The Query Complexity of Certification

The Query Complexity of Certification

Guy Blanc
Stanford University
Stanford, CA, USA

Caleb Koch
Stanford University
Stanford, CA, USA

Jane Lange
MIT
Cambridge, MA, USA

Li-Yang Tan
Stanford University
Stanford, CA, USA

ABSTRACT

We study the problem of *certification*: given queries to a function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\leq k$ and an input $x^\star$, output a size-k certificate for f's value on $x^\star$.

For monotone functions, a classic local search algorithm of Angluin accomplishes this task with n queries, which we show is optimal for local search algorithms. Our main result is a new algorithm for certifying monotone functions with $O(k^8 \log n)$ queries, which comes close to matching the information-theoretic lower bound of $\Omega(k \log n)$. The design and analysis of our algorithm are based on a new connection to threshold phenomena in monotone functions.

We further prove exponential-in-k lower bounds when f is non-monotone, and when f is monotone but the algorithm is only given random examples of f. These lower bounds show that assumptions on the structure of f and query access to it are both necessary for the polynomial dependence on k that we achieve.

CCS CONCEPTS

• **Theory of computation** → *Complexity theory and logic*; *Oracles and decision trees*; *Boolean function learning*; *Query learning*; Lower bounds and information complexity;

KEYWORDS

Boolean function complexity, certificate complexity, query algorithms, monotone functions

ACM Reference Format:
Guy Blanc, Caleb Koch, Jane Lange, and Li-Yang Tan. 2022. The Query Complexity of Certification. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3519993

1 INTRODUCTION

Given a function $f : \{0,1\}^n \to \{0,1\}$ and an input $x^\star$, *why* does f output $f(x^\star)$ on $x^\star$? Among the many possibilities for what constitutes such an "explanation", the notion of *certificates* is perhaps the simplest: a set $S \subseteq [n]$ of $x^\star$'s coordinates that determines f's value on $x^\star$. That is, $f(y) = f(x^\star)$ for all y that agree with $x^\star$ on the coordinates in S.

It is natural to seek *small* certificates, i.e. succinct explanations: the smaller S is, the more inputs it covers, and the more general it is as an explanation. This leads us to the following standard definition from complexity theory:

Definition 1 (Certificate complexity). *For a function $f : \{0,1\}^n \to \{0,1\}$ and an input $x^\star$, the complexity of certifying f's value on $x^\star$ is the quantity:*

$$C(f, x^\star) := \min_{S \subseteq [n]} \left\{ |S| : f(y) = f(x^\star) \text{ for all } y \text{ s.t. } y_S = x^\star_S \right\}.$$

The certificate complexity of f is the quantity

$$C(f) := \max_{x \in \{0,1\}^n} \{C(f, x)\}.$$

We can now state the algorithmic problem that we study in this work, that of efficiently finding small certificates:

> **Certification Problem:** *Given queries to a function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\leq k$ and an input $x^\star$, output a size-k certificate for f's value on $x^\star$.*

Motivation. In addition to being a basic and natural problem, this is also an abstraction of a problem of interest in *explainable machine learning*, where f represents a black box model that we seek to explain the predictions of. Modern machine learning algorithms, powered by large amounts of computational resources and trained on massive datasets, produce models that perform very well, but are so complicated that they are essentially inscrutable black boxes. This is a concern as we increasingly delegate weighty decisions to these models. The field of explainable machine learning seeks to address this by developing techniques to explain the predictions of these models [6, 14].

There are numerous notions of "explanations" in this literature [2, 13, 15, 21, 25–27]; Ribero, Singh, and Guestrin [22] were the first to propose certificates. Their work introduced a relaxed "approximate" notion of certificates, where the set S of coordinates *mostly* determines f's value rather than fully determines it, and "mostly" is measured with respect to a distribution over inputs. We discuss [22], this notion of "approximate certificates", and corresponding approximate certification algorithms in more detail in Section 1.2.

1.1 Our Results

1.1.1 Local Search for Monotone Functions and Its Limitations. The certification problem can be viewed as the problem of efficiently finding an "f-monochromatic" subcube in $\{0,1\}^n$ of codimension

STOC '22, June 20–24, 2022, Rome, Italy

ACM ISBN 978-1-4503-9264-8/22/06…$15.00
https://doi.org/10.1145/3519935.3519993

$\leq k$ containing $x^\star$, where a subcube is f-monochromatic if f takes the same value on all inputs in that subcube. From this perspective, it is natural to proceed by *local search*: first query f on $x^\star$ and its immediate Hamming neighbors, and iteratively expand this neighborhood until it contains an f-monochromatic subcube of the desired size.

Indeed, a classic algorithm due to Angluin [1] shows how such a local search can be carried out systematically for *monotone* functions, and solves the certification problem with just n queries:

Angluin's algorithm: *Given queries to a monotone function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\leq k$ and an input $x^\star$, Angluin's algorithm makes n queries to f and returns a size-k certificate for f's value on $x^\star$.*

Angluin's algorithm is a modification of a similar algorithm given by Valiant [28].

We begin by observing that Angluin's algorithm is optimal among local search algorithms. We consider a local search algorithm to be any algorithm whose first query is $x^\star$, and whose subsequent queries are Hamming neighbors of some input that has been previously queried. In other words, at any point in the execution of a local search algorithm, the set of inputs that have been queried so far forms a connected subgraph of $\{0,1\}^n$ containing $x^\star$. We show the following lower bound:

Claim 1.1 (Lower bound against local search algorithms). *For any $\varepsilon > 0$ the following holds. Any local search algorithm solving the certification problem for monotone functions $f : \{0,1\}^n \to \{0,1\}$ must have query complexity $\Omega(\varepsilon n)$, even if f is promised to have certificate complexity $k = 1$ and even if the algorithm is only required to return a size-$\Omega(\varepsilon n)$ certificate with probability ε.*

1.1.2 Near-Optimal Certification Algorithm for Monotone Functions. Our main result is an algorithm for certifying monotone functions that is substantially more efficient than Angluin's:

Theorem 1 (Efficient certification of monotone functions). *Given queries to a monotone function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\leq k$ and an input $x^\star$, our algorithm makes $O(k^8 \log n)$ queries to f and w.h.p. returns a size-k certificate for f's value on $x^\star$.*

As one would expect given Claim 1.1, our algorithm does not proceed by local search. In fact, our algorithm takes the exact *opposite* approach. A local search algorithm for monotone functions starts with the trivial certificate $S = \{i \in [n] : x^\star_i = f(x^\star_i)\}$ and trims it down in size by removing coordinates that are "irrelevant to S". Our algorithm proceeds the opposite way: we start with the empty set $S = \emptyset$ and add to it coordinates that we deem "important". We describe our approach in detail in Section 2.

We complement Theorem 1 with a lower bound showing that the query complexity of our algorithm is near optimal, even if the algorithm only has to return a certificate of size $\ell \gg k$:

Claim 1.2 (Lower bound for monotone functions). *For any $c < 1$ and any $k \leq \ell \leq n^c$, let $\mathcal{A}$ be an algorithm which, given query access to a monotone function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\leq k$ and an input $x^\star$, returns a size-ℓ certificate for f's value on $x^\star$ w.h.p. The query complexity of $\mathcal{A}$ must be $\Omega(k \log n)$.*

1.1.3 Algorithms and Lower Bounds for Other Settings. Finally, we study the extent to which the setting of Theorem 1 can be relaxed: what if f is an arbitrary function, one that is not necessarily monotone? What if the algorithm is only given uniformly-distributed random examples $(\boldsymbol{x}, f(\boldsymbol{x}))$ instead of query access to f? We obtain fairly tight upper and lower bounds for both these settings. Table 1 summarizes these bounds and contrasts them with our results as described in the previous subsection:

The exponential-in-k lower bounds for these alternative settings (the last two rows of Table 1) show that some assumption on the structure of f, such as monotonicity, and query access to it are both necessary for the polynomial dependence on k that we achieve in Theorem 1. As in Claim 1.2, these lower bounds hold even if the algorithm is only required to return a size-ℓ certificate where ℓ can be significantly larger than k; we defer the precise statements to the body of the paper.

1.2 Prior Work on "Approximate" and Exact Certificates

We discuss two works from the explainable machine learning literature, [22] and [3], that are direct precursors to ours.

[22]. Ribero, Singh, and Guestrin were the first to propose certificates as explanations for black box machine learning models. They introduced a relaxed notion of certificates that allows for errors[1]:

Definition 2 (Approximate certificates [22]). *For a function $f : \{0,1\}^n \to \{0,1\}$, an input $x^\star$, a distribution $\mathcal{D}$ over $\{0,1\}^n$, and $\varepsilon > 0$, we say that a set $S \subseteq [n]$ is an ε-error certificate for f's value on $x^\star$ with respect to $\mathcal{D}$ if $\Pr_{\boldsymbol{y} \sim \mathcal{D}}[\, f(\boldsymbol{y}) \neq f(x^\star) \mid \boldsymbol{y}_S = x^\star_S \,] \leq \varepsilon$.*

[22]'s work was empirical in nature: their paper demonstrated, through experiments and a user study, the effectiveness of succinct certificates as explanations. Their work also gave heuristics for finding succinct approximate certificates, but these heuristics do not come with provable performance guarantees.

[22]'s work has been influential in explainable machine learning. For more, see the discussion of their work in the book [18, Chapter §5.9], and the open source library [12] for implementation details of their heuristics.

[3]. Motivated by [22], [3] gave an algorithm for finding succinct approximate certificates that comes with performance guarantees with respect to the *uniform distribution*:

Theorem 2 ([3]'s approximate certification algorithm; informal). *Let $\mathcal{U}$ denote the uniform distribution over $\{0,1\}^n$ and $\varepsilon > 0$. Given query access to $f : \{0,1\}^n \to \{0,1\}$ with "ε-error certificate complexity" $\leq k$ and an input $x^\star$, [3]'s algorithm makes $\mathrm{poly}(k, 1/\varepsilon, n)$ queries to f and returns a set of coordinates $S(x^\star)$.*

With probability $\geq 1 - \varepsilon$ over $\boldsymbol{x}^\star \sim \mathcal{U}$, the set $S(\boldsymbol{x}^\star)$ is an ε-error certificate for f's value on $\boldsymbol{x}^\star$ with respect to $\mathcal{U}$ and $|S(\boldsymbol{x}^\star)| \leq \mathrm{poly}(k, 1/\varepsilon)$.

[1] They termed such explanations *anchors*, which has since become standard in the explainable machine learning literature. We stick with the term certificates in our description of their results.

Table 1: Bounds on the query complexity of certification.

Algorithm is given:	Upper bound	Lower bound
Queries to monotone f, and proceeds by local search	Angluin's algorithm: n queries	Claim 1.1: $\Omega(n)$ queries
Queries to monotone f	Theorem 1: $O(k^8 \log n)$ queries	Claim 1.2: $\Omega(k \log n)$ queries
Queries to arbitrary f	Claim 8.1: $O(2^k k \log n)$ examples	Claim 8.3: $\Omega(2^k + k \log n)$ queries
Random examples of monotone f		Claim 8.6: $\Omega(2^k + k \log n)$ examples

Comparing Theorem 2 to our algorithm in Theorem 1, we see that Theorem 2 applies to all functions whereas Theorem 1 only applies to monotone ones. On the other hand, there are two sources of errors in Theorem 2, neither of which are present in Theorem 1: the guarantees of [3]'s algorithm only hold for most $x^\star$ and not for all of them, and the certificates returned are ε-error certificates and not actual certificates. Even if one is willing to tolerate both sources of errors, the fact that they are measured with respect to the uniform distribution remains a significant shortcoming—this was identified in [3] as the main limitation of their result.

A primary motivation for our work was to develop certification algorithms that, like [3]'s, come with provable performance guarantees, but where these guarantees hold in the much more challenging *errorless* setting.

Other related work on finding certificates. There has been significant work on finding prime implicants in the ML and AI literature (see e.g. [5, 7–9] and the references therein), including for monotone functions [17, 24]. In our terminology, a prime implicant is a 1-certificate which is minimal under set inclusion (relatedly a minimal 0-certificate is a prime implicant for $\neg f$). These algorithms for computing prime implicants all have worst-case query complexity and runtime that is at least linear in n. In contrast, our algorithm has only a logarithmic dependence on n and always returns a prime implicant.

2 OVERVIEW OF OUR ALGORITHM AND ITS ANALYSIS

Before describing our algorithm, we first give an overview of Angluin's and [3]'s algorithms, in tandem with a discussion of how these algorithms led to ours and how ours differs from them. Throughout this section, let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function and suppose without loss of generality that $f(x^\star) = 1$ for the input $x^\star$ that we seek to certify.

Angluin's algorithm. By the monotonicity of f, the set $S_{x^\star} := \{i \in [n] : x^\star_i = 1\}$ is certainly a certificate for f's value at $x^\star$. The assumption that f has certificate complexity $\leq k$ implies the existence of at least one subset $T \subseteq S_{x^\star}$ of size $\leq k$ that remains a certificate for f's value at $x^\star$. The goal of Angluin's algorithm is to find one of them.

Definition 3 (Irrelevant coordinate of a certificate). *For a function f, an input $x^\star$, a certificate $S \subseteq [n]$ for f's value at $x^\star$, and a coordinate $i \in S$, we say that i is irrelevant to S if $S \setminus \{i\}$ remains a certificate for f's value at $x^\star$, and otherwise say that it is relevant.*

Angluin's algorithm starts with $S_{x^\star}$ and trims it down in size, removing irrelevant coordinates one by one, all the while maintaining the invariant that the current set remains a certificate. A naive implementation of this plan results in a query complexity of $\Theta(|S_{x^\star}|^2)$. A simple but key observation yields an improved query complexity of $O(|S_{x^\star}|) \leq O(n)$: if i is relevant for a certificate S, it remains relevant for any certificate $S' \subseteq S$. Therefore, each coordinate $i \in S_{x^\star}$ is processed at exactly once throughout the entire execution of the algorithm. (For completeness, we give a formal description of Angluin's algorithm and its analysis in Appendix A.)

[3]'s approximate certification algorithm. [3]'s algorithm, as well as ours, takes an approach that is the opposite of Angluin's, and indeed, the opposite of all local search algorithms. Instead of starting with $S_{x^\star}$ and removing irrelevant coordinates, we start with the empty set and add to it coordinates that we deem "important". The notion of *influence* from the analysis of boolean functions provides a way to quantify the importance of coordinates:

Definition 4 (Influence). *For a function $f : \{0,1\}^n \to \{0,1\}$ and a coordinate $i \in [n]$, the influence of i on f is the quantity $\mathrm{Inf}_i(f) := \Pr_{\text{uniform } \boldsymbol{x}}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})]$, where $\boldsymbol{x}^{\oplus i}$ denotes $\boldsymbol{x}$ with its i-th coordinate flipped.*

[3]'s algorithm is simple: using queries to f, determine the coordinate i with (approximately) the largest influence[2] on f; restrict the i-th coordinate of f according to $x^\star_i$ and recurse. [3] proved that for most $x^\star$'s, running this recursion to a certain depth suffices to guarantee a low-error certificate for f's value on $x^\star$, where "most" and "low-error" are both with respect to the uniform distribution.

2.1 The Three Components of Our Algorithm

The difference between our setting and [3]'s is akin to the difference between exact and uniform-distribution learning: exact learning is more challenging than distribution-independent learning, which is in turn more challenging than uniform-distribution learning. [3]'s algorithm can be seen to fail badly in the setting of zero-error certificates: there are monotone functions f with certificate complexity $k \ll n$ such that their recursion has to be run to the

[2] This is slightly imprecise, since [3] actually uses a notion of "noisy influence" which generalizes Definition 4. We do not need this generalization in this work.

maximum depth of n (corresponding to the trivial certificate $S = [n]$) in order to return a zero-error certificate.

Our algorithm is more involved than [3]'s and has three main components:

(1) *Finding a small certificate.* This component is independent of the input $x^\star$ that we seek to certify. We design an algorithm that finds an *arbitrary* $\mathrm{poly}(k)$-size certificate for a monotone f—by arbitrary, we mean that this can be a certificate for f's value on any input, not necessarily a specific one. In other words, this is a set $S \subseteq [n]$ and a bit $b \in \{0,1\}$ such that f with all the coordinates $i \in S$ restricted to b is a constant function.
(2) *Finding a small certificate for $x^\star$.* We then show how the algorithm above can be called $O(k)$ times to find a $\mathrm{poly}(k)$-size certificate for f's value on $x^\star$. The fact that $O(k)$ calls suffice follows from a basic result in query complexity, that every 1-certificate and 0-certificate of a function share at least one variable. (We defer the definitions of these terms to the body of the paper.)
(3) *Trimming the certificate.* Finally, we use Angluin's algorithm to trim the size of this certificate from $\mathrm{poly}(k)$ down to $\leq k$. Crucially, we enter this trimming process with a certificate whose size is already bounded by $\leq \mathrm{poly}(k)$, in contrast to Angluin's algorithm which starts with the certificate $S_{x^\star}$, the size of which can be as large as n. The number of queries that we require for this step is therefore only $\leq \mathrm{poly}(k)$, independent of n.

2.1.1 Killing a monotone function. We elaborate on the first component; the other two are fairly straightforward. It will be useful for us to view this as the task of "killing" a monotone function efficiently: using as few queries to f as possible, find an assignment to a small set of coordinates that *kills* f, meaning that the corresponding restriction of f is a constant function.

Our algorithm for this step is most easily understood from the perspective of *threshold phenomena* in monotone functions—this connection is the key new ingredient in our work. A wealth of techniques has been developed for the study of this topic, which is central to the theory of random graphs and percolation theory. We will only need a few of the fundamentals.

Every monotone function $f : \{0,1\}^n \to \{0,1\}$ can be associated with a function $\Phi_f : [0,1] \mapsto [0,1]$,

$$\Phi_f(p) := \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n_p}[f(\boldsymbol{x})],$$

where $\{0,1\}^n_p$ denotes the p-biased product distribution over $\{0,1\}^n$. If f is non-constant, this is a strictly increasing function of p, going from 0 to 1 as p goes from 0 to 1.

Definition 5 (Critical probability). *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant monotone function. The critical probability of f is the unique value $p(f) \in (0,1)$ for which $\Phi_f(p(f)) = \frac{1}{2}$.*

We use the critical probability of f as a proxy for how close to constant it is, i.e. how "dead" the function is. If f's critical probability is $\geq \frac{1}{2}$, our algorithm kills it to the constant-0 function by driving its critical probability towards 1; otherwise, we kill it to the constant-1 function by driving its critical probability towards 0. Our algorithm for doing so is similar in spirit to [3]'s algorithm, with the crucial difference being that ours "continually adapts" to the critical probability of f and its subfunctions:

(1) Estimate the critical probability $p(f)$ of f.
(2) Determine the coordinate i with approximately the largest $p(f)$*-biased influence* on f. The p-biased influence of a coordinate is the generalization of Definition 4 to p-biased product distributions over $\{0,1\}^n$.
(3) Recurse on the subfunction $f_{x_i=b}$, the restriction of f to $x_i = b$, where $b = 0$ if $p(f) \geq \frac{1}{2}$ and $b = 1$ otherwise.

Our analysis of this process relies on two basic results from the study of graph properties and percolation. We first use the O'Donnell–Saks–Schramm–Servedio inequality [20] to show that restricting f by the coordinate with the largest $p(f)$-biased influence changes its $p(f)$-biased expectation substantially:

$$\left| \mathop{\mathbb{E}}_{p(f)\text{-biased } \boldsymbol{x}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{p(f)\text{-biased } \boldsymbol{x}}[f_{x_i=b}(\boldsymbol{x})] \right| \geq \Omega\left(\frac{1}{k^2}\right).$$

We then show, via the Russo–Margulis lemma [16, 23], that the above implies that the critical probability of f changes substantially:

$$|p(f) - p(f_{x_i=b})| \geq \Omega\left(\frac{1}{k^3}\right). \tag{1}$$

It follows that our algorithm kills f within $O(k^3)$ recursive calls. Figure 1 on page 9 illustrates our proof strategy.

A slight optimization. The query complexity of this algorithm can be bounded by $O(k^8 \log k \log n)$. To shave off a factor of $\log k$, we consider an optimization where we estimate the critical probability of f just once, at the very beginning of the algorithm, rather than in each recursive call. Throughout the recursive process, we assume conservatively that each restriction only changes the critical probability by the minimum amount guaranteed by Equation (1). A simple adjustment of our analysis accounts for this modification (i.e. for the possibility that the true critical probability drifts away from what we assume it to be as we recurse).

3 DISCUSSION AND FUTURE WORK

Concrete directions for future work include closing the remaining gap between our upper and lower bounds of $O(k^8 \log n)$ and $\Omega(k \log n)$, as well as identifying other natural classes of functions that admit efficient certification algorithms.

More broadly, a novel aspect of our techniques is the use of concepts and results from the study of threshold phenomena: p-biased analysis, the critical probability of monotone functions, the Russo–Margulis lemma, etc. While the certification problem was the focus of this work, we speculate that there are further applications of this toolkit in learning theory, where monotonicity of the target function is a common assumption. For example, while the variance of function is often used as progress measure in learning theory, our work suggests that for monotone target functions, its critical probability could be a more useful notion. Can our idea of "continually adapting" to the critical probability be used to design new learning algorithms?

Finally, circling back to the motivation for the certification problem, we mention that there is a growing flurry of work in explainable machine learning, the vast majority of which is empirical in nature; see slide 7 of [11] for some staggering numbers. Hallmarks

of problems in this area—query access to a black box f ("post-hoc explanations"); the focus on f's values at and near a specific input $x^\star$ ("local explanations"); various notions of influence of variables ("feature attribution"); etc.—strongly suggest the potential for connections to areas of theoretical computer science such as query complexity, the analysis of boolean functions, learning theory, and sublinear algorithms. Our work fleshes out a few of these connections, but we believe that there are more near at hand.

4 PRELIMINARIES

We use **boldface** often denote random variables (e.g. $\boldsymbol{x} \sim \{0,1\}^n$) and we write "w.h.p." to mean with probability $\geq 1 - 1/\mathrm{poly}(n)$. We write $a = b \pm \varepsilon$ as shorthand for $a \in [b - \varepsilon, b + \varepsilon]$.

Boolean function complexity. In addition to certificate complexity (Definition 1), we will need a few other standard notions and facts from boolean function complexity. For an in-depth treatment (including proofs of the facts below), see [4, 10].

For a function $f : \{0,1\}^n \to \{0,1\}$ and an input $x \in \{0,1\}^n$, the *sensitivity of f at x* is the quantity

$$\mathrm{Sens}_f(x) = |\{i \in [n] : f(x) \neq f(x^{\oplus i})\}|,$$

where $x^{\oplus i}$ denotes f with its i-th coordinate flipped.

Proposition 4.1 (Sensitivity and certificate complexity). *For all functions $f : \{0,1\}^n \to \{0,1\}$ and inputs $x \in \{0,1\}^n$, we have $\mathrm{Sens}_f(x) \leq C_f(x)$.*

For a function $f : \{0,1\}^n \to \{0,1\}$, we write $D(f)$ to denote its *decision tree complexity*, the depth of the shallowest decision tree that computes f.

Fact 4.2 (Decision tree complexity and certificate complexity). *For all functions $f : \{0,1\}^n \to \{0,1\}$, we have $D(f) \leq C(f)^2$.*

We also will occasionally distinguish between 0-certificates and 1-certificates.

Definition 6 (0, 1-certificate complexity). *For a function $f : \{0,1\}^n \to \{0,1\}$ and input $x \in \{0,1\}^n$, a certificate $S \subseteq [n]$ of x is a 0-certificate if $f(x) = 0$ and 1-certificate if $f(x) = 1$. The 0-certificate complexity and 1-certificate complexity of f are defined as*

$$C_0(f) := \max_{x \in f^{-1}(0)} \{C_f(x)\} \text{ and } C_1(f) := \max_{x \in f^{-1}(1)} \{C_f(x)\}$$

respectively.

p-biased analysis. We write $\{0,1\}_p^n$ to denote the p-biased product distribution on n bit strings (that is, each bit is 1 with probability p) and $\Pr_p$ to denote the p-biased probability measure on strings. When sampling from $\{0,1\}_p^n$, we will often just write the subscript p. In particular, $\mathbb{E}_p[f]$ denotes the expectation of f with respect to $\boldsymbol{x} \sim \{0,1\}_p^n$ and similarly $\mathrm{Var}_p[f] = \mathbb{E}_p[f^2] - \mathbb{E}_p[f]^2 = \mathbb{E}_p[f](1 - \mathbb{E}_p[f])$ is the p-biased variance of f.

We'll use two common notions of influence.

Definition 7 (p-biased flip influence; generalization of Definition 4). *Let $f : \{0,1\}^n \to \{0,1\}$ be a function, $p \in [0,1]$, and $i \in [n]$. The p-biased flip influence of i on f is the quantity:*

$$\mathrm{Inf}_{i,p}^{\oplus}[f] := \Pr_p[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})].$$

Definition 8 (p-biased rerandomized influence). *Let $f : \{0,1\}^n \to \{0,1\}$ be a function, $p \in [0,1]$, and $i \in [n]$. The p-biased rerandomized influence of i on f is the quantity:*

$$\mathrm{Inf}_{i,p}^{\sim}[f] := 2\Pr_p[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\sim i})]$$

where $\boldsymbol{x}^{\sim i}$ is the string $\boldsymbol{x}$ with its i-th coordinate rerandomized according to $\{0,1\}_p$.

For each notion of influence, the *total* influence is the sum of the influences of all the coordinates. We write $\mathrm{Inf}_p^{\oplus}[f]$ and $\mathrm{Inf}_p^{\sim}[f]$ for the total flip and rerandomized influence, respectively.

We record a few basic properties of p-biased influence. For a proof of these properties, see the appendix of the full version of the paper.

Proposition 4.3. *For any boolean function $f : \{0,1\}^n \to \{0,1\}$ and $i \in [n]$,*

1. $\mathrm{Inf}_p^{\oplus}[f] = \mathbb{E}_p[\mathrm{Sens}_f(\boldsymbol{x})]$.
2. $\mathrm{Inf}_{i,p}^{\oplus}[f] = \Pr_p[f_{x_i=1}(\boldsymbol{x}) \neq f_{x_i=0}(\boldsymbol{x})]$.
3. $\mathrm{Inf}_{i,p}^{\sim}[f] = 4p(1-p)\mathrm{Inf}_{i,p}^{\oplus}[f]$.
4. $\mathrm{Inf}_p^{\sim}[f] \geq \mathrm{Var}_p[f]$.

If f is monotone,

5. $\mathbb{E}_p[f] = \mathbb{E}_p[f_{x_i=0}] + p \cdot \mathrm{Inf}_{i,p}^{\oplus}[f] = \mathbb{E}_p[f_{x_i=1}] - (1-p) \cdot \mathrm{Inf}_{i,p}^{\oplus}[f]$.

5 FIRST COMPONENT OF THEOREM 1: FINDING AN ARBITRARY CERTIFICATE

In this section, we show how to find an *arbitrary* size-poly(k) certificate of a monotone function in $O(k^7 \log n)$ queries where k is the certificate complexity of the function. We first state the algorithm below then show each step can be implemented in a query efficient manner and with high probability of success. In particular, we'll give a $O(k^7 \log k \log n)$ query upper bound and then we'll show how a simple modification of the algorithm can obtain a $O(k^7 \log n)$ upper bound.

Algorithm 1 Finding a certificate of a monotone function

Given: Query access to a monotone function $f : \{0,1\}^n \to \{0,1\}$ and parameter k.
Initialize $S \leftarrow \emptyset$
while f is nonconstant **do**
  Find an ε-approximate critical probability p of f, where $\varepsilon = O(1/k^3)$
  Estimate $\mathrm{Inf}_{i,p}^{\sim}[f]$ to additive accuracy $\pm O(1/k^2)$ for all i
  Add coordinate i to S where $\mathrm{Inf}_{i,p}^{\sim}[f]$ is the largest influence estimate
  $f \leftarrow f_{x_i=b}$ where $b = 0$ if $p \geq 1/2$ and 1 otherwise
end while
return the certificate S

Theorem 3. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \leq k$. There is an implementation of Algorithm 1 that w.h.p. makes $O(k^7 \log k \log n)$ queries to f and returns a certificate of size $O(k^3)$.*

5.1 Structural Properties of Φ_f

As discussed in Section 2, the function $\Phi_f : [0,1] \to [0,1]$,

$$\Phi_f(p) \coloneqq \mathbb{E}_p[f(\boldsymbol{x})]$$

will be central to our analysis. In this section we record and establish a few structural properties of Φ_f that will be useful for the proof of Theorem 3.

The first is the Russo–Margulis lemma [16, 23] which states that the derivative of $\Phi_f(p)$ is exactly the total flip influence of f under the p-biased distribution.

Lemma 5.1 (Russo–Margulis). *Let f be a monotone function, then*

$$\frac{d}{dp}\Phi_f(p) = \mathrm{Inf}^{\oplus}_p[f].$$

For a Fourier-analytic proof of the Russo–Margulis lemma, see [19]. For the sake of completeness, we give a self-contained combinatorial proof in the appendix of the full version.

We leverage three important corollaries of the Russo–Margulis lemma in our analysis. Applying the lemma twice, to $\Phi_f(p)$ and $\mathrm{Inf}^{\oplus}_{i,p}[f]$, we can upper bound the Lipschitz constants of these quantities by k when viewed as functions of p. We then apply it again to lower bound the derivative of $\Phi_f(p)$ near the critical probability $p(f)$ of f, to show that that any p for which $\Phi_f(p)$ is close to $1/2$ must be close to $p(f)$.

Corollary 5.2 (Lipschitz constant of Φ_f). *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \le k$, then for all $q \neq r$ we have*

$$\frac{\Phi_f(q) - \Phi_f(r)}{q - r} \le k.$$

Proof. By the mean value theorem, the slope of the tangent line $(\Phi_f(q) - \Phi_f(r))/(q-r)$ is the derivative of $\Phi_f(p)$ at some point $\hat{p}$ in between q and r. Applying the Russo–Margulis lemma, we have that

$$\frac{\Phi_f(q) - \Phi_f(r)}{q - r} = \frac{d}{dp}\Phi_f(p)\bigg|_{p=\hat{p}} = \mathrm{Inf}^{\oplus}_{\hat{p}}[f].$$

By Propositions 4.3.1 and 4.1,

$$\mathrm{Inf}^{\oplus}_{\hat{p}}[f] = \mathbb{E}_{\hat{p}}[\mathrm{Sens}_f(\boldsymbol{x})] \le \mathbb{E}_{\hat{p}}[C_f(\boldsymbol{x})] \le C(f)$$

and the proof is complete. □

Corollary 5.3 (Lipschitz constant of $\mathrm{Inf}^{\oplus}_{i,p}$). *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \le k$. Then for all $q \neq r$ and $i \in [n]$ we have*

$$\left|\frac{\mathrm{Inf}^{\oplus}_{i,q}[f] - \mathrm{Inf}^{\oplus}_{i,r}[f]}{q - r}\right| \le k.$$

Proof. When f is monotone, Proposition 4.3.2 can be written as $\Pr_p[f_{x_i=1}(\boldsymbol{x}) \neq f_{x_i=0}(\boldsymbol{x})] = \Phi_{f_{x_i=1}}(p) - \Phi_{f_{x_i=0}}(p)$. Hence,

$$\begin{aligned}\frac{d}{dp}\mathrm{Inf}^{\oplus}_{\hat{p}}[f] &= \frac{d}{dp}\left[\Phi_{f_{x_i=1}}(p) - \Phi_{f_{x_i=0}}(p)\right] \\ &= \mathrm{Inf}^{\oplus}_p[f_{x_i=1}] - \mathrm{Inf}^{\oplus}_p[f_{x_i=0}]\end{aligned}$$

by the Russo–Margulis lemma. Since $0 \le \mathrm{Inf}^{\oplus}_p[f_{x_i=b}] \le C(f_{x_i=b}) \le C(f)$ for $b \in \{0,1\}$, the result then follows from the application of the mean value theorem as in the proof of Corollary 5.2. □

Corollary 5.4. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function and let $p \in [0,1]$ be any point satisfying $\Phi_f(p) = 1/2 \pm \varepsilon$. Then*

$$p = p(f) \pm \frac{4\varepsilon}{1 - 4\varepsilon^2}.$$

Proof. Suppose without loss of generality that $p \le p(f)$ (the case where $p > p(f)$ is symmetric). Again applying the mean value theorem, there is some $\hat{p} \in [p, p(f)]$ satisfying $\mathrm{Inf}^{\oplus}_{\hat{p}}[f] = (\Phi_f(p(f)) - \Phi_f(p))/(p(f) - p)$. Then, we have

$$\begin{aligned}\frac{\varepsilon}{p(f) - p} &\ge \frac{\Phi_f(p(f)) - \Phi_f(p)}{p(f) - p} \\ &= \mathrm{Inf}^{\oplus}_{\hat{p}}[f] \ge \mathrm{Var}_{\hat{p}}[f] && \text{(Proposition 4.3.4)} \\ &\ge \mathrm{Var}_p[f] = \Phi_f(p)(1 - \Phi_f(p)) && \text{(monotonicity)} \\ &\ge \left(\frac{1}{2} + \varepsilon\right)\left(\frac{1}{2} - \varepsilon\right) = \frac{1}{4} - \varepsilon^2\end{aligned}$$

which gives the desired inequality. □

The next lemma quantifies the change in the critical probability of f when we restrict one of its coordinates. In particular, we use the Lipschitz constant for $\Phi_f(p)$ to show this change is large when the restricted coordinate is influential.

Lemma 5.5. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \le k$. Then for all $i \in [n]$, we have*

$$p(f_{x_i=0}) - p(f) \ge \frac{p(f) \cdot \mathrm{Inf}^{\oplus}_{i,p(f)}[f]}{k}$$

and analogously,

$$p(f) - p(f_{x_i=1}) \ge \frac{(1 - p(f)) \cdot \mathrm{Inf}^{\oplus}_{i,p}[f]}{k}.$$

Proof. We prove the lower bound on $p(f_{x_i=0}) - p(f)$. The proof for $p(f) - p(f_{x_i=1})$ is symmetric. First, rewriting Proposition 4.3.5 in the Φ_f notation we have

$$\Phi_{f_{x_i=0}}(p) = \Phi_f(p) - p \cdot \mathrm{Inf}^{\oplus}_{i,p}[f]. \tag{2}$$

Therefore,

$$\begin{aligned}k &\ge \frac{\Phi_{f_{x_i=0}}(p(f_{x_i=0})) - \Phi_{f_{x_i=0}}(p(f))}{p(f_{x_i=0}) - p(f)} && \text{(Corollary 5.2)} \\ &= \frac{\Phi_{f_{x_i=0}}(p(f_{x_i=0})) - \left(\Phi_f(p(f)) - p(f) \cdot \mathrm{Inf}^{\oplus}_{i,p(f)}[f]\right)}{p(f_{x_i=0}) - p(f)} && \text{(Equation (2))} \\ &= \frac{p(f) \cdot \mathrm{Inf}^{\oplus}_{i,p(f)}[f]}{p(f_{x_i=0}) - p(f)}\end{aligned}$$

which completes the proof. □

Finally, we need an inequality of O'Donnell, Saks, Schramm, and Servedio [20] which says that f has an influential p-biased coordinate when the p-biased variance of f is large.

Theorem 4 (OSSS inequality). *For all functions $f : \{0,1\}^n \to \{0,1\}$ and $p \in [0,1]$,*

$$\max_{i \in [n]} \left\{\mathrm{Inf}^{\sim}_{i,p}[f]\right\} \ge \frac{\mathrm{Var}_p[f]}{D(f)},$$

where $D(f)$ denotes the decision tree complexity of f.

5.2 Algorithmic Lemmas

We will need a few lemmas to bound the query complexity of Algorithm 1. First we show that we can find an approximation of the critical probability of f by finding a value p for which $\Phi_f(p)$ is close to $1/2$. Next we show that we can efficiently estimate *rerandomized* influence to an additive accuracy. Finally, we show that if all of the influences are estimated under the p-biased distribution for p close to $p(f)$, the critical probability of f, then the most influential coordinate under the p-biased distribution must also be influential under the $p(f)$-biased distribution.

Lemma 5.6 (Finding an approximate expectation of f). *Given queries to a monotone $f : \{0,1\}^n \to \{0,1\}$ with $C(f) \le k$, for any $\varepsilon > 0$ we can find some $p \in [0,1]$ satisfying $\Phi_f(p) = 1/2 \pm \varepsilon$ w.h.p. using $O(\log(k/\varepsilon)\log(n)/\varepsilon^2)$ many queries.*

Proof. Since Φ_f has Lipschitz constant $\le k$ (Corollary 5.2), any value $\hat{p}$ that is within an additive $\pm\varepsilon/3k$ of the true critical probability $p(f)$ of f is an $\varepsilon/3$-critical probability of f. That is,

$$\hat{p} = p(f) \pm \frac{\varepsilon}{3k} \quad \Longrightarrow \quad \Phi_f(\hat{p}) = \frac{1}{2} \pm \frac{\varepsilon}{3}.$$

We split the $[0,1]$ into $3k/\varepsilon$ intervals each of length $\varepsilon/3k$. As observed above, the interval containing the critical probability will satisfy $\Phi_f(\hat{p}) = \frac{1}{2} \pm \frac{\varepsilon}{3}$ for all $\hat{p}$ in that interval. By the Chernoff bound, for any value $p \in [0,1]$ we can estimate $\Phi_f(p) = \mathbb{E}_p[f]$ to accuracy $\pm\varepsilon/3$ and confidence $1-\delta$ using $O(\log(1/\delta)/\varepsilon^2)$ queries.

Performing binary search over the $3k/\varepsilon$ intervals, with $O(\log(k/\varepsilon))$ estimations of $\Phi_f(p)$ we are guaranteed to find a $\hat{p}$ such that our estimate of $\Phi_f(\hat{p})$ is $\frac{1}{2} \pm \frac{\varepsilon}{3} \pm \frac{\varepsilon}{3} = \frac{1}{2} \pm \frac{2\varepsilon}{3}$; this implies that its true value is $\Phi_f(\hat{p}) = \frac{1}{2} \pm \frac{2\varepsilon}{3} \pm \frac{\varepsilon}{3} = \frac{1}{2} \pm \varepsilon$, i.e. $\hat{p}$ is indeed an ε-approximate critical probability. Choosing $\delta = 1/\mathrm{poly}(n)$ and noting that this is small enough to union bound over the $O(\log(k/\varepsilon))$ many estimations (with much room to spare), we get that the overall query complexity is

$$O(\log(k/\varepsilon)) \cdot O(\log(n)/\varepsilon^2) = O(\log(k/\varepsilon)\log(n)/\varepsilon^2). \qquad \square$$

Lemma 5.7 (Finding an approximate critical probability). *Given queries to a monotone $f : \{0,1\}^n \to \{0,1\}$ with $C(f) \le k$ for any $0 < \varepsilon < 1$, we can find $p \in [0,1]$ satisfying $p = p(f) \pm \varepsilon$ w.h.p. using $O(\log(k/\varepsilon)\log(n)/\varepsilon^2)$ queries.*

Proof. We show that any $p \in [0,1]$ satisfying $\Phi_f(p) = 1/2 \pm \varepsilon/8$ satisfies the constraints of the lemma statement. The result then follows from Lemma 5.6 which says that we can compute such a p w.h.p. using $O(\log(k/\varepsilon)\log n/\varepsilon^2)$ queries.

Let $p \in [0,1]$ satisfy $\mathbb{E}_p[f] = 1/2 \pm \varepsilon/8$. Then we have

$$\begin{aligned} p &= p(f) \pm \frac{4(\varepsilon/8)}{1 - 4(\varepsilon/8)^2} && \text{(Corollary 5.4)} \\ &= p(f) \pm \frac{\varepsilon}{2 - \varepsilon^2/8} \\ &= p(f) \pm \varepsilon. && (\varepsilon^2/8 < 1) \end{aligned}$$

$\square$

Lemma 5.8 (Estimating influences). *Given queries to a monotone $f : \{0,1\}^n \to \{0,1\}$, some $p \in [0,1]$, and $\varepsilon > 0$, we can approximate $\widetilde{\mathrm{Inf}}_{i,p}[f]$ to accuracy $\pm\varepsilon$ for all $i \in [n]$ w.h.p. using $O(\log n/\varepsilon^2)$ many queries.*

Proof. Rewriting Proposition 4.3.5 using $\widetilde{\mathrm{Inf}}_{i,p}[f] = 4p(1-p)\mathrm{Inf}^{\oplus}_{i,p}[f]$ we have

$$\begin{aligned} \widetilde{\mathrm{Inf}}_{i,p}[f] &= 4(1-p)\left(\mathbb{E}_p[f] - \mathbb{E}_p[f_{x_i=0}]\right) \\ &= 4p\left(\mathbb{E}_p[f_{x_i=1}] - \mathbb{E}_p[f]\right) \end{aligned} \tag{3}$$

We show with a single random sample $S \subseteq \{0,1\}^n$ of size $O(\log n/\varepsilon^2)$ we can estimate $\widetilde{\mathrm{Inf}}_{i,p}[f]$ to accuracy ε for all $i \in [n]$ by estimating $\mathbb{E}_p[f]$ and either $\mathbb{E}_p[f_{x_i=1}]$ or $\mathbb{E}_p[f_{x_i=0}]$. We write $\overline{\mathbb{E}}_S[f]$ for the p-biased expectation of f estimated from the set S. For each $i \in [n]$ and $b \in \{0,1\}$, we define $S_b = \{x^{-i} \in \{0,1\}^{n-1} : x \in S \text{ and } x_i = b\}$ where x^{-i} denotes the string x with the ith coordinate removed. Since $|S| = |S_1| + |S_0|$ we must have $|S_b| \ge |S|/2$ for some $b \in \{0,1\}$. We then estimate $\mathbb{E}_p[f_{x_i=b}]$ for this value of b and use the appropriate identity from eq. (3) to estimate the ith influence. Note that we can perform this estimate of $\mathbb{E}_p[f_{x_i=b}]$ because the strings in S_b are distributed according to $\{0,1\}^{n-1}_p$ and we already know the values of $f_{x_i=b}$ for all strings in S_b (since the query values of f on S are known). Thus by a Chernoff bound we can estimate both $\mathbb{E}_p[f_{x_i=b}]$ and $\mathbb{E}_p[f]$ to accuracy $\pm\varepsilon/8$ and confidence $1-\delta$ using $O(\log(1/\delta)/\varepsilon^2)$ random samples. These estimates then ensure that our estimate of $\widetilde{\mathrm{Inf}}_{i,p}[f]$ has accuracy $\pm\varepsilon$. For example, if $b = 0$, our estimates $\overline{\mathbb{E}}_S[f]$ and $\overline{\mathbb{E}}_{S_0}[f_{x_i=0}]$ satisfy

$$\begin{aligned} \overline{\widetilde{\mathrm{Inf}}}_{i,p}[f] &= 4(1-p)\left(\overline{\mathbb{E}}_S[f] - \overline{\mathbb{E}}_{S_0}[f_{x_i=0}]\right) \\ &= 4(1-p)\Big(\left(\mathbb{E}_p[f] \pm \varepsilon/8\right) - \left(\mathbb{E}_p[f_{x_i=0}] \pm \varepsilon/8\right)\Big) \\ &= \widetilde{\mathrm{Inf}}_{i,p}[f] \pm (1-p)\varepsilon = \widetilde{\mathrm{Inf}}_{i,p}[f] \pm \varepsilon \end{aligned}$$

where $\overline{\widetilde{\mathrm{Inf}}}_{i,p}[f]$ denotes the influence estimate. We choose $\delta = 1/\mathrm{poly}(n)$ small enough to union bound over all $i \in [n]$ which makes the total number of random samples/queries $O(\log n/\varepsilon^2)$ as desired. $\square$

Lemma 5.9. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \le k$. Let $\overline{p} = p(f) \pm \varepsilon$ for some $0 < \varepsilon < 1/k^2$ and suppose $\overline{\widetilde{\mathrm{Inf}}}_{i,\overline{p}}[f] = \widetilde{\mathrm{Inf}}_{i,\overline{p}}[f] \pm k\varepsilon$ for all $i \in [n]$. Then*

$$\mathrm{Inf}^{\oplus}_{i,p(f)}[f] \ge \frac{1}{8k^2} - 3k\varepsilon$$

where $i = \arg\max_{i \in [n]} \overline{\widetilde{\mathrm{Inf}}}_{i,\overline{p}}[f]$.

Proof. Recall that $\mathrm{Var}_p[f] = \Phi_f(p)(1-\Phi_f(p))$ and for our estimate $\overline{p} = p(f) \pm \varepsilon$ we have $\Phi_f(\overline{p}) = 1/2 \pm k\varepsilon$ since Φ_f has Lipschitz constant $\le k$ (Corollary 5.2). Thus by monotonicity $\mathrm{Var}_{\overline{p}}[f] \ge (1/2 - k\varepsilon)(1/2 + k\varepsilon) \ge 1/8$ (using the assumption that $\varepsilon < 1/k^2$). The OSSS inequality, Theorem 4, then states

$$\max_{i \in [n]} \left\{\widetilde{\mathrm{Inf}}_{i,\overline{p}}[f]\right\} \ge \frac{\mathrm{Var}_{\overline{p}}[f]}{D(f)} \ge \frac{1}{8D(f)}.$$

Furthermore, we can lower bound $1/8D(f) \geq 1/8k^2$ using Fact 4.2. Since our estimate $\overline{\mathrm{Inf}}^{\sim}_{i,\overline{p}}[f]$ has accuracy $\pm k\varepsilon$ the maximum influence estimate satisfies

$$\max_i \overline{\mathrm{Inf}}^{\sim}_{i,\overline{p}}[f] \geq \max_i \mathrm{Inf}^{\sim}_{i,\overline{p}}[f] - k\varepsilon \geq \frac{1}{8k^2} - k\varepsilon.$$

Hence, the true influence at this maximal i satisfies $\mathrm{Inf}^{\sim}_{i,\overline{p}}[f] \geq (1/8k^2 - k\varepsilon) - k\varepsilon = 1/8k^2 - 2k\varepsilon$. Finally, to translate this bound to a lower bound on $\mathrm{Inf}^{\oplus}_{i,p(f)}[f]$ we switch to flip influence and apply our Lipschitz bound on $\mathrm{Inf}^{\oplus}_{i,p}$. In other words,

$$\begin{aligned}
\mathrm{Inf}^{\oplus}_{i,p(f)}[f] &\geq \mathrm{Inf}^{\oplus}_{i,\overline{p}}[f] - |p(f) - \overline{p}| \cdot k && \text{(Corollary 5.3)} \\
&\geq \mathrm{Inf}^{\oplus}_{i,\overline{p}}[f] - k\varepsilon \\
&\geq \mathrm{Inf}^{\sim}_{i,\overline{p}}[f] - k\varepsilon \\
&\geq \frac{1}{8k^2} - 3k\varepsilon. && \square
\end{aligned}$$

5.3 Proof of Theorem 3

For our proof, we first show that accurate estimates of the critical probability of f and the influences will ensure quick progress towards termination. Then we analyze the query complexity required to estimate these quantities to the specified accuracy with high confidence. This proof can be read in conjunction with Figure 1 which illustrates the main idea.

Proof of correctness. Our measure of progress is the critical probability of f. At a high level we show that if we find an $O(1/k^3)$-approximate critical probability and estimate influences to accuracy $O(1/k^2)$ at each step of the algorithm, then the critical probability of f is guaranteed to increase or decrease by $\Omega(1/k^3)$. Since the function is constant when the critical probability is 0 or 1, we know that the algorithm must terminate after $O(k^3)$ steps.

To be more specific, let f be a nonconstant function obtained at some point in the algorithm with $C(f) \leq k$. Let $0 < \varepsilon < 1/k^2$ be arbitrary and let $\overline{p} = p(f) \pm \varepsilon$ be an approximate critical probability and suppose each $\mathrm{Inf}^{\sim}_{i,p}[f]$ is estimated to accuracy $\pm k\varepsilon$. Then, we can write

$$\begin{aligned}
p(f_{x_i=0}) - p(f) &\geq \frac{p(f) \cdot \mathrm{Inf}^{\oplus}_{i,p(f)}[f]}{k} && \text{(Lemma 5.5)} \\
&\geq p(f) \cdot \left(\frac{1}{8k^3} - 3\varepsilon\right) && \text{(Lemma 5.9)}
\end{aligned}$$

and likewise

$$p(f) - p(f_{x_i=1}) \geq (1 - p(f)) \cdot \left(\frac{1}{8k^3} - 3\varepsilon\right).$$

In the final step of the algorithm's loop f is restricted to $x_i = 0$ if $\overline{p} \geq 1/2$ in which case we have $p(f) \geq 1/2 - \varepsilon$ and thus $p(f_{x_i=0}) - p(f) \geq (1/2 - \varepsilon)(1/8k^3 - 3\varepsilon)$. Note importantly that if $\overline{p} \geq 1/2$ the next estimate will also be greater than $1/2$ and so on, ensuring that the final certificate will be a 0-certificate. We can then choose $\varepsilon = O(1/k^3)$ small enough to ensure $p(f_{x_i=0}) - p(f) \geq \Omega(1/k^3)$ and likewise in the case that $\overline{p} < 1/2$.

In both cases, one step of the main loop makes at least $\Omega(1/k^3)$ progress towards termination and so the loop iterates $O(k^3)$ times. Hence, the final certificate has at most $O(k^3)$ coordinates since each iteration of the loop adds one coordinate.

Query complexity. Lemma 5.6 shows we can compute a $O(1/k^3)$-approximate critical probability using $O(k^4 \log k \log n)$ queries. Moreover, computing a $O(1/k^2)$-approximation of influence requires $O(k^4 \log n)$ queries by Lemma 5.8. Note also that we can test whether f is constant with ≤ 2 queries using monotonicity (f is constant if and only if $f(0^n) = f(1^n)$). Thus, one iteration of the main loop makes $O(k^4 \log k \log n)$ queries to f. Since the main loop executes $O(k^3)$ times, the total number of queries is at most $O(k^7 \log k \log n)$.

5.4 Reducing the Query Complexity via Fewer Critical Probability Estimates

We can reduce the query complexity of Algorithm 1 by a $\log k$ factor if we instead estimate the critical probability of f once at the beginning of the algorithm then deterministically update it by the error term we calculated as ε in the proof above. At a high level, the idea is that the analysis for Theorem 3 shows that restricting f by an influential coordinate will shift its critical probability by at least $\Omega(1/k^3)$. Hence, in the worst case, the algorithm makes the smallest amount of progress, approximately $1/k^3$, in each step. We can thus manually shift our critical probability estimate after each iteration by the minimal amount of progress we expect instead of using additional queries to f to determine the new critical probability.

In the lemma below we assume that the critical probability of f is initially $\geq 1/2 - \varepsilon$, and hence f is simplified by repeatedly restricting 0-coordinates. The proof shows these restrictions force its critical probability to approach 1. The alternate case where the initial critical probability is less than $1/2 + \varepsilon$ is analogous. In this case, one can show via symmetric arguments that the estimate $\overline{p}_t = 1/2 - (t-1)\varepsilon$ satisfies $p(f_t) \leq \overline{p}_t$ for all t.

Lemma 5.10. *Fix an error term $0 < \varepsilon \leq 1/40k^3$ and suppose $p(f) \geq 1/2 - \varepsilon$. Consider a variant of Algorithm 1 where at the t^{th} step we estimate the critical probability as $\overline{p}_t = 1/2 + (t-1)\varepsilon$ and we always set $f \leftarrow f_{x_i=0}$. Let $f_t : \{0,1\}^{n-t} \to \{0,1\}$ denote the function at the t^{th} step. Then $p(f_t) \geq \overline{p}_t$ for all t for which f_t is nonconstant.*

Proof. The proof is by induction on t. The statement holds for $t = 0$ by assumption. Otherwise assume that $p(f_t) \geq \overline{p}_t$. Then we show $p(f_{t+1}) \geq \overline{p}_{t+1}$. If $p(f_t) > \overline{p}_{t+1}$ then there's nothing left to show since $\Phi_{f_t}(p(f_t)) \leq \Phi_{f_{t+1}}(p(f_{t+1}))$ always holds by Proposition 4.3.5 and hence $p(f_t) \leq p(f_{t+1})$. Otherwise, assume $p(f_t) \leq \overline{p}_{t+1}$. In particular, $\overline{p}_t \leq p(f_t) \leq \overline{p}_{t+1} = \overline{p}_t + \varepsilon$ which shows that $\overline{p}_t = p(f_t) \pm \varepsilon$. The influence estimates have accuracy $\pm k\varepsilon$ which then allows us to apply Lemmas 5.5 and 5.9 as in the proof of Theorem 3 above, to conclude

$$p(f_{t+1}) - p(f_t) \geq p(f_t) \cdot \left(\frac{1}{8k^3} - 3\varepsilon\right) \geq \left(\frac{1}{2} - \varepsilon\right)\left(\frac{1}{8k^3} - 3\varepsilon\right).$$

Choosing $\varepsilon \leq 1/40k^3$ then ensures $p(f_{t+1}) - p(f_t) \geq \varepsilon$ which completes the induction since $p(f_t) + \varepsilon \geq \overline{p}_t + \varepsilon = \overline{p}_{t+1}$. $\square$

Equipped with Lemma 5.10, we can give a slight improvement on the query complexity of Theorem 3.

Theorem 5. *Given a monotone function $f : \{0,1\}^n \to \{0,1\}$ with $C(f) \leq k$, there is an algorithm which w.h.p. returns a certificate of size $O(k^3)$ and makes $O(k^7 \log n)$ queries to f.*

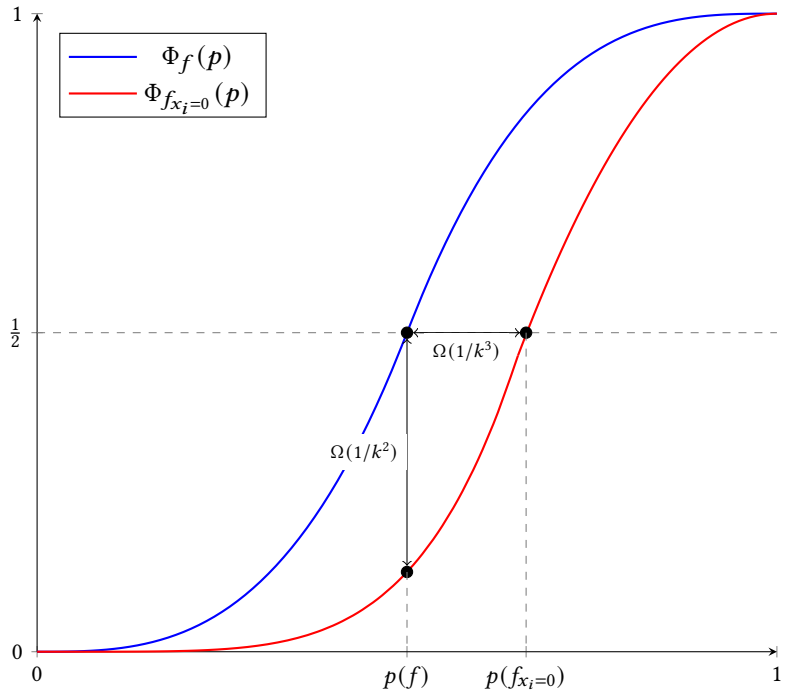

Figure 1: Illustration of the key atomic step in the proof of Theorem 3. Let $p(f)$ denote the critical probability of f. The OSSS inequality implies the existence of a coordinate $i \in [n]$ such that $\Phi_f(p(f)) - \Phi_{f_{x_i=0}}(p(f)) \geq \Omega(k^{-2})$, and we bound, using the Russo–Margulis lemma, the Lipschitz constant of $\Phi_{f_{x_i=0}}$ by $\leq k$. We therefore conclude that the critical probabilities of f and $f_{x_i=0}$ differ by $\Omega(k^{-3})$.

Proof. We modify Algorithm 1 to estimate the critical probability of f once at the start and then increment/decrement it by $\varepsilon = 1/(40k^3)$ after each iteration. Then the algorithm terminates after at most $O(k^3)$ iterations of the main loop by Lemma 5.10. We use Lemma 5.7 to estimate the critical probability of f initially which requires $O(k^6 \log k \log n)$ queries for our choice of ε. Since this estimate $\overline{p}$ satisfies $\overline{p} = p(f) \pm \varepsilon$ if $\overline{p} \geq 1/2$ then $p(f) \geq 1/2 - \varepsilon$ ensures the desired precondition for Lemma 5.10 and otherwise $p(f) \leq 1/2 + \varepsilon$ and the symmetric case applies.

Each step of the algorithm's loop requires $O(k^4 \log n)$ queries to estimate the influences to accuracy $k\varepsilon = 1/40k^2$ by Lemma 5.8. Hence the algorithm makes $O(k^7 \log n)$ queries overall. □

6 COMPLETING THE PROOF OF THEOREM 1

In this section we show how to find a certificate for a given input using Algorithm 1 as a subroutine. The algorithm itself is fairly straightforward. For a monotone function f and an input $x^\star$, we find an arbitrary certificate of f using Algorithm 1 and then restrict f on the coordinates in the certificate to the values specified by $x^\star$. Then we recurse on the subfunction and repeat until the function is constant.

We prove the following guarantee on Algorithm 2.

Theorem 6. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \leq k$, then Algorithm 2 iterates $O(k)$ times and w.h.p. outputs a certificate of size $O(k^4)$.*

Algorithm 2 Finding a certificate for a given input

Given: A monotone function $f : \{0,1\}^n \to \{0,1\}$ and input $x^\star$.

```
Initialize S ← ∅
while f is nonconstant do
    s ← the output of Algorithm 1 on f
    S ← S ∪ s        ▷ Update certificate S with coordinates from s
    f ← f_{x_i = x*_i, i∈s}        ▷ restrict f according to s
end while
return the certificate S.
```

Combining this theorem with Theorem 5, we get that a certificate for an input to a monotone function can be found using at most $O(k^8 \log n)$ queries to f.

Corollary 6.1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function with $C(f) \leq k$. Then, a certificate of size $O(k^4)$ can be computed w.h.p. for any input $x^\star$ using $O(k^8 \log n)$ queries to f.*

The progress measure in our analysis of Algorithm 2 is $C_0(f) + C_1(f)$, the sum of the 0-certificate complexity and 1-certificate complexity of f. In particular, each iteration of the main loop is guaranteed to decrease this quantity by at least 1 which gives an upper bound on $2C(f)$ on the total number of iterations. For the proof, we use the fact that, for any Boolean function, every 0-certificate must intersect every 1-certificate (since otherwise there would be one input string having both a 0-certificate and a 1-certificate).

Fact 6.2. *Let S_0 be a 0-certificate for a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and let S_1 be a 1-certificate. Then $S_0 \cap S_1 \neq \emptyset$.*

Proof of Theorem 6. Let f be a nonconstant function during the execution of the algorithm. We'll show that $C_0(f) + C_1(f)$ decreases by at least 1 after each iteration of the main loop. Let s denote the certificate that Algorithm 1 returns and suppose without loss of generality that s is a 1-certificate (the argument is symmetric for a 0-certificate). Then we'll show that $C_0(f_s) \leq C_0(f) - 1$ where f_s is the restriction according to s and $x^\star$: $f_s = f_{x_i = x_i^\star, i \in s}$. Consider any $x \in f_s^{-1}(0)$. Let $x' \in \{0,1\}^n$ be the string formed by inserting $x^\star|_s$ into the string x so that $f(x') = f_s(x)$ and $x'|_s = x^\star|_s$. Let s_0 be a 0-certificate of f on x' with $|s_0| \leq C_0(f)$. Then $s_0 \setminus s$ is a 0-certificate of f_s on x. We can bound the size of this 0-certificate:

$$\begin{aligned} |s_0 \setminus s| &\leq |s_0| - 1 && (s_0 \cap s \neq \emptyset \text{ by Fact 6.2}) \\ &\leq C_0(f) - 1. \end{aligned}$$

Since x is any arbitrary 0-input to f_s, we have that $C_0(f_s) \leq C_0(f) - 1$ as desired.

Since f must be constant when either $C_0(f)$ or $C_1(f)$ is 0, the algorithm must terminate after at most $C_0(f) + C_1(f) \leq 2C(f)$ iterations. Each iteration adds at most $O(k^3)$ coordinates to the certificate S and hence $|S|$ is $O(k^4)$ at the end of the algorithm. □

6.1 Trimming the Certificate Using Angluin's Algorithm

Algorithm 2 returns a certificate of size $O(k^4)$. In this section, we show how to reduce that certificate to size $\leq k$ using $O(k^4)$ additional queries.

Claim 6.3. *Let S be a certificate for an input $x^\star$ of a monotone function $f : \{0,1\}^n \to \{0,1\}$. If $|S| > C(f)$ then a certificate $S' \subseteq S$ with $|S'| \leq C(f)$ can be computed from S using $O(|S|)$ queries to f.*

The proof of this claim is implicit in [1, Theorem 1]. We give a self-contained exposition of the proof adapted to our setting in Appendix A.

We apply Claim 6.3 as a postprocessing step after executing Algorithm 2. Since this postprocessing step only requires an additional $O(k^4)$ queries to f the overall number of queries is still upper bounded by $O(k^8 \log n)$, the query bound on Algorithm 2. Thus, the combination of Corollary 6.1 with Claim 6.3 establishes Theorem 1.

7 LOWER BOUNDS: PROOFS OF CLAIM 1.1 AND CLAIM 1.2

Our lower bounds in this section and the next will rely on the easy direction of Yao's lemma:

Lemma 7.1 ([29]). *For any $q \in \mathbb{N}$, let $\mathcal{R}_q$ and $\mathcal{D}_q$ be the set of all q-query randomized and deterministic algorithms respectively, and let I be the set of all possible pairs $f : \{0,1\}^n \to \{0,1\}$ and $x^\star \in \{0,1\}^n$ (i.e. instances of the certification problem).*

For any distribution μ supported on I,

$$\min_{R \in \mathcal{R}_q} \max_{(f,x^\star) \in I} [\mathrm{error}_R(f, x^\star)] \geq \min_{D \in \mathcal{D}_q} \mathop{\mathbb{E}}_{(f,\boldsymbol{x}^\star) \sim \mu} [\mathrm{error}_D(\boldsymbol{f}, \boldsymbol{x}^\star)]$$

where $\mathrm{error}_R(f, x^\star)$ is the probability that R does not successfully return a certificate for f's value on $x^\star$, and $\mathrm{error}_D(f, x^\star) = \mathbb{1}[D \text{ does not successfully return a certificate for } f\text{'s value on } x^\star]$.

7.1 Proof of Claim 1.1

Claim 1.1 is a special case of the following claim:

Claim 7.2. *Let $n, q, \ell \in \mathbb{N}$ and $\mathcal{A}$ be a q-query randomized local search algorithm. There is a monotone $f : \{0,1\}^n \to \{0,1\}$ with $C(f) = 1$ and input $x^\star \in \{0,1\}^n$ on which $\mathcal{A}$ successfully returns a size-ℓ certificate for x with probability $\leq (\ell + q - 1)/n$.*

We use Yao's lemma with the distribution μ where:

(1) $\boldsymbol{x}$ is a constant, supported entirely on $x^\star = [1, \ldots, 1]$, and
(2) $\boldsymbol{f}$ is a random dictator: we select $\boldsymbol{i} \in [n]$ uniformly at random and set $\boldsymbol{f}(x) = x_{\boldsymbol{i}}$.

We will assume that $\mathcal{A}$ is deterministic and prove that the probability, over the randomness of $\boldsymbol{f}$, that $\mathcal{A}$ successfully finds a size-ℓ certificate $\boldsymbol{f}$'s value on $x^\star$ is at most $(\ell + q - 1)/n$).

Proposition 7.3. *Let $\mathcal{A}$ be any deterministic q-query local search algorithm. For any $f : \{0,1\}^n \to \{0,1\}$, let $x^{(1)}, \ldots, x^{(q)}$ be $\mathcal{A}$'s queries when it is asked to certify f's value on $x^\star = [1, \ldots, 1]$. The number of coordinates i on which $x_i^{(j)} = 0$ for some $j \in [q]$ is at most $q - 1$.*

Proof. By induction on j. For $j = 1$, a local search algorithm's first query must be $x^{(1)} = x^\star = [1, \ldots, 1]$ which has no coordinates set to 0. For $j > 1$, we know that $x^{(j)}$ is Hamming adjacent to some $x^{(j')}$ where $j' < j$. Thus, $x^{(j)}$ can have at most one coordinate i on which $x_i^{(j)} = 0$ but $x_i^{(j')} = 1$. The desired result holds by induction. □

Proposition 7.4. *Let $\mathcal{A}$ be any deterministic q-query local search algorithm and $\boldsymbol{f} : \{0,1\}^n \to \{0,1\}$ be a uniformly random dictator. The probability, over the randomness of $\boldsymbol{f}$, that $\boldsymbol{f}$'s value is 0 on least one of $\mathcal{A}$'s queries is at most $(q-1)/n$.*

Proof. For each $j \in [q]$, let $x^{(j)}$ be $\mathcal{A}$'s j^{th} query when $\boldsymbol{f}$'s value on its first $j - 1$ queries are all 1. Note that $\boldsymbol{f}$'s value is 0 on at least one of $\mathcal{A}$'s queries iff $\boldsymbol{f}(x^{(j)}) = 0$ for some $j \in [q]$. Hence

$$\begin{aligned} &\Pr_{\boldsymbol{f}} \left[\boldsymbol{f}\text{'s value is 0 on at least one of } \mathcal{A}\text{'s queries}\right] \\ &= \Pr_{\boldsymbol{f}} \left[\boldsymbol{f}(x^{(j)}) = 0 \text{ for some } j \in [q]\right] \\ &= \Pr_{\boldsymbol{i} \in [n]} \left[x_{\boldsymbol{i}}^{(j)} = 0 \text{ for some } j \in [q]\right] && \text{(Definition of } \boldsymbol{f}) \\ &\leq \frac{q-1}{n}. && \text{(Proposition 7.3)} \end{aligned}$$

□

We upper bound the probability any set S of size ℓ is a certificate for $\boldsymbol{f}$'s value on $x^\star = [1, \ldots, 1]$.

Proposition 7.5. *Fix any set $S \subseteq [n]$ of size ℓ. The probability, over the randomness of $\boldsymbol{f}$, that S is a certificate for $\boldsymbol{f}$'s value on $x^\star = [1, \ldots, 1]$ is at most ℓ/n.*

Proof. Recall that $\boldsymbol{f}(x) = x_{\boldsymbol{i}}$ for uniformly random $\boldsymbol{i} \in [n]$. Therefore S is a certificate for $\boldsymbol{f}$'s value on $x^\star$ iff $\boldsymbol{i} \in S$, which happens with probability $|S|/n = \ell/n$. □

With Propositions 7.4 and 7.5, we can now complete the proof of Claim 7.2:

Proof of Claim 7.2. As $\mathcal{A}$ is a deterministic algorithm, when $\boldsymbol{f}$'s values on $\mathcal{A}$'s queries are all 1, there is a single set of coordinates S output by $\mathcal{A}$. Then,

$$\begin{aligned}
&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ returns a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star\right] \\
=&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ returns a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star \;\&\right. \\
&\quad \left.\boldsymbol{f}\text{'s values on all queries are } 1\right] + \\
&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ returns a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star \;\&\right. \\
&\quad \left.\boldsymbol{f}\text{'s value on some query is } 0\right] \\
\le&\Pr_{\boldsymbol{f}}\left[S \text{ is a certificate for } \boldsymbol{f}\text{'s value on } x^\star\right] + \\
&\quad \Pr_{\boldsymbol{f}}\left[\boldsymbol{f}\text{'s value is 0 on at least one of } \mathcal{A}\text{'s queries}\right] \\
\le&\frac{\ell}{n} + \frac{q-1}{n}. \qquad \text{(Propositions 7.4 and 7.5)}
\end{aligned}$$

□

7.2 Proof of Claim 1.2

The proof is simple and is essentially an instantiation of the following elementary fact: if a problem P has $\ge M$ possible outputs, and the input to P can be accessed only via queries with binary answers, then $\log M$ is a lower bound on the query complexity of solving P. In our context of certification, since there are $\binom{n}{k}$ many sets of size k, this fact suggests that if every such set is a possible certificate, then $\log\left(\binom{n}{k}\right) \approx k \log n$ would be a lower bound on query complexity. Indeed this is what we show, and the argument extends easily to certification algorithms that are allowed to return a certificate of size $\ell \ge k$:

Claim 7.6. *Let $k, \ell, n, q \in \mathbb{N}$ and $\mathcal{A}$ be a q-query randomized algorithm. There is some monotone function $f : \{0,1\}^n \to \{0,1\}$ with $C(f) \le k$ and input $x^\star \in \{0,1\}^n$ on which $\mathcal{A}$ successfully returns a size-ℓ certificate for $x^\star$ with probability at most $2^q \cdot \binom{\ell}{k}/\binom{n}{k} \le 2^q \cdot \left(\frac{\ell e}{n}\right)^k$.*

Claim 1.2 follows as an immediate consequence of Claim 7.6: if $k \le \ell \le n^c$ for any $c < 1$, then $q \ge \Omega(k \log n)$ queries are necessary even to succeed with probability 0.1.

Proof. We will once again use Yao's lemma. Consider the distribution μ where:

(1) $\boldsymbol{x}$ is constant, supported entirely on $x^\star = [1, \ldots, 1]$, and
(2) $\boldsymbol{f}$ is drawn uniformly at random from the set of monotone conjunctions of k variables.

We observe that if f is the monotone conjunction of the variables some set T, then a set S certifies f's value on $x^\star$ iff $S \supseteq T$. Therefore, for any fixed set S of size at most ℓ,

$$\begin{aligned}
&\Pr_{\boldsymbol{f}}\left[S \text{ certifies } \boldsymbol{f}\text{'s value on } x^\star\right] \\
&= \Pr_{\boldsymbol{f}}\left[\boldsymbol{f} \text{ is a conjunction of } k \text{ variables within } S\right] \\
&= \frac{\binom{|S|}{k}}{\binom{n}{k}} \le \frac{\binom{\ell}{k}}{\binom{n}{k}}.
\end{aligned}$$

Since any deterministic q-query algorithm $\mathcal{A}$ can take on at most 2^q many output values, we have by a union bound that

$$\begin{aligned}
&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ finds a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star\right] \\
&\le 2^q \cdot \frac{\binom{\ell}{k}}{\binom{n}{k}} \le 2^q \cdot \frac{(\ell e/k)^k}{(n/k)^k} = 2^q \cdot \left(\frac{\ell e}{n}\right)^k.
\end{aligned}$$

Claim 7.6 follows from the above and an application of Yao's lemma. □

8 ALGORITHMS AND LOWER BOUNDS FOR OTHER SETTINGS

8.1 An Algorithm for Certifying Arbitrary Functions with Random Examples

Claim 8.1. *For any $k, m, n \in \mathbb{N}$, there is an algorithm which, given access to uniform random samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ of a function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\le k$, an input $x^\star \in \{0,1\}^n$, and f's value on $x^\star$, uses m random samples and returns a size-k certificate for f's value on $x^\star$ with probability at least*

$$1 - (1 - 2^{-k})^m \cdot \binom{n}{k}.$$

In particular, the algorithm succeeds with high probability if $m = \Theta(2^k k \log n)$.

Our proof of Claim 8.1 uses the following easy fact:

Proposition 8.2. *For every non-constant $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $\le k$ and every $b \in \{0,1\}$,*

$$\Pr_{\boldsymbol{x} \sim \{0,1\}^n}[f(\boldsymbol{x}) = b] \ge 2^{-k}.$$

Proof. Without loss of generality, we only prove that the probability $f(\boldsymbol{x}) = 1$ is at least 2^{-k}. As f is non-constant, there is some input y on which $f(y) = 1$. Since f has certificate complexity $\le k$, there is some set S of size $\le k$ where $f(x) = 1$ whenever $x_S = y_S$. Finally,

$$\Pr_{\boldsymbol{x} \sim \{0,1\}^n}[f(\boldsymbol{x}) = 1] \ge \Pr_{\boldsymbol{x} \sim \{0,1\}^n}[\boldsymbol{x}_S = y_S] \ge 2^{-k}.$$ □

Proof of Claim 8.1. We say that a set $S \subseteq [n]$ is *eliminated* by a sample $(x, f(x))$ if $x_S = x^\star_S$ and $f(x) \ne f(x^\star)$. The algorithm is simple: it iterates over all $\binom{n}{k}$ candidate size-k certificates (i.e. all size-k sets), keeping only those not eliminated by any of the m sample points, and returns an arbitrary one. Any actual certificate for f's value on $x^\star$ will not be eliminated by the above procedure. Therefore, if all non-certificates are eliminated, the output of this algorithm will be correct.

Fix any size-k set S that is *not* a certificate for f's value on $x^\star$, and consider $f_{x^\star_S}$, the subfunction of f obtained by restricting the

coordinates in S according to $x^\star$. Since f has certificate $\leq k$, all its subfunctions, including $f_{x^\star_S}$, also have certificate complexity $\leq k$. Furthermore, since S is not a certificate for f's value on $x^\star$, we have that $f_{x^\star_S}$ is non-constant. Hence, by Proposition 8.2,

$$\Pr_{\boldsymbol{x}\sim\{0,1\}^n}[f_{x^\star_S}(\boldsymbol{x}) \neq f(x^\star)] \geq 2^{-k}.$$

Therefore, the probability a random sample $(\boldsymbol{x}, f(\boldsymbol{x}))$ eliminates S is at least 2^{-k}. Since the samples are independent, the probability S is not eliminated after m samples is at most $(1-2^{-k})^m$. Union bounding over all $\binom{n}{k}$ possible non-certificates S of size k gives the desired result. □

8.2 Lower Bound on the Query Complexity of Certifying an Arbitrary Function

Claim 8.3. *Let $k, n, q, \ell \in \mathbb{N}$ and $\mathcal{A}$ be a q-query randomized algorithm. There is some $f : \{0,1\}^n \to \{0,1\}$ with $C(f) = k$ and input $x^\star \in \{0,1\}^n$ on which $\mathcal{A}$ successfully returns a size-ℓ certificate for $x^\star$ with probability at most $q \cdot 2^{-k} + (k\ell)/n$.*

Claim 8.3 implies that as long as $k \leq \ell$ satisfy $k\ell \leq 0.01n$, then $q \geq \Omega(2^k)$ queries are necessary even to succeed with probability 0.1. Combining this with the $q \geq \Omega(k \log n)$ lower bound we showed in Claim 1.2 yields the $q \geq \Omega(2^k + k \log n)$ lower bound stated in Table 1.

We apply Yao's lemma with the distribution μ where:

(1) $\boldsymbol{x}$ is constant, supported entirely on $x^\star = [1, \ldots, 1]$,
(2) $\boldsymbol{f}$ is the indicator function of a uniformly random subcube of codimension k. More formally, we select k uniformly random unique coordinates $\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_k \in [n]$ and k uniform random bits $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_k \sim \{0,1\}$, and let:

$$\boldsymbol{f}(x) = \begin{cases} 1 & \text{if } x_{\boldsymbol{i}_j} = \boldsymbol{b}_j \text{ for all } j \in [k] \\ 0 & \text{otherwise.} \end{cases}$$

By Yao's lemma, in order to prove Claim 8.3, we need only show that every q-query deterministic strategy successfully finds a size-ℓ certificate for $x^\star$ with probability at most $\frac{q}{2^k} + \frac{k\ell}{n}$ (over the randomness of $\boldsymbol{f}$). The proof of Claim 8.3 is similiar in spirit to Claim 7.2, and will follow from Propositions 8.4 and 8.5:

Proposition 8.4. *Let $\mathcal{A}$ be a q-query deterministic algorithm. The probability, over the randomness of $\boldsymbol{f}$, that $\boldsymbol{f}$'s value is 1 on at least one of $\mathcal{A}$'s queries is at most $q \cdot 2^{-k}$.*

Proof. Since $\mathcal{A}$ is a deterministic algorithm, the queries it makes are a deterministic function of the previous query outputs. For each $j \in [q]$, let $x^{(j)}$ be $\mathcal{A}$'s j^{th} query when $\boldsymbol{f}$'s value on its first $j-1$ queries are all 0. Note that $\boldsymbol{f}$'s value is 1 on at least one of $\mathcal{A}$'s queries iff there is some $j \in [q]$ for which $\boldsymbol{f}(x^{(j)}) = 1$. Hence

$$\begin{aligned}
&\Pr_{\boldsymbol{f}}\left[\boldsymbol{f}\text{'s value is 1 on at least one of } \mathcal{A}\text{'s queries}\right] \\
&= \Pr_{\boldsymbol{f}}\left[\boldsymbol{f}(x^{(j)}) = 1 \text{ for some } j \in [q]\right] \\
&\leq \sum_{j\in[q]} \Pr_{\boldsymbol{f}}\left[\boldsymbol{f}(x^{(j)}) = 1\right] && \text{(Union bound)} \\
&= \frac{q}{2^k}. && \text{(Definition of } \boldsymbol{f})
\end{aligned}$$

□

Proposition 8.5. *Fix a set $S \subseteq [n]$ of size ℓ. The probability, over the randomness of $\boldsymbol{f}$, that S is a certificate for $\boldsymbol{f}$'s value on $x^\star = [1, \ldots, 1]$ is at most $(k\ell)/n$.*

Proof. Recall that $\boldsymbol{f}$ is a function of k random coordinates $\boldsymbol{i}_1, \ldots, \boldsymbol{i}_k \sim [n]$. In order for S to be a certificate for $\boldsymbol{f}$'s value on $x^\star$, it has to contain at least one $\boldsymbol{i}_j$. Hence,

$$\begin{aligned}
&\Pr_{\boldsymbol{f}}\left[S \text{ is a certificate for } \boldsymbol{f}\text{'s value on } x^\star\right] \\
&\leq \Pr_{\boldsymbol{f}}\left[\boldsymbol{i}_j \in S \text{ for some } j \in [k]\right] \\
&\leq \sum_{j\in[k]} \Pr_{\boldsymbol{f}}\left[\boldsymbol{i}_j \in S\right] && \text{(Union bound)} \\
&\leq k \cdot \frac{\ell}{n}.
\end{aligned}$$

□

Proof of Claim 8.3. Let S be the set of coordinates output by $\mathcal{A}$ when $\boldsymbol{f}$'s values on its queries are all 0. Then,

$$\begin{aligned}
&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ returns a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star\right] \\
=&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ returns a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star \;\&\right. \\
&\left.\quad \boldsymbol{f}\text{'s values on all queries are } 0\right] + \\
&\Pr_{\boldsymbol{f}}\left[\mathcal{A} \text{ returns a size-}\ell \text{ certificate for } \boldsymbol{f}\text{'s value on } x^\star \;\&\right. \\
&\left.\quad \boldsymbol{f}\text{'s value on some query is } 1\right] \\
\leq&\Pr_{\boldsymbol{f}}\left[S \text{ is a certificate for } \boldsymbol{f}\text{'s value on } x^\star\right] + \\
&\Pr_{\boldsymbol{f}}\left[\boldsymbol{f}\text{'s value is 1 on at least one of } \mathcal{A}\text{'s queries}\right] \\
\leq&\frac{k\ell}{n} + \frac{q}{2^k}. && \text{(Propositions 8.4 and 8.5)}
\end{aligned}$$

□

8.3 Lower Bound on the Sample Complexity of Certifying a Monotone Function

Claim 8.6. *For $k \leq \ell \leq cn$ where c is a sufficiently small constant. Suppose $\mathcal{A}$ is an algorithm which satisfies the following: given q uniform random examples $(\boldsymbol{x}, f(\boldsymbol{x}))$ labeled by a monotone function $f : \{0,1\}^n \to \{0,1\}$ with $C(f) \leq k$ and an input $x^\star \in \{0,1\}^n$, we have that $\mathcal{A}$ returns a size-ℓ certificate for f's value on $x^\star$ w.h.p. Then $q = \Omega(2^k)$.*

Combining Claim 8.6 with the $q \geq \Omega(k \log n)$ lower bound we showed in Claim 7.6 yields the $q \geq \Omega(2^k + k \log n)$ lower bound stated in Table 1.

Proof. We will again apply Yao's lemma with $\boldsymbol{f}$ being a monotone conjunction of k random variables and $\boldsymbol{x}$ supported entirely on $x^\star = [1, \ldots, 1]$. (This is the same distribution as in the proof of Claim 7.6.) Let $\boldsymbol{Q}$ be q independent and uniform random elements $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(q)} \sim \{0,1\}^n$, and $\mathcal{A}$ be a deterministic algorithm.

By a union bound,

$$\Pr_{\boldsymbol{Q},\boldsymbol{f}}\left[\exists\, j \in [q] \text{ such that } \boldsymbol{f}(\boldsymbol{x}^{(j)}) = 1\right] \leq \frac{q}{2^k},$$

and so if $q \le c2^k$ for a sufficiently small constant c, it then follows by Markov's inequality that:

$$\Pr_Q\left[\Pr_f\left[\exists\, j \in [q] \text{ such that } f(x^{(j)}) = 1\right] \ge 0.01\right] \le 0.01. \quad (4)$$

Fix a $Q = \{x^{(1)}, \ldots, x^{(q)}\}$ for which

$$\Pr_f\left[f(x^{(j)}) = 0 \text{ for all } j \in [q]\right] \ge 0.99. \quad (5)$$

Since $\mathcal{A}$ is deterministic, it has to return the same size-ℓ set, call it S, for all f's that satisfy $f(x^{(i)}) = 0$ for all $j \in [q]$. This set S is a certificate for f's value on $x^\star = [1, \ldots, 1]$ iff f is the conjunction of k variables T where $T \subseteq S$, the probability of which is:

$$\Pr[T \subseteq S] = \frac{\binom{\ell}{k}}{\binom{n}{k}} \le \frac{\left(\frac{e\ell}{k}\right)^k}{\left(\frac{n}{k}\right)^k} = \left(\frac{e\ell}{n}\right)^k \le 0.01, \quad (6)$$

where the final inequality holds as long as $\ell \le cn$ for a sufficiently small constant c. Equations (4) to (6) imply that $\mathcal{A}$ succeeds with probability at most 0.1 over the randomness of f, and the claim follows by Yao's lemma. □

ACKNOWLEDGMENTS

We thank the STOC reviewers for their useful comments and feedback.

Guy, Caleb, and Li-Yang are supported by NSF CAREER Award 1942123. Caleb is also supported by an NDSEG fellowship. Jane is supported by NSF Award CCF-2006664.

A ANGLUIN'S ALGORITHM

In this section we give an overview of Angluin's algorithm adapted to our setting and a proof of correctness.

Algorithm 3 Reducing a certificate

Given: A monotone function $f : \{0,1\}^n \to \{0,1\}$, a b-certificate S for $b \in \{0,1\}$, and input $x^\star$.

1: SEEN $\leftarrow \emptyset$
2: Initialize $z_S \in \{0,1\}^n$ to be equal to $x^\star$ on coordinates in S and $1-b$ everywhere else
3: **while** $|S| \le C(f)$ **do**
4: Pick some $i \in S \setminus$ SEEN
5: If $f(z_S^{\oplus i}) \ne f(z_S)$ then add i to SEEN, otherwise remove i from S and update z_S
6: **end while**
7: **return** S.

Proof of Claim 6.3. Algorithm 3 gives a sketch of the procedure. Suppose without loss of generality that $f(x^\star) = 1$ and so S is a 1-certificate. We can continuously attempt to remove coordinates from S one at a time until $|S| \le C(f)$. For a 1-certificate S, write $z_S \in \{0,1\}^n$ for the string which has a 1 at each coordinate in S and 0s everywhere else. Note that $z_S \le x^\star$, $f(z_S) = 1$, and also $z_S \le y$ for all y satisfying $y|_S = z_S|_S$. For $i \in S$, we check if i is an irrelevant coordinate (Definition 3) by checking if flipping the i^{th} coordinate in z_S flips the output of the function. That is, we check if i is sensitive on z_S. If i is not sensitive, we remove i from S and recurse on $S \setminus \{i\}$. Otherwise, we leave i in S and do not check it again. We proceed in this fashion until $|S| \le C(f)$. Since we only check coordinates in S and check each such coordinate at most once we make $\le 2|S|$ queries to f.

To establish correctness, suppose this procedure returns S'. Since we only remove non-sensitive coordinates from S we have $f(z_{S'}) = 1$. For any y satisfying $y|_{S'} = z_{S'}|_{S'}$ we know that $y \ge z_{S'}$ and hence $f(y) = 1$ by monotonicity. It follows that S' is a 1-certificate for $z_{S'}$ and likewise for $x^\star$ as $S' \subseteq S$. Note also that if i is in S and i is sensitive for z_S then i remains sensitive for all $z_{S'}$ with $i \in S' \subseteq S$. In particular, $z_{S'} \le z_S$ and $z_{S'}^{\oplus i} \le z_S^{\oplus i}$ which shows $0 = f(z_S^{\oplus i}) \ge f(z_{S'}^{\oplus i})$ by monotonicity. Thus, any sensitive coordinate can be left in the certificate without having to check again. Moreover, since $\text{Sens}_f(z_S) \le C(f)$ we know that the number of sensitive indices we keep in the certificate S is at most $C(f)$ which ensures that if $|S| > C(f)$ there will always be some non-sensitive index that we can remove from S. □

REFERENCES

[1] Dana Angluin. 1988. Queries and concept learning. *Machine learning* 2, 4 (1988), 319–342.
[2] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to Explain Individual Classification Decisions. *Journal of Machine Learning Research* 11, 61 (2010), 1803–1831.
[3] Guy Blanc, Jane Lange, and Li-Yang Tan. 2021. Provably efficient, succinct, and precise explanations. *Advances in Neural Information Processing Systems* 34 (2021). Available at https://arxiv.org/abs/2111.01576.
[4] Harry Buhrman and Ronald de Wolf. 2002. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science* 288, 1 (2002), 21–43.
[5] Adnan Darwiche and Auguste Hirth. 2020. On The Reasons Behind Decisions. *European Conference on Artificial Intelligence (ECAI)* (2020), 712–720.
[6] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *ArXiv preprint* abs/1702.08608v2 (2017).
[7] Alexey Ignatiev. 2020. Towards Trustable Explainable AI. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 5154–5158. https://doi.org/10.24963/ijcai.2020/726 Early Career.
[8] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. 2019. Abduction-based explanations for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1511–1519.
[9] Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. 2019. On Validating, Repairing and Refining Heuristic ML Explanations. *CoRR* abs/1907.02509 (2019). arXiv:1907.02509 http://arxiv.org/abs/1907.02509
[10] Stasys Jukna. 2012. *Boolean function complexity: advances and frontiers*. Vol. 27. Springer.
[11] Been Kim. 2018. Introduction to Interpretable Machine Learning. Slides for a tutorial at the Deep Learning Summer School at the University of Toronto, Vector Institute. Available at https://beenkim.github.io/slides/DLSS2018Vector_Been.pdf.
[12] Janis Klaise, Arnaud Van Looveren, Giovanni Vacanti, and Alexandru Coca. 2021. Alibi Explain: Algorithms for Explaining Machine Learning Models. *Journal of Machine Learning Research* 22, 181 (2021), 1–7. http://jmlr.org/papers/v22/21-0017.html
[13] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 1885–1894.
[14] Zachary C. Lipton. 2018. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. *Queue* 16, 3 (June 2018), 31–57.
[15] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st Annual Conference on Advances in Neural Information Processing Systems (NeurIPS)*. 4765–4774.
[16] G. A. Margulis. 1974. Probabilistic properties of graphs with large connectivity. *Probl. Peredachi Inf.* 10, 2 (1974), 101–109.
[17] João Marques Silva, Thomas Gerspacher, Martin Cooper, Alexey Ignatiev, and Nina Narodytska. 2021. Explanations for Monotonic Classifiers. In *38th International Conference on Machine Learning (ICML 2021) (Proceedings of International Conference on Machine Learning (PMLR), Vol. 139)*, Marina Meila and Tong Zhang (Eds.). Machine Learning Research press, virtual, Austria. https://hal-univ-tlse3.archives-ouvertes.fr/hal-03311393

[18] Christoph Molnar. 2020. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. (2020). Availabe at https://christophm.github.io/interpretable-ml-book/.
[19] Ryan O'Donnell. 2014. *Analysis of Boolean Functions*. Cambridge University Press.
[20] Ryan O'Donnell, Michael Saks, Oded Schramm, and Rocco Servedio. 2005. Every Decision Tree Has an Influential Variable. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 31–39.
[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1135–1144.
[22] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. 1527–1535.
[23] Lucio Russo. 1978. A note on percolation. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 43, 1 (1978), 39–48.
[24] Andy Shih, Arthur Choi, and Adnan Darwiche. 2018. A symbolic approach to explaining Bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 5103–5111.
[25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop at International Conference on Learning Representations (ICLR)*.
[26] Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications Using Game Theory. *Journal of Machine Learning Research* 11 (March 2010), 1–18.
[27] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 3319–3328.
[28] Leslie Valiant. 1984. A theory of the learnable. *Commun. ACM* 27, 11 (1984), 1134–1142.
[29] Andrew Chi Chih Yao. 1977. Probabilistic computations: toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*. 222–227.