above conditional entropy is referred to as the entropy $\mathsf{H}(X)$ of X.

Definition C.2 (Entropy)**.**

$$\mathsf{H}(X) := \sum_{i=1}^{s} \mathbb{P}[X = x_i] \log_2 \frac{1}{\mathbb{P}[X = x_i]}.$$

Let Y be another random variable in the same probability space as X, taking values from a finite set $\{y_1, \ldots, y_t\}$. Then the conditional entropy of X conditioned on Y, $\mathsf{H}(X \mid Y)$, is defined as follows.

Definition C.3.

$$\mathsf{H}(X \mid Y) = \sum_{i=1}^{t} \mathbb{P}[Y = y_i] \cdot \mathsf{H}(X \mid Y = y_i).$$

Definition C.4 (Mutual information)**.** Let X, Y and Z be two random variables in the same probability space, taking values from finite sets. The mutual information between X and Y conditioned on Z, $\mathrm{I}(X; Y \mid Z)$, is defined as follows.

$$\mathrm{I}(X; Y \mid Z) := \mathsf{H}(X \mid Z) - \mathsf{H}(X \mid Y, Z).$$

It can be shown that $\mathrm{I}(X; Y \mid Z)$ is symmetric in X and Y: $\mathrm{I}(X; Y \mid Z) = \mathrm{I}(Y; X \mid Z) = \mathsf{H}(Y \mid Z) - \mathsf{H}(Y \mid X, Z)$.

Theorem C.5 (Chain rule of mutual information)**.** *Let $X_1, \ldots, X_k, Y, Z$ be random variables in the same probability space, taking values from finite sets. Then,*

$$\mathrm{I}(X_1, \ldots, X_k : Y \mid Z) = \sum_{i=1}^{k} \mathrm{I}(X_i : Y \mid Z, X_1, \ldots, X_{i-1}).$$

Definition C.6 (Kullback-Leibler Divergence)**.** Given two probability distributions P and $\mathbb{Q}$ on a finite set $\mathcal{U}$, the *Kullback-Leibler divergence* from $\mathbb{Q}$ to P, denoted by $\mathbb{D}(\mathrm{P}||\mathbb{Q})$, is defined as:

$$\mathbb{D}(\mathrm{P}||\mathrm{Q}) := -\sum_{u \in \mathcal{U}} \mathrm{P}(u) \log \frac{\mathrm{P}(u)}{\mathrm{Q}(u)}.$$

Given two random variables, X and Y, taking values in a finite set $\mathcal{U}$, let $\mathsf{X} \otimes \mathsf{Y}$ be the distribution over ordered pairs of elements of $\mathcal{U}$ (i. e., over elements of $\mathcal{U} \times \mathcal{U}$), where the first and the second element are sampled independently according to the distributions of X and Y, respectively. Let (X, Y) denote the joint distribution of X and Y. The following fact can be easily verified.

Fact C.7. $\mathsf{I}(\mathsf{X} : \mathsf{Y}) = D((\mathsf{X}, \mathsf{Y})||(\mathsf{X} \otimes \mathsf{Y}))$.

Definition C.8. Given two probability distributions P and Q on a finite set $\mathcal{U}$, the L_1-*distance* between P and Q, denoted by $||\mathrm{P} - \mathrm{Q}||_1$, is defined as:

$$||\mathrm{P} - \mathrm{Q}||_1 := \sum_{u \in \mathcal{U}} |\mathrm{P}(u) - \mathrm{Q}(u)|.$$

Pinsker's inequality, stated below, bounds $\mathbb{D}(P||Q)$ in terms of $|\mathrm{P}(u) - \mathrm{Q}(u)|$ from below.

Theorem C.9 (Pinsker's inequality)**.** *Given two probability distributions P and Q on a finite set $\mathcal{U}$,*

$$\mathbb{D}(P||Q) \geq \frac{1}{2}||P - Q||_1^2.$$

Acknowledgements. We thank Rahul Jain for useful discussions. We thank Srijita Kundu and Jevgēnijs Vihrovs for their helpful comments on the manuscript. We thank Yuval Filmus for suggesting to look at the min-max version of conflict complexity, which led to the development of max-conflict complexity. We thank the anonymous reviewers for their helpful comments.

This research was supported by the National Research Foundation, Singapore, and A*STAR under its CQT Bridging Grant and its Quantum Engineering Programme under grant NRF2021-QEP2-02-P05. Part of this work was conducted while T. L. and S. S. were at the Nanyang Technological University and the Centre for Quantum Technologies, supported by the Singapore National Research Foundation under NRF RF Award No. NRF-NRFF2013-13. Part of this work was done while D.G. was visiting the Centre for Quantum Technologies at the National University of Singapore.

References

[1] SCOTT AARONSON: Quantum certificate complexity. *J. Comput. System Sci.*, 74(3):313–322, 2008. Preliminary version in CCC'03. [doi:10.1016/j.jcss.2007.06.020, arXiv:quant-ph/0210020] 12

[2] DANA ANGLUIN AND LESLIE G. VALIANT: Fast probabilistic algorithms for Hamiltonian circuits and matchings. *J. Comput. System Sci.*, 18(2):155–193, 1979. Preliminary version in STOC'77. [doi:10.1016/0022-0000(79)90045-X] 29

[3] ANURAG ANSHU, DMITRY GAVINSKY, RAHUL JAIN, SRIJITA KUNDU, TROY LEE, PRIYANKA MUKHOPADHYAY, MIKLOS SANTHA, AND SWAGATO SANYAL: A composition theorem for

randomized query complexity. In *Proc. 37th Found. Softw. Techn. Theoret. Comp. Sci. Conf. (FSTTCS'17)*, pp. 10:1–13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [doi:10.4230/LIPIcs.FSTTCS.2017.10, arXiv:1706.00335] 5

[4] BOAZ BARAK, MARK BRAVERMAN, XI CHEN, AND ANUP RAO: How to compress interactive communication. *SIAM J. Comput.*, 42(3):1327–1363, 2013. [doi:10.1137/100811969] 6, 8, 21

[5] SHALEV BEN-DAVID AND ERIC BLAIS: A tight composition theorem for the randomized query complexity of partial functions. In *Proc. 61st FOCS*, pp. 240–246. IEEE Comp. Soc., 2020. [doi:10.1109/FOCS46700.2020.00031, arXiv:2002.10809] 3, 4, 9

[6] SHALEV BEN-DAVID, ERIC BLAIS, MIKA GÖÖS, AND GILBERT MAYSTRE: Randomised composition and small-bias minimax. In *Proc. 63rd FOCS*, pp. 624–635. IEEE Comp. Soc., 2022. [doi:10.1109/FOCS54457.2022.00065, arXiv:2208.12896] 9

[7] SHALEV BEN-DAVID AND ROBIN KOTHARI: Randomized query complexity of sabotaged and composed functions. *Theory of Computing*, 14(5):1–27, 2018. Preliminary version in ICALP'16. [doi:10.4086/toc.2018.v014a005, arXiv:1605.09071] 4, 5, 12, 30

[8] AMIT CHAKRABARTI AND ODED REGEV: An optimal lower bound on the communication complexity of Gap-Hamming-Distance. *SIAM J. Comput.*, 41(5):1299–1317, 2012. Preliminary version in STOC'11. [doi:10.1137/120861072, arXiv:1009.3460] 27, 28

[9] DMITRY GAVINSKY, TROY LEE, AND MIKLOS SANTHA: On the randomised query complexity of composition, 2018. [arXiv:1801.02226] 12

[10] DMITRY GAVINSKY, TROY LEE, MIKLOS SANTHA, AND SWAGATO SANYAL: A composition theorem for randomized query complexity via max-conflict complexity. In *Proc. 46th Internat. Colloq. on Automata, Languages, and Programming (ICALP'19)*, pp. 64:1–13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.ICALP.2019.64] 1, 9

[11] JUSTIN GILMER, MICHAEL SAKS, AND SRIKANTH SRINIVASAN: Composition limits and separating examples for some Boolean function complexity measures. *Combinatorica*, 36(3):265–311, 2016. Preliminary version in CCC'13. [doi:10.1007/s00493-014-3189-x, arXiv:1306.0630] 12

[12] PETER HØYER, TROY LEE, AND ROBERT ŠPALEK: Negative weights make adversaries stronger. In *Proc. 39th STOC*, pp. 526–535. ACM Press, 2007. [doi:10.1145/1250790.1250867, arXiv:quant-ph/0611054] 3

[13] YAQIAO LI: Conflict complexity is lower bounded by block sensitivity. *Theoret. Comput. Sci.*, 856:169–172, 2021. [doi:10.1016/j.tcs.2020.12.038, arXiv:1810.08873] 12

[14] ASHLEY MONTANARO: A composition theorem for decision tree complexity. *Chicago J. Theoret. Comp. Sci.*, 2014(6):1–8. [doi:10.4086/cjtcs.2014.006] 3

[15] BEN W. REICHARDT: Reflections for quantum query algorithms. In *Proc. 22nd Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'11)*, pp. 560–569. SIAM, 2011. [doi:10.1137/1.9781611973082.44, arXiv:1005.1601] 3

[16] SWAGATO SANYAL: A composition theorem via conflict complexity, 2018. [arXiv:1801.03285] 12

[17] AVISHAY TAL: Properties and applications of boolean function composition. In *Proc. 4th Innovations in Theoret. Comp. Sci. Conf. (ITCS'13)*, pp. 441–454. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013. [doi:10.1145/2422436.2422485, ECCC:TR12-163] 3, 12

[18] JOHN VON NEUMANN: Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928. EuDML. 8, 30

AUTHORS

Dmytro Gavinsky
Researcher
Institute of Mathematics
Czech Academy of Sciences
115 67 Žitna 25, Praha 1, Czechia
gavinsky@math.caz.cz
https://users.math.cas.cz/~gavinsky/

Troy Lee
Associate Professor
Centre for Quantum Software and Information
University of Technology Sydney
troyjlee@gmail.com
https://troylee.org/

Miklos Santha
Research Director emeritus
CNRS, IRIF, Université Paris Cité, F-75013 Paris, France
and
Research Professor and Principal Investigator
CQT, National University of Singapore
Singapore 117543, Singapore
miklos.santha@gmail.com
https://www.irif.fr/~santha/

Swagato Sanyal
Assistant Professor
Department of Computer Science and Engineering
IIT Kharagpur
India
swagato@cse.iitkgp.ac.in
http://cse.iitkgp.ac.in/~swagato/

ABOUT THE AUTHORS

In the good old days DMYTRO GAVINSKY studied at the Technion - Israel Institute of Technology and at the University of Calgary. Thanks to the support of his scientific advisers Nader Bshouty, Richard Cleve and John Watrous, he graduated from both institutions and launched his own research activities.

TROY LEE received his Ph. D. in 2006 from the University of Amsterdam, where his advisor was Harry Buhrman. He is currently an Associate Professor at the University of Technology Sydney.

MIKLOS SANTHA received his Diploma in Mathematics in 1979 from Eötvös University in Budapest, and his Ph. D. in Mathematics in 1983 from the Université Paris 7. His advisor was Jacques Stern. After having been a CNRS researcher between 1988 and 2021 he is currently Research Director emeritus at the Université Paris Cité. He is also Research Professor and Principal Investigator at CQT, National University of Singapore.

SWAGATO SANYAL graduated from TIFR, Mumbai in 2017; his advisor was Prahladh Harsha. The work presented in this article was partly carried out when he was a postdoctoral fellow at Centre for Quantum Technologies, National University of Singapore. He is currently an Assistant Professor at the Department of Computer Science and Engineering, IIT Kharagpur.