Electronic Colloquium on Computational Complexity, Report No. 10 (2026)

Nearly Tight Bounds on the Block Number of Boolean Functions in Terms of Sensitivity

Sourav Chakraborty* Anna Gál†

Abstract

This paper explores the previously studied measure called block number of Boolean functions, that counts the maximum possible number of minimal sensitive blocks for any input. We present close to tight upper bounds on the block number in terms of the function's sensitivity and the allowed block size, improving previous bounds by a quadratic factor. Moreover, our bound on the block number yields sharper upper bounds on DNF size and decision tree size. We obtain these results by introducing and estimating a novel measure called *brick number*, which not only upper bounds the block number but also leads to a new characterization of block sensitivity.

1 Introduction

Since the early nineties when Nisan [16] and Nisan and Szegedy [17] first conjectured till 2019 when Huang [10] proved the conjecture, the sensitivity conjecture was one of the most widely studied and fascinating conjectures in theoretical computer science. The conjecture, now theorem, can be stated in several equivalent forms and it establishes that a number of interesting complexity measures are all polynomially related to sensitivity.

Definition 1.1. *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and an input $x \in \{0,1\}^n$ the* sensitivity *of f at x, denoted $\mathsf{s}(f,x)$ is the number of indices $i \in \{1,\ldots,n\}$ such that $f(x) \neq f(x^i)$, where x^i denotes the string x with the ith bit flipped. If for an input x and an index i we have $f(x) \neq f(x^i)$, we call i a sensitive bit for x. The sensitivity of f at x is the number of sensitive bits of f for x. The* sensitivity *of the function f, denoted $\mathsf{s}(f)$, is defined as*

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

The *block sensitivity* of f at x, is defined as the number of disjoint blocks that are sensitive for x.

Definition 1.2. *A subset (or block) $B \subseteq \{1,\ldots,n\}$ is said to be sensitive for x if $f(x) \neq f(x^B)$, where x^B is the string obtained from x by flipping every index in B. For an input*

*Indian Statistical Institute, Kolkata, India, Email: sourav@isical.ac.in
†University of Texas at Austin, Austin, USA, Email: panni@cs.utexas.edu

ISSN 1433-8092

$x \in \{0,1\}^n$ the block sensitivity of f at x, denoted $\mathsf{bs}(f,x)$ is defined as the size of the largest collection $\mathcal{B}_x$ of disjoint sensitive blocks for x. The block sensitivity of f, denoted $\mathsf{bs}(f)$, is defined as

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

The sensitivity conjecture in its original form [16, 17] stated that there exists a universal constant $c \in \mathbb{R}^+$ such that for any Boolean function f,

$$\mathsf{bs}(f) \leq \mathsf{s}(f)^c. \tag{1}$$

This conjecture remained open for nearly three decades. Finally in 2019, Huang [10] proved the conjecture, by proving a tight quadratic upper bound on degree in terms of sensitivity.

Minimal Sensitive Blocks and Block Number

A simple observation of Nisan [16] is that the definition of block sensitivity is in fact equivalent to defining block sensitivity with respect to minimal sensitive blocks only.

Definition 1.3. *A block $B \subset \{1,\dots,n\}$ is said to be a* minimal sensitive block *for x if $f(x) \neq f(x^B)$, but for any strict subset $B \neq D \subset B$, $f(x) = f(x^D)$.*

To see that we can equivalently define block sensitivity considering minimal sensitive blocks only, first notice that disjoint minimal sensitive blocks are of course disjoint sensitive blocks, thus $bs(f,x) \geq bs'(f,x)$ (where bs' denotes block sensitivity with minimal sensitive blocks only). On the other hand, notice that $bs'(f,x) \geq bs(f,x)$, since each sensitive block contains at least one minimal sensitive block as a subset, and if two blocks are disjoint, then their subsets are also disjoint.

While block sensitivity is defined with respect to *disjoint* blocks, what happens if we do not insist on the blocks being disjoint has also been considered [15, 13, 5].

Definition 1.4. *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and an input $x \in \{0,1\}^n$ the* block number *of f at x, denoted $\mathsf{Nbs}(f,x)$ is the number of (not necessarily disjoint) minimal sensitive blocks of f for x. The* block number *of the function f, denoted $\mathsf{Nbs}(f)$, is defined as*

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

*The ℓ-*block number *of f at x, denoted $\mathsf{Nbs}_\ell(f,x)$ is the number of minimal sensitive blocks of size at most ℓ for x. The ℓ-*block number *of the function f, denoted $\mathsf{Nbs}_\ell(f)$, is defined as*

$$\mathsf{Nbs}_\ell(f) = \max_{x \in \{0,1\}^n} \mathsf{Nbs}_\ell(f,x).$$

Note that the size of any minimal sensitive block is at most $\mathsf{s}(f)$, so $\mathsf{Nbs}(f,x) = \mathsf{Nbs}_{\mathsf{s}(f)}(f,x)$.

1.1 Our Results

Bounds on block number and brick number In this paper we further strengthen the previous upper bounds [15, 13] on block number, and prove the following.

Theorem 1.5. *For any Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *and* $1 \leq \ell \leq n$,

$$\mathsf{Nbs}_\ell(f) \leq (6\mathsf{s}(f))^\ell$$

and thus we have $\mathsf{Nbs}(f) \leq (6\mathsf{s}(f))^{\mathsf{s}(f)}$.

We obtain our results by proving a stronger statement about a more general measure. Instead of considering minimal sensitive blocks for x, we consider blocks B such that f is sensitive to each bit of B on the input x^B. We refer to these blocks as *bricks*. We formally define the *brick number* of a function in Section 2, and prove upper bounds on it which will imply Theorem 1.5.

We show that our bounds on ℓ-block number and ℓ-brick number are tight up to constant factors for every fixed block size ℓ in Section 5. In addition, we also show that the block number as well as the brick number can be an exponential function of sensitivity (see Section 5). Note that this is in sharp contrast to block sensitivity which by now we know to be polynomially related to sensitivity [10].

Furthermore, we compare our new measure brick number with block number and show that brick number can be exponentially larger than block number. However, for monotone functions the two measures are equivalent (see Section 4).

New Characterization of Block Sensitivity Our main motivation to consider brick number was that this allows us to obtain upper bounds on the block number, but we find it to be an interesting measure on its own right. Recall that both measures count the number of possibly *overlapping* subsets (bricks and sensitive blocks, respectively). While brick number may be exponentially larger than block number (see Section 4), we observe that considering pairwise *disjoint* bricks we get a measure that turns out to be equivalent to block sensitivity (see Section 3).

Improved Bounds on Decision Tree Size and DNF Size Block number was used in [5] to obtain upper bounds on deterministic decision tree size in terms of randomized decision tree size. This was achieved in [5] by using block number as an intermediate measure two ways: in an upper bound on DNF size in terms of randomized decision tree size, and as a lower bound on a function of randomized decision tree size. Combining these two bounds implies upper bounds in terms of randomized decision tree size on DNF size directly, and on deterministic decision tree size using a theorem of Ehrenfeucht and Haussler [7]. We observe that using our upper bound on block number at various steps of the above reasoning gives improved bounds on deterministic decision tree size as well as DNF size. For some functions, our upper bounds on decision tree size and DNF size may be exponentially smaller than those obtained by previous methods. We discuss these implications in Section 6.

Implications on Randomized Query Complexity Midrijanis [15] proved that $\mathsf{R}_0(f) = O(\mathsf{R}_2(f) \log \mathsf{Nbs}(f))$, thus block number can be used to upper bound the zero-error randomized query complexity in terms of randomized query complexity. Our upper bound on block number gives at least a quadratic improvement over the previous best bounds by Kulkarni and Tal [13] for functions with a large enough gap between fractional block sensitivity and sensitivity, thus we also get a small (at most constant factor) improvement for the upper bounds on R_0 for such functions.

1.2 Prior Work

Sensitivity vs. Block Sensitivity

Block sensitivity was defined by Nisan [16], who showed that the CREW PRAM complexity of any Boolean function f is equal to $\Theta(\log \mathsf{bs}(f))$. It was then shown that block sensitivity is polynomially related to a number of interesting complexity measures such as certificate complexity and decision tree depth [16], as well as degree and approximate degree [17]. See [3] and [9] for a survey of more related measures and conjectures. It remained open for decades how these measures relate to sensitivity. Since all the above measures are polynomially related to block sensitivity, proving for any one of them a polynomial upper bound in terms of sensitivity implies the sensitivity conjecture as stated in Equation 1. Huang [10] achieved this by proving a tight quadratic upper bound on degree in terms of sensitivity.

Huang's result [10] was further strengthened by Laplante et al. [14] who showed that the degree of any Boolean function f is at most $\mathsf{s}_0(f)\mathsf{s}_1(f)$, where $\mathsf{s}_0(f)$ and $\mathsf{s}_1(f)$ denote maximum sensitivity over 0-inputs and 1-inputs, respectively. Huang's result implies the statement in Equation 1 about block sensitivity vs sensitivity with $c = 4$. It is still open whether the constant c can be lowered to 2, which is the current best lower bound on the value of c in Equation 1 [18, 4, 2, 21, 8, 6].

Before Huang's result one of the most significant attempts towards proving the sensitivity conjecture was made by Kenyon and Kutin [12]. They defined a refinement of the block sensitivity, restricting attention to blocks of bounded size.

Definition 1.6. *For any $x \in \{0,1\}^n$, the ℓ-block sensitivity of f at x, denoted $\mathsf{bs}_\ell(f,x)$, is the size of the largest collection $\mathcal{B}_x$ of disjoint sensitive blocks of x where each block in $\mathcal{B}_x$ has size at most ℓ. We define the ℓ-block sensitivity of f as*

$$\mathsf{bs}_\ell(f) = \max_{x \in \{0,1\}^n} \mathsf{bs}_\ell(f,x).$$

Kenyon and Kutin [12] proved that for any Boolean function f and any $1 \leq \ell \leq n$, $\mathsf{bs}_\ell(f) \leq O(\mathsf{s}(f)^\ell)$.

Block Number To the best of our knowledge, this measure was first considered by Midrijanis [15] and later by Kulkarni and Tal [13] who used it to upper bound the zero-error randomized query complexity in terms of randomized query complexity. Chattopadhyay et al. [5] used this measure to estimate deterministic vs randomized decision tree *size*, and introduced the name "block number" for the measure.

It is not hard to see [15] that for $f : \{0,1\}^n \to \{0,1\}$ we have $\mathsf{Nbs}(f) \leq n^{\mathsf{s}(f)}$. Kulkarni and Tal [13] further improved this bound to $\mathsf{Nbs}(f) \leq 2\mathsf{fbs}(f)^{\mathsf{s}(f)}$, by proving that

$$\mathsf{Nbs}_\ell(f) \leq 2\mathsf{fbs}(f)^\ell, \tag{2}$$

where $\mathsf{fbs}(f)$ denotes fractional block sensitivity.

1.3 Comparison with Previous Bounds

Our upper bounds show that essentially the same bounds (up to constant factors for fixed ℓ) hold for the ℓ-block number as the bounds of Kenyon and Kutin [12] for ℓ-block sensitivity. This is somewhat surprising, since the bound of [12] was proved for *disjoint* minimal sensitive blocks, and we prove that essentially the same bound holds even if the blocks are not required to be disjoint. This is a significantly different scenario, and the number of overlapping sensitive blocks may be exponentially larger than the number of pairwise disjoint sensitive blocks (see e.g. the examples in Section 5). In fact, the argument of Kenyon and Kutin [12] breaks down when trying to directly generalize it to "block number", that is, when relaxing the requirement that the blocks have to be disjoint. We discuss this further below.

We obtain our bounds by considering the brick number instead of directly estimating the block number, and this allows us to circumvent the difficulties in generalizing the proof.

The current best upper bound on fractional block sensitivity in terms of sensitivity is $\mathsf{fbs}(f) \leq O(\mathsf{s}(f)^4)$ [1], thus the bound in Equation 2 of [13] gives

$$\mathsf{Nbs}_\ell(f) \leq 2\mathsf{fbs}(f)^\ell \leq (c \cdot \mathsf{s}(f))^{4\ell}).$$

for some constant $c > 1$. Our theorem refines this upper bound on $\mathsf{Nbs}_\ell(f)$ from $(c \cdot \mathsf{s}(f))^{4\ell}$ to $(6\mathsf{s}(f))^\ell$. Since $\mathsf{fbs}(f) \geq \mathsf{bs}(f) \geq \mathsf{s}(f)$ for any Boolean function f (see e.g. [13]), our theorem yields a better bound than Equation 2 on the block number of functions f with $\mathsf{s}(f)$ significantly smaller than $\mathsf{fbs}(f)$. It is currently open what is the largest possible separation between $\mathsf{s}(f)$ and $\mathsf{fbs}(f)$, but we do know examples with $\mathsf{bs}(f)$ (and thus $\mathsf{fbs}(f)$) quadratically larger than $\mathsf{s}(f)$ [18, 4, 2, 21, 8, 6]. Thus, our bound gives at least a quadratic improvement over the previous bounds for functions with a large enough gap between fractional block sensitivity and sensitivity. Using our bounds on block number to upper bound deterministic decision tree size and DNF size leads to exponential improvements over the previous bounds in [5] for some functions.

Obstacles in Generalizing the Argument of Kenyon and Kutin

We briefly describe why the argument of Kenyon and Kutin [12] breaks down when trying to generalize it to "block number", that is when relaxing the requirement that the sensitive blocks have to be disjoint.

For the argument to work, it is crucial that the objects in A_k (see notation in Section 2) are the same type of objects with respect to x^k, as the objects we are trying to count with respect to x. In Kenyon and Kutin's argument, these are DISJOINT sensitive blocks. For their argument, it is ok to assume in addition that the blocks are MINIMAL sensitive with respect to x, but minimality with respect to x^k is not required to be included in A_k (otherwise the argument would break down).

If we try to extend this directly to block number (that is to counting minimal sensitive blocks without requiring disjointness), then in order to have the same type of objects in A_k with respect to x^k as the objects we are trying to count, we would have to include only MINIMAL sensitive blocks with respect to x^k into the set A_k. But then the argument breaks down, because the number of ways a block can fail to be a minimal sensitive block for x^k becomes too large. This is a subtle but crucial difference between the original argument for disjoint blocks and allowing overlappig blocks.

Our proof gets around these obstacles by working with brick number instead of block number.

2 Upper Bounds on Block Number and Brick Number

In this section we prove Theorem 1.5. In fact, we prove a stronger result, upper bounding *brick number* instead of *block number*.

Definition 2.1. *A block $B \subset \{1, \ldots, n\}$ is said to be a brick of f for x if for all $i \in B$, $f(x^B) \neq f(x^{B \setminus \{i\}})$.*

Notice that every minimal sensitive block of f for x is also a brick of f for x, but not the other way around. In fact, we do not even require that $f(x) \neq f(x^B)$ for a brick B, only that f is sensitive to each bit in B on x^B.

Definition 2.2. *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and an input $x \in \{0,1\}^n$ the* ℓ-brick number *of f at x, denoted $\mathsf{Nbr}_\ell(f, x)$ is the number of bricks of size at most ℓ of f for x. The* ℓ-brick number *of f, denoted $\mathsf{Nbr}_\ell(f)$, is*

$$\mathsf{Nbr}_\ell(f) = \max_{x \in \{0,1\}^n} \mathsf{Nbr}_\ell(f, x).$$

The brick number *of f at x, denoted $\mathsf{Nbr}(f, x)$ is the number of bricks of f for x. The* brick number *of the function f, denoted $\mathsf{Nbr}(f)$, is defined as*

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

Note that since the size of any brick of f is at most $\mathsf{s}(f)$, $\mathsf{Nbr}(f, x) = \mathsf{Nbr}_{\mathsf{s}(f)}(f, x)$.

We prove the following theorem, which immediately implies Theorem 1.5.

Theorem 2.3. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ and any $1 \leq \ell \leq n$,*

$$\mathsf{Nbr}_\ell(f) \leq (6\mathsf{s}(f))^\ell$$

Proof. This argument is a generalization of an argument of Kenyon and Kutin [12]. Kenyon and Kutin counted the maximum possible number of *disjoint* blocks B, such that $f(x) \neq f(x^B)$.

We change the conditions on what to count two ways: In our case, the blocks we count do not have to be disjoint, and instead of requiring that $f(x) \neq f(x^B)$, we require that for subsets B' of B that only differ from B in one coordinate, $f(x^{B'}) \neq f(x^B)$.

Remark 2.4. *Notice that our first change in the conditions is quite significant: we do not require the blocks to be disjoint. The number of overlapping blocks may be exponentially larger than the number of pairwise disjoint blocks (see e.g. Section 5). Thus while our argument follows the general approach of [12], there are subtle but important differences. Our second change corresponds to considering "bricks" instead of "sensitive blocks". We note that the argument of [12] breaks down when trying to directly generalize it to "block number", that is when relaxing the requirement that the blocks have to be disjoint. We discuss this in Section 1.3.*

Definition 2.5. *Given $x \in \{0,1\}^n$ and an integer ℓ, denote by $\mathcal{BR}^f_{x,\ell}$ the set of all blocks $B \subseteq [n]$ with the following properties:*

1. $1 \leq |B| \leq \ell$
2. *for any $j \in B$ we have $f(x^B) \neq f(x^{B \setminus \{j\}})$*

We refer to the blocks in $\mathcal{BR}^f_{x,\ell}$ as good blocks for x.

Note that with our terminology, $\mathcal{BR}^f_{x,\ell}$ is the set of bricks of f of size at most ℓ for x, thus the good blocks for x are simply the bricks of size at most ℓ for x.

We will estimate the maximum possible cardinality of $\mathcal{BR}^f_{x,\ell}$ as a function of the sensitivity of the given Boolean function f. Similarly to Kenyon and Kutin's argument, we will use a sequence of weights $w_1 \geq \ldots \geq w_\ell = 1$ that we specify later. For a collection $\mathcal{C}$ of blocks of size at most ℓ, we define

$$t(\mathcal{C}) = \sum_{B \in \mathcal{C}} w_{|B|}.$$

Notice that since each weight is at least 1, we have for any x,

$$t(\mathcal{BR}^f_{x,\ell}) \geq |\mathcal{BR}^f_{x,\ell}| \,. \tag{3}$$

Definition 2.6. *For given $f : \{0,1\}^n \to \{0,1\}$ let x be an input where $t(\mathcal{BR}^f_{x,\ell})$ is largest possible over all inputs in $\{0,1\}^n$. For the next part of the argument, we fix such x, and use the notation $\mathcal{B} = \mathcal{BR}^f_{x,\ell}$.*

Definition 2.7. *For each $k \in [n]$ we define $\mathcal{A}_k$ as follows.*

- *For each block $B \in \mathcal{B}$ of size $|B| \geq 2$ that contains k, include $B \setminus \{k\}$ in $\mathcal{A}_k$.*
- *In addition, if $f(x) \neq f(x^k)$ then include the block $\{k\}$ in $\mathcal{A}_k$.*
- *For each block $B \in \mathcal{B}$ that does not contain k, include B in $\mathcal{A}_k$ if B is a good block for x^k.*

Recall that by our notation, each block in $\mathcal{B}$ is a good block for x, but may or may not be a good block for x^k. Next we show that each block in $\mathcal{A}_k$ is a good block for x^k.

Claim 2.8. $\mathcal{A}_k \subseteq \mathcal{BR}^f_{x^k,\ell}$

Proof. First note that if $f(x) \neq f(x^k)$, then the block $\{k\}$ is a good block for x^k (as well as for x). Given the way we defined $\mathcal{A}_k$, now we only have to show that when $k \in B$ for $B \in \mathcal{B}$ of size $|B| \geq 2$, then the block $B \setminus \{k\}$ is a good block for x^k. Note that for any block A that contains k, we have $(x^k)^{A\setminus\{k\}} = x^A$. Therefore, considering $A = B \setminus \{j\}$ for $k \neq j \in B$, we see that the conditions on B being a good block for x imply that $B \setminus \{k\}$ is a good block for x^k. □

Let Γ_k denote the set of blocks $B \in \mathcal{B}$ of size $|B| \geq 2$ that contain k, and let Λ_k denote the set of blocks $B \in \mathcal{B}$ that do not contain k and are not good for x^k. Then, we have

$$t(\mathcal{A}_k) = t(\mathcal{B}) + \sum_{B\in\Gamma_k} (w_{|B|-1} - w_{|B|}) - \sum_{B\in\Lambda_k} w_{|B|} \,. \tag{4}$$

Since all the weights are positive, we have that for each $k \in [n]$

$$t(\mathcal{A}_k) \leq t(\mathcal{B}\mathcal{R}^f_{x^k,\ell}) \leq t(\mathcal{B}) \,,$$

where the first inequality follows by Claim 2.8, and the second follows by our choice of x and the set $\mathcal{B}$ in Definition 2.6. Thus, we get that

$$0 \leq \sum_{k=1}^{n} (t(\mathcal{B}) - t(\mathcal{A}_k)) \,. \tag{5}$$

For $1 \leq i \leq \ell$, let m_i denote the number of blocks of size i in $\mathcal{B}$. Notice that

$$\sum_{k=1}^{n} \sum_{B\in\Gamma_k} (w_{|B|-1} - w_{|B|}) = \sum_{i=2}^{\ell} i \cdot m_i \cdot (w_{i-1} - w_i) \,, \tag{6}$$

since each block $B \in \mathcal{B}$ of size i belongs to sets Γ_k for i different values of k.

Next, note that (since each block in $\mathcal{B}$ is good for x) if a block $B \in \mathcal{B}$ that does not contain k is not good for x^k, then either $f(x^B) \neq f((x^k)^B)$, or for some $j \in B$, $f(x^{B\setminus\{j\}}) \neq f((x^k)^{B\setminus\{j\}})$. Since for any block A we have $(x^k)^A = (x^A)^k$, we get that any block $B \in \mathcal{B}$ of size i can be not good for x^k for at most $\mathsf{s}(f) - i + i(\mathsf{s}(f) - 1) \leq \mathsf{s}(f)(i + 1)$ different values of k.

Thus, if $\mathsf{s}(f) \leq s$, combining equations 4, 5, and 6 we get

$$0 \leq \sum_{i=1}^{\ell} m_i \cdot w_i \cdot \mathsf{s} \cdot (i + 1) - \sum_{i=2}^{\ell} i \cdot m_i \cdot (w_{i-1} - w_i) \,. \tag{7}$$

Now, set $w_\ell = 1$ and $w_i = w$ for $1 \leq i \leq \ell - 1$. (We will fix the value of w later.) Then (7) gives

$$0 \leq sw \left(\sum_{i=1}^{\ell-1} m_i(i + 1) \right) + m_\ell \mathsf{s}(\ell + 1) - \ell m_\ell (w - 1) \,.$$

Using that $i + 1 \leq \ell$ inside the sum and rearranging, we get

$$m_\ell(\ell w - \ell - s - \ell s) \leq s\ell w \left(\sum_{i=1}^{\ell-1} m_i \right) .$$

This gives $m_\ell \leq \left(\sum_{i=1}^{\ell-1} m_i \right) \frac{s\ell w}{\ell w - \ell - s - \ell s}$ and

$$t(\mathcal{B}) = m_\ell + w \sum_{i=1}^{\ell-1} m_i \leq \left(\sum_{i=1}^{\ell-1} m_i \right) \left(w + \frac{s\ell w}{\ell w - \ell - s - \ell s} \right) . \tag{8}$$

Let N_ℓ denote the maximum possible cardinality of the set $\mathcal{BR}^f_{x,\ell}$ for any $x \in \{0,1\}^n$ and any Boolean function f with sensitivity at most s. Let f^* and x^* denote the function and input where this maximum is achieved for ℓ. Note that by definition, if m_i denotes the number of blocks of size i in $\mathcal{BR}^{f^*}_{x^*,\ell}$, then $\sum_{i=1}^{\ell} m_i = N_\ell$ and $\sum_{i=1}^{\ell-1} m_i \leq N_{\ell-1}$.

Since (8) holds for any Boolean function f with sensitivity at most s, repeating the above argument with f^* and x^* in mind, by (8) and (3) we get

$$N_\ell \leq N_{\ell-1} \left(w + \frac{s\ell w}{\ell w - \ell - s - \ell s} \right) .$$

Setting $w = 3s$ gives $N_\ell \leq 6sN_{\ell-1}$ for $\ell \geq 2$. Since $N_1 \leq s$, we get $N_\ell \leq (6s)^\ell$. It remains to note that for any Boolean function $f : \{0,1\}^n \to \{0,1\}$, any $x \in \{0,1\}^n$ and any $1 \leq \ell \leq n$,

$$\mathsf{Nbr}_\ell(f,x) = |\mathcal{BR}^f_{x,\ell}| \leq N_\ell,$$

and the statement of the theorem follows. □

3 A New Characterization of Block Sensitivity

One could ask what happens if we insist on *disjoint* bricks, and one could consider defining the *disjoint brick number* denoted $\mathsf{dbr}(f,x)$. We note that this definition is equivalent to block sensitivity.

Observation 3.1. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ and any input $x \in \{0,1\}^n$ we have $\mathsf{dbr}(f,x) = \mathsf{bs}(f,x)$ and $\mathsf{dbr}_\ell(f,x) = \mathsf{bs}_\ell(f,x)$.*

Proof. To see this, first note that since each minimal sensitive block is also a brick, we immediately have that $\mathsf{dbr}(f,x) \geq \mathsf{bs}(f,x)$. The perhaps more interesting (but still simple) direction is that $\mathsf{bs}(f,x) \geq \mathsf{dbr}(f,x)$ also holds: Let B be a brick of f for x. This implies that there is a subset $B' \subseteq B$ (where possibly $B' = B$) such that $f(x^{B'}) \neq f(x)$. This in turn implies that there is a subset $B'' \subseteq B'$ (where possibly $B'' = B'$) such that B'' is a minimal sensitive block of f for x. Since if two blocks are disjoint, then their subsets are also disjoint, we get that $\mathsf{bs}(f,x) \geq \mathsf{dbr}(f,x)$. By the same reasoning as above $\mathsf{dbr}_\ell(f,x) = \mathsf{bs}_\ell(f,x)$. □

Furthermore, we observe that this argument extends to considering arbitrary blocks where the function value is not constant on the corresponding subcube.

Given an input $x \in \{0,1\}^n$ and a subset of coordinates $B \subseteq [n]$, we denote by $x_{[n]\setminus B}$ the $(n-|B|)$-bit string that agrees with x on coordinates outside B. For $y \in \{0,1\}^{|B|}$ we denote by $x_{[n]\setminus B}|y$ the n-bit string that agrees with x on coordinates outside B and agrees with y on coordinates in B. We denote by $f_{B,x}$ the sub-function $f_{B,x} : \{0,1\}^{|B|} \to \{0,1\}$ defined as $f_{B,x}(y) = f(x_{[n]\setminus B}|y)$.

Observation 3.2. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ and any input $x \in \{0,1\}^n$, the block sensitivity $\mathsf{bs}(f,x)$ is equal to the size of the largest collection $\mathcal{C}_x$ of pairwise disjoint subsets of $[n]$, such that for each subset $B \in \mathcal{C}_x$ the sub-function $f_{B,x}$ is not constant. Furthermore, the ℓ-block sensitivity $\mathsf{bs}_\ell(f,x)$ is equal to the largest number of pairwise disjoint subsets of size at most ℓ such that for each subset B, the corresponding sub-function $f_{B,x}$ is not constant.*

The proof is similar to the proof of the previous observation.

Thus, one could equivalently define the block sensitivity of f at input x as the largest number of pairwise disjoint bricks of f for x, and we could also equivalently define block sensitivity of f at x as the largest number of pairwise disjoint subsets resulting in non-constant sub-functions $f_{B,x}$. Furthermore, these equivalences extend to ℓ-block sensitivity.

4 Brick Numbers vs Block Numbers

It is interesting to note that while considering *disjoint* bricks we get a measure that turns out to be equivalent to block sensitivity, brick number and block number (both allowing overlapping blocks) can be very different. Since every minimal sensitive block is also a brick, we have

$$\mathsf{Nbr}_\ell(f,x) \geq \mathsf{Nbs}_\ell(f,x),$$

as well as $\mathsf{Nbr}(f,x) \geq \mathsf{Nbs}(f,x)$ and $\mathsf{Nbr}(f) \geq \mathsf{Nbs}(f)$.

On the other hand, the brick number can be significantly larger than the block number. Consider the parity function, PARITY, where $\mathsf{PARITY}(x_1,\ldots,x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$. For any $x \in \{0,1\}^n$ note that PARITY is sensitive to every bit of x. Thus, the only minimal sensitive blocks are the sets of size 1, implying

$$\mathsf{Nbs}(\mathsf{PARITY},x) = \mathsf{Nbs}_1(\mathsf{PARITY},x) = n.$$

But we note that every nonempty set is a brick of PARITY for any input x. Thus, the ℓ-brick number is the number of nonempty subsets of $[n]$ of size at most ℓ, and we have $\mathsf{Nbr}_\ell(f,x) = \sum_{i=1}^{\ell} \binom{n}{i}$, and

$$\mathsf{Nbr}(\mathsf{PARITY},x) = \mathsf{Nbr}_n(\mathsf{PARITY},x) = 2^n - 1.$$

For the special case of *monotone* functions, the brick number is the same as the block number.

Lemma 4.1. *For any* monotone *Boolean function $f : \{0,1\}^n \to \{0,1\}$ and any input $x \in \{0,1\}^n$, we have $\mathsf{Nbr}(f,x) = \mathsf{Nbs}(f,x)$ and $\mathsf{Nbr}_\ell(f,x) = \mathsf{Nbs}_\ell(f,x)$.*

Proof. For any $x \in \{0,1\}^n$ let One_x and Zero_x be the partition of $\{1,\ldots,n\}$ where One_x is the set of indices of x that are 1 and similarly Zero_x is the set of indices of x that are 0.

The following simple observation follows directly by the definition of monotone functions.

Observation 4.2. *For a monotone Boolean function f, if $x, y, z \in \{0,1\}^n$ are such that $\mathsf{One}_x \subsetneq \mathsf{One}_y \subsetneq \mathsf{One}_z$ then it cannot be that $f(x) = f(z)$ but $f(x) \neq f(y)$.*

Now we have the following claim:

Claim 4.3. *If $f : \{0,1\}^n$ is a monotone function then for any x, if B is a brick for x, then either $B \subseteq \mathsf{Zero}_x$ or $B \subseteq \mathsf{One}_x$.*

Proof. Assume the contrary, and let B be a brick for x, with $B \cap \mathsf{Zero}_x \neq \emptyset$ and $B \cap \mathsf{One}_x \neq \emptyset$. Let $i \in B \cap \mathsf{Zero}_x$ and $j \in B \cap \mathsf{One}_x$ be two such indices. By the definition of a brick, $f(x^{B\setminus\{j\}}) \neq f(x^B)$ and $f(x^{B\setminus\{i\}}) \neq f(x^B)$. That is $f(x^{B\setminus\{j\}}) = f(x^{B\setminus\{i\}})$.

But notice that $\mathsf{One}_{x^{B\setminus\{i\}}} = \mathsf{One}_{x^B} \setminus \{i\}$ and $\mathsf{One}_{x^B} = \mathsf{One}_{x^{B\setminus\{j\}}} \setminus \{j\}$. Thus,

$$\mathsf{One}_{x^{B\setminus\{i\}}} \subsetneq \mathsf{One}_{x^B} \subsetneq \mathsf{One}_{x^{B\setminus\{j\}}}.$$

Observation 4.2 says that this is impossible for a monotone function f. □

Using the above claim we will be able to complete the proof of the lemma. We will show that if $f : \{0,1\}^n$ is a monotone Boolean function then for any $x \in \{0,1\}^n$, if B is a brick for x then B is a minimal sensitive block for x.

Let B be a brick for x. Assume that $B \subseteq \mathsf{Zero}_x$, which means that $\mathsf{One}_x \subsetneq \mathsf{One}_{x^B}$. By Claim 4.3, the only remaining case is that $B \subseteq \mathsf{One}_x$, which can be handled analogously.

If $|B| = 1$ (say $B = \{i\}$), then, by the definition of a brick, $f(x^B) \neq f(x^{B\setminus\{i\}}) = f(x)$ and thus clearly B is a minimal sensitive block.

Next if $|B| \geq 2$, let $i \in B$. Then by the definition of a brick $f(x^B) \neq f(x^{B\setminus\{i\}})$. Also, observe that $\mathsf{One}_x \subsetneq \mathsf{One}_{x^{B\setminus\{i\}}} \subsetneq \mathsf{One}_{x^B}$. So by Observation 4.2 it must be that $f(x) \neq f(x^B)$. Thus, any brick for x is a sensitive block.

Finally, we show that any brick for x is also a minimal sensitive block. The arguments are similar. Let $A \subsetneq B$ be a smaller sensitive block. Thus, $f(x^A) \neq f(x)$ and $f(x^A) = f(x^B)$ (since we just proved that $f(x) \neq f(x^B)$. But by the definition of a brick, we know that $|B \setminus A| \geq 2$, else $A = B \setminus \{i\}$ and in that case $f(x^B) \neq f(x^{B\setminus\{i\}}) = f(x^A)$. Thus, let $j \in B \setminus A$, with $A \subsetneq B \setminus \{j\}$. So we have $f(x^A) = f(x^B) \neq f(x^{B\setminus\{j\}})$. But $One_{x^A} \subsetneq \mathsf{One}_{x^{B\setminus\{j\}}} \subsetneq \mathsf{One}_{x^B}$ and this cannot be possible by Observation 4.2. So B must be a minimal sensitive block. □

Recall that we observed yet another characterization of block sensitivity of f at x as the largest number of pairwise disjoint blocks resulting in non-constant sub-functions $f_{B,x}$. While considering pairwise disjoint blocks with this property is again equivalent to block sensitivity, the number of possibly overlapping blocks yielding non-constant sub-functions can be much larger than either the brick number or the block number. In contrast to brick number and block number, this is not bounded by any polynomial function of sensitivity for block sizes up to fixed ℓ, and can be as large as $2^{\Omega(\mathsf{s}(f)\ell)}$. Please see Section C for more details.

5 Tightness of Our Result

Since $\mathsf{bs}(f) \leq O(\deg(f))^2$ [17], as a consequence of Huang's proof of the sensitivity conjecture [10] we know that $\mathsf{bs}(f) \leq O(\mathsf{s}(f)^4)$. This implies that $\mathsf{bs}_\ell(f) \leq O(\mathsf{s}(f)^4)$ for all ℓ.

In contrast, we argue that our bounds on ℓ-block numbers and ℓ-brick numbers (considering non-disjoint minimal sensitive blocks and non-disjoint bricks) are close to tight, and the block number (as well as the brick number) can be an exponential function of sensitivity.

Consider the threshold function, $\mathsf{Threshold}^n_t : \{0,1\}^n \to \{0,1\}$, defined as

$$\mathsf{Threshold}^n_t(x) = 1 \text{ iff } |x| \geq t,$$

where $|x|$ denotes the number of 1-s in x (the Hamming weight of x). We know that for any $t \geq 1$, $\mathsf{s}(\mathsf{Threshold}^n_t) = \max(t, n-t) = \Theta(n)$.

In contrast, both the block number and the brick number of $\mathsf{Threshold}^n_t$ heavily depend on t. For any x, with $|x| < t$ a block S is a brick for x if and only if $|x^S| = t$ and if $|x| \geq t$ then a block S is a brick for x if and only if $|x^S| = (t-1)$. Notice that this means that each brick of a threshold function is also a minimal sensitive block. Thus, in the special case of threshold functions, the brick numbers are equal to the corresponding block numbers. (In fact, this property holds for any monotone function, as noted in Lemma 4.1.) We get that for any x, with $t - \ell \leq |x| < t$,

$$\mathsf{Nbr}_\ell(\mathsf{Threshold}^n_t, x) \geq \binom{n-|x|}{t-|x|}$$

Similarly, if $t \leq |x| < t + \ell$

$$\mathsf{Nbr}_\ell(\mathsf{Threshold}^n_t, x) \geq \binom{|x|}{|x|-t+1}.$$

Thus, for $\ell \leq \min(t, n-t+1)$ we have

$$\mathsf{Nbr}_\ell(\mathsf{Threshold}^n_t) \geq \max\left\{\binom{t+\ell-1}{\ell}, \binom{n-t+\ell}{\ell}\right\}.$$

If t is constant then the brick number, and thus the block number, of $\mathsf{Threshold}^n_t$ are polynomially related to its sensitivity. However, if t is large say $\lfloor n/2 \rfloor$, then the brick number and block number can be a super-polynomial function of sensitivity.

The TRIBES function illustrates that our results are tight up to constant factors for every fixed ℓ. The function $\mathsf{TRIBES}_{(w,s)} : \{0.1\}^{ws} \to \{0,1\}$ is defined as

$$\mathsf{TRIBES}_{(w,s)}(x) = \bigwedge_{i=1}^{w} \bigvee_{j=1}^{s} x_{ij},$$

where $x = x_{11} \ldots x_{1s} x_{21} \ldots x_{2s} \ldots x_{w1} \ldots x_{ws}$. The sensitivity of $\mathsf{TRIBES}_{(\sqrt{n},\sqrt{n})}$ is $\sqrt{n}$. Now consider the input x where for all $(\ell+1) \leq i \leq \sqrt{n}$, $x_{i1} = 1$ and rest of the bits are 0. Clearly, $\mathsf{TRIBES}_{(\sqrt{n},\sqrt{n})}(x) = 0$ and a block B is sensitive if and only if for all $1 \leq i \leq \ell$ the set $\{x_{i1}, \ldots, x_{i\sqrt{n}}\}$ has a non-zero intersection with B. Thus, the number of minimal sensitive

blocks of size ℓ for x is $\sqrt{n}^{\ell}$. Hence, the ℓ-block number for $\mathsf{TRIBES}_{(\sqrt{n},\sqrt{n})}$ is at least $\sqrt{n}^{\ell}$. Hence,

$$\mathsf{Nbr}_{\ell}(\mathsf{TRIBES}_{(\sqrt{n},\sqrt{n})}) \geq \mathsf{Nbs}_{\ell}(\mathsf{TRIBES}_{(\sqrt{n},\sqrt{n})}) \geq \mathsf{s}(\mathsf{TRIBES}_{(\sqrt{n},\sqrt{n})})^{\ell}.$$

Both the TRIBES function and the Threshold function are monotone. Hence for both of these functions fractional block sensitivity is the same as sensitivity and also the ℓ-brick number is the same as the ℓ-block number. Thus, for both functions the upper bound on the ℓ-block number of [13] is also tight. However for the Rubinstein function [18] sensitivity is $O(\sqrt{n})$ but fractional block sensitivity is $\Theta(n)$, and the upper bound on ℓ-block number obtained by [13] is $(O(n))^{\ell}$, whereas our upper bound is $(O(\sqrt{n}))^{\ell}$.

6 Bounds on Decision Tree Size and DNF Size

First we discuss some definitions and notation. Additional (standard) definitions related to decision trees can be found in Appendix A.

Definition 6.1 (Cover Number by Monochromatic Subcubes)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. A* subcube *of $\{0,1\}^n$ is a set of inputs obtained by fixing some subset of variables and leaving the others free. Formally, for a partial assignment $p \in \{0,1,*\}^n$, define*

$$C(p) = \{\, x \in \{0,1\}^n : \forall i \ (p_i \neq * \Rightarrow x_i = p_i) \,\}.$$

We say that $C(p)$ is a 1-subcube *of f if $f(x) = 1$ for all $x \in C(p)$.*

A collection of 1-subcubes $\mathcal{C} = \{C(p_1), C(p_2), \ldots, C(p_t)\}$ is said to cover *$f^{-1}(1)$ if $f^{-1}(1) = \bigcup_{i=1}^{t} C(p_i)$.*

The 1-cover number *of f, denoted $\mathsf{Cov}_1(f)$, is the minimum number of 1-subcubes needed to cover all 1-inputs of f.*

Similarly, the 0-cover number *of f, denoted $\mathsf{Cov}_0(f)$, is defined as the minimum number of 0-subcubes needed to cover all 0-inputs of f:*

Finally, the cover number *of f is defined as $\mathsf{Cov}(f) = \mathsf{Cov}_0(f) + \mathsf{Cov}_1(f)$..*

Relation to DNF Size Here we refer to DNF size (denoted DNFSize) as the number of terms of the DNF and to CNF size (denoted CNFSize) as the number of clauses of the CNF. A DNF representation of size s for f gives a 1-*cover* of f with s 1-subcubes. On the other hand, a 1-*cover* of f with s 1-subcubes gives a DNF of size s for f.

Similarly, we have a DNF of size t for $\neg f$ if and only if there is a 0-*cover* of f with t 0-cubes. By De Morgan rules, there is a DNF of size t for $\neg f$ if and only if there is a CNF of size t for f. Thus we have the following observation.

Observation 6.2. *$\mathsf{DNFSize}(f) = \mathsf{Cov}_1(f)$ and $\mathsf{CNFSize}(f) = \mathsf{Cov}_0(f)$.*

Relation to Decision Tree Size Every leaf of a deterministic decision tree computing f corresponds to a monochromatic subcube of $\{0,1\}^n$ (all inputs reaching that leaf share the same function value). Hence, any decision tree of size s yields a cover of both $f^{-1}(1)$ and $f^{-1}(0)$ by at most s subcubes.

Thus, the cover number of f provides a lower bound on the deterministic decision tree size, and an upper bound on DNF size (and CNF size):

$$\mathsf{DSize}(f) \ \geq \ \mathsf{Cov}(f) \geq \mathsf{DNFSize}(f).$$

The following theorem of Ehrenfeucht and Haussler [7] upper bounds the deterministic decision tree size by a function of the cover number.

Theorem 6.3. *[[7], see also Theorem 14.32 in [11]] For any Boolean function $f : \{0,1\}^n \to \{0,1\}$*

$$\mathsf{DSize}(f) = O((\mathsf{Cov}(f))^2 \log n).$$

Using this theorem, Chattopadhyay et al. [5] gave the following upper bound on the deterministic decision tree size by randomized decision tree size.

Theorem 6.4. *[5] For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\log(\mathsf{DSize}(f)) = O((\log(\mathsf{RSize}_{\frac{1}{3}}(f)))^4 \log^3 n).$$

They achieved proving this in two steps, and then using Theorem 6.3.

(1) They proved an upper bound on the block number in terms of randomized decision tree size.

$$\mathsf{Nbs}(f) \leq O(\mathsf{RSize}_{\frac{1}{3}}(f)^{2\log n}). \tag{9}$$

(2) They proved the following upper bound on the cover number

$$\mathsf{Cov}(f) \leq n\mathsf{RSize}_{\frac{1}{3}}(f)^{2\log(\mathsf{Nbs}(f))}. \tag{10}$$

This together with their bound in step (1) gives

$$\log(\mathsf{Cov}(f)) = O((\log(\mathsf{RSize}_{\frac{1}{3}}(f)))^2 \log n) \tag{11}$$

Using our upper bound on block number (Theorem 1.5), we obtain the following bounds on the cover number and on the deterministic decision tree size respectively, in terms of the randomized decision tree size and sensitivity.

Theorem 6.5. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\log(\mathsf{Cov}(f)) = O(\mathsf{s}(f) \log(\mathsf{s}(f)) \log(\mathsf{RSize}_{\frac{1}{3}}(f))$$

Theorem 6.5 follows by using our upper bound on the block number from Theorem 1.5 in Equation 10.

Theorem 6.6. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\log(\mathsf{DSize}(f)) = O(\mathsf{s}(f)^2 (\log(\mathsf{s}(f))^2 (\log \mathsf{RSize}_{\frac{1}{3}}(f))^2 \log n).$$

Theorem 6.6 follows by combining Theorem 6.3 and Theorem 6.5.

The upper bounds obtained by [5] are in terms of randomized decision tree size alone, while our upper bounds on decision tree and DNF size are in terms of sensitivity and randomized decision tree size together. It is natural to consider what bounds can we obtain in terms of sensitivity alone. Upper bounds on decision tree size in terms of sensitivity alone follow by bounds on decision tree depth, denoted $\mathsf{D}(f)$, since $\mathsf{DSize}(f) \leq 2^{\mathsf{D}(f)}$. Huang's breakthrough work [10] (upper bounding degree by the square of sensitivity) together with [15], where $\mathsf{D}(f)$ is upper bounded by the cube of the degree, imply that for any Boolean function $f : \{0,1\}^n \to \{0,1\}$

$$\log(\mathsf{DSize}(f)) = O(\mathsf{s}(f)^6). \tag{12}$$

We observe that our Theorem 6.6 may give better upper bounds on deterministic decision tree size, compared to both Theorem 6.4 and Equation 12, depending on the relation between randomized decision tree size and sensitivity. We get improved bounds compared to Theorem 6.4 when sensitivity is small with respect to the logarithm of randomized decision tree size. Notice that this is the case for example when sensitivity is small with respect to randomized query complexity. Comparing with Equation 12 we get better bounds as long as the logarithm of randomized decision tree size is smaller (by at least logarithmic factors) than $O(\mathsf{s}(f)^2)$.

Similarly, our Theorem 6.5 may give better upper bounds on DNF and CNF size compared to Equation 11 obtained by [5] when sensitivity is small with respect to the logarithm of randomized decision tree size. Considering upper bounds in terms of sensitivity alone, one can upper bound DNF size in terms of sensitivity by noticing that log of DNF size is at most $O(\log n)$ times the certificate complexity of the function, which by Huang's result is at most $O(s(f)^5)$. Thus, for any Boolean function $f : \{0,1\}^n \to \{0,1\}$

$$\log(\mathsf{DNFSize}(f)) = O(\mathsf{s}(f)^5 \log n). \tag{13}$$

Our Theorem 6.5 gives better bounds on DNF size as long as the logarithm of randomized decision tree size is smaller (by at least logarithmic factors) than $O(\mathsf{s}(f)^4)$.

In Appendix B we construct a Boolean function whose sensitivity is significantly smaller than the logarithm of the randomized decision tree size, which in turn is significantly smaller than $O(\mathsf{s}(f)^2)$. This shows that our results may yield stronger upper bounds on deterministic decision tree size and DNF size, compared to previous bounds.

References

[1] Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of Huang's sensitivity theorem. In *STOC*, pages 1330–1342. ACM, 2021.

[2] Andris Ambainis and Xiaoming Sun. New separation between *s(f)* and *bs(f)*. *Electron. Colloquium Comput. Complex.*, TR11-116, 2011.

[3] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.

[4] Sourav Chakraborty. On the sensitivity of cyclically-invariant boolean functions. *Discret. Math. Theor. Comput. Sci.*, 13(4):51–60, 2011.

[5] Arkadev Chattopadhyay, Yogesh Dahiya, Nikhil S. Mande, Jaikumar Radhakrishnan, and Swagato Sanyal. Randomized versus deterministic decision tree size. In *STOC*, pages 867–880, 2023.

[6] Siddhesh Chaubal and Anna Gál. New constructions with quadratic separation between sensitivity and block sensitivity. In *FSTTCS*, volume 122, pages 13:1–13:16, 2018.

[7] Andrzej Ehrenfeucht and David Haussler. Learning decision trees from random examples. *Inf. Comput.*, 82(3):231–246, 1989.

[8] Parikshit Gopalan, Rocco A. Servedio, and Avi Wigderson. Degree and sensitivity: Tails of two distributions. In *CCC*, volume 50 of *LIPIcs*, pages 13:1–13:23, 2016.

[9] Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. Variations on the sensitivity conjecture. *Theory Comput.*, 4:1–27, 2011.

[10] Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190 (3), 2019.

[11] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27. Springer, 2012.

[12] Claire Kenyon and Samuel Kutin. Sensitivity, block sensitivity, and l-block sensitivity of boolean functions. *Inf. Comput.*, 189(1):43–53, 2004.

[13] Raghav Kulkarni and Avishay Tal. On fractional block sensitivity. *Chic. J. Theor. Comput. Sci.*, 2016, 2016.

[14] Sophie Laplante, Reza Naserasr, and Anupa Sunny. Sensitivity lower bounds from linear dependencies. In *MFCS*, volume 170, pages 62:1–62:14, 2020.

[15] Gatis Midrijanis. On randomized and quantum query complexities, 2005.

[16] Noam Nisan. CREW prams and decision trees. *SIAM J. Comput.*, 20(6):999–1007, 1991.

[17] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994.

[18] David Rubinstein. Sensitivity vs. block sensitivity of boolean functions. *Comb.*, 15(2):297–299, 1995.

[19] Michael Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *FOCS*, pages 29–38, 1986.

[20] Avishay Tal. Properties and applications of boolean function composition. In *ITCS*, pages 441–454. ACM, 2013.

[21] Madars Virza. Sensitivity versus block sensitivity of boolean functions. *Inf. Process. Lett.*, 111(9):433–435, 2011.

A Additional Definitions

Definition A.1 (Deterministic Decision Tree Size)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. A* deterministic decision tree *for f is a rooted binary tree where:*

- *each internal node is labeled with an input variable x_i;*
- *the two outgoing edges correspond to the two possible values of x_i: one for $x_i = 0$ and one for $x_i = 1$;*
- *each leaf is labeled with an output value in $\{0,1\}$.*

The tree computes *f if, for every input $x \in \{0,1\}^n$, the value $f(x)$ equals the label of the leaf reached by following the path determined by the bits of x.*

The size *of a decision tree is the number of leaves (or equivalently, the number of nodes) in the tree. The* deterministic decision tree size *of f, denoted $\mathsf{DSize}(f)$, is the minimum size among all deterministic decision trees that compute f:*

$$\mathsf{DSize}(f) = \min_{T \text{ computes } f} \text{size}(T).$$

The deterministic decision tree depth *of f, denoted $\mathrm{D}(f)$, is the minimum depth of a deterministic decision tree computing f. While $\mathsf{DSize}(f)$ captures the overall size of a decision procedure, $\mathrm{D}(f)$ measures the worst-case number of variable queries.*

Definition A.2 (Randomized Decision Tree Size)**.** *A* randomized decision tree *for $f : \{0,1\}^n \to \{0,1\}$ is a probability distribution $\mathcal{T}$ over deterministic decision trees, such that for every input $x \in \{0,1\}^n$,*

$$\Pr_{T \sim \mathcal{T}}[T(x) = f(x)] \geq 2/3,$$

where the probability is over the random choice of the tree T from the distribution $\mathcal{T}$.

The size *of a randomized decision tree $\mathcal{T}$ is defined as the expected size of a tree drawn from $\mathcal{T}$:*

$$\text{size}(\mathcal{T}) = \mathbb{E}_{T \sim \mathcal{T}}[\text{size}(T)].$$

The randomized decision tree size *of f, denoted $\mathsf{RSize}_{\frac{1}{3}}(f)$, is the minimum expected size over all distributions $\mathcal{T}$ that compute f with error at most $1/3$:*

$$\mathsf{RSize}_{\frac{1}{3}}(f) = \min_{\mathcal{T}} \ \mathbb{E}_{T \sim \mathcal{T}}[\text{size}(T)] \quad \textit{subject to} \quad \Pr_{T \sim \mathcal{T}}[T(x) = f(x)] \geq \frac{2}{3} \textit{ for all } x.$$

B Examples of Functions with Improved Bounds on Decision Tree Size and DNF Size

Let us first consider the function $\mathsf{AND\text{-}OR\text{-}tree}_n : \{0,1\}^n \to \{0,1\}$.

Definition B.1 (AND-OR-tree Function)**.** *For any n that is a power of 2, that is, $n = 2^t$ for some integer t, the* $\mathsf{AND\text{-}OR\text{-}tree}_n$ *function is a Boolean function defined by a complete binary tree whose internal nodes alternate between $\wedge$ (AND) and $\vee$ (OR) gates, and whose leaves correspond to the input variables.*

Formally, for height $h \geq 0$, define the function $T_h : \{0,1\}^{2^h} \to \{0,1\}$ recursively by

$$T_0(x_1) = x_1,$$

$$T_h(x_1, \ldots, x_{2^h}) = \begin{cases} T_{h-1}(x_1, \ldots, x_{2^{h-1}}) \wedge T_{h-1}(x_{2^{h-1}+1}, \ldots, x_{2^h}) & \text{if } h \text{ is even,} \\ T_{h-1}(x_1, \ldots, x_{2^{h-1}}) \vee T_{h-1}(x_{2^{h-1}+1}, \ldots, x_{2^h}) & \text{if } h \text{ is odd.} \end{cases}$$

For instance, $T_1(x_1, x_2) = x_1 \vee x_2$, and

$$T_2(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \wedge (x_3 \vee x_4).$$

We make the following observation.

Observation B.2. $\mathsf{s}(\mathsf{AND\text{-}OR\text{-}tree}_n) = \Theta(\sqrt{n})$.

Proof. Recall that $\mathsf{s}_0(f)$ and $\mathsf{s}_1(f)$ denote maximum sensitivity over 0-inputs and 1-inputs, respectively. The statement follows from the fact that for f and g over disjoint sets of input variables, $\mathsf{s}_1(f \vee g) = \max\{\mathsf{s}(f), \mathsf{s}(g)\}$, while $\mathsf{s}_0(f \vee g) = \mathsf{s}(f) + \mathsf{s}(g)$. On the other hand, $\mathsf{s}_0(f \wedge g) = \max\{\mathsf{s}(f), \mathsf{s}(g)\}$ while $\mathsf{s}_1(f \wedge g) = \mathsf{s}(f) + \mathsf{s}(g)$.

Since the $\mathsf{AND\text{-}OR\text{-}tree}$ function has alternate compositions of $\vee$ and $\wedge$ over disjoint inputs we obtain the bound on $\mathsf{s}(\mathsf{AND\text{-}OR\text{-}tree}_n)$. □

It was proved in [19] that $\mathsf{RSize}_{\frac{1}{3}}(\mathsf{AND\text{-}OR\text{-}tree}_n) = 2^{\widetilde{O}(n^{0.753\ldots})}$. [5] proved a nearly matching lower bound, showing that $\log(\mathsf{RSize}_{\frac{1}{3}}(\mathsf{AND\text{-}OR\text{-}tree}_n)) = \widetilde{\Theta}(n^{0.753\ldots})$.

For our example we compose this function with the addressing function.

Definition B.3. *The "addressing function"* $\mathsf{ADDRESS}_k : \{0,1\}^{k+2^k} \to \{0,1\}$ *is defined as*

$$\mathsf{ADDRESS}_k(x_1 \ldots x_k y_0 \ldots y_{2^k-1}) = y_{<x_1,\ldots,x_k>}$$

where $< x_1, \ldots, x_k >$ is the unique natural number between 0 and $2^k - 1$ whose binary representation is $x_1 \ldots x_k$.

The addressing function $\mathsf{ADDRESS}_k$ has sensitivity $(k+1)$ and its randomized query complexity is $\Theta(k)$ (since randomized query complexity is at least sensitivity [16] and at most deterministic query complexity).

The definition of function composition is given below.

Definition B.4 (Composition of Boolean Functions)**.** *Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be Boolean functions. The* composition *of f and g, denoted $f \circ g$, is the Boolean function on mn variables defined as follows.*

For an input

$$x = (x_{1,1}, x_{1,2}, \ldots, x_{1,n},\ x_{2,1}, \ldots, x_{m,n}) \in \{0,1\}^{mn},$$

partition the input into m consecutive blocks:

$$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \quad \textit{for } i = 1, 2, \ldots, m.$$

Then

$$(f \circ g)(x) = f\big(g(x_1), g(x_2), \ldots, g(x_m)\big).$$

In words, we first apply g to each block of n inputs, obtaining m intermediate bits, and then apply f to these bits.

Consider the composed function of $\mathsf{ADDRESS}_k : \{0,1\}^{k+2^k} \to \{0,1\}$ and $\mathsf{AND\text{-}OR\text{-}tree}_t : \{0,1\}^t \to \{0,1\}$. We make the following observations:

Observation B.5. *The function $\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t$ is a function from $\{0,1\}^{t(k+2^k)} \to \{0,1\}$ and*

1. $\mathsf{s}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t) = O(k\sqrt{t})$
2. $\log(\mathsf{RSize}_{\frac{1}{3}}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t)) = \widetilde{\Theta}(t^{0.753\ldots})$

where the tilde hides a polynomial factor of $\log(t(k+2^k))$.

Proof. The upper bound on sensitivity is obtained from the fact that $\mathsf{s}(f \circ g) \leq \mathsf{s}(f)\mathsf{s}(g)$ [20].

The bounds on $\log \mathsf{RSize}_{\frac{1}{3}}$ follow by the fact that

$$\begin{aligned}\max\{\mathsf{RSize}_{\frac{1}{3}}(f), \mathsf{RSize}_{\frac{1}{3}}(g)\} \leq \mathsf{RSize}_{\frac{1}{3}}(f \circ g) &\leq O(\mathsf{RSize}_{\frac{1}{3}}(f) \cdot \mathsf{RSize}_{\frac{1}{m}}(g)) \\ &\leq O(\mathsf{RSize}_{\frac{1}{3}}(f) \cdot (\mathsf{RSize}_{\frac{1}{3}}(g))^{\log m}),\end{aligned}$$

where m is the arity of the function f. The first inequality follows from the fact that each query made in the randomized decision tree of f can be replaced by a randomized decision tree of g with the error made sufficiently low so that the eventual error is still low (even after union bound over the queries made on any path of the decision tree of f which is of length at most m). The second inequality follows by the standard technique of reducing the error of the randomized decision tree by repeating and then taking majority.

□

Using Observation B.5, from our theorem (Theorem 6.6) we get the following.

Corollary B.6. *Set k such that $k + 2^k = n^{2/3}$ and $t = n^{1/3}$. Then the function $\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t$ is a function from $\{0,1\}^n \to \{0,1\}$, and*

$$\log(\mathsf{DSize}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t)) = \widetilde{O}(n^{0.835\ldots}).$$

To see this, note that using Observation B.5 and Theorem 6.6 gives that

$$\log(\mathsf{DSize}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t)) = \widetilde{O}((k\sqrt{t})^2(t^{0.753\ldots})^2)$$

where the tilde hides a polynomial factor of $\log(t(k+2^k)) = \log n$.

Next we note that the upper bounds obtained by both previous methods for this function are trivial.

1. The upper bound on $\log(\mathsf{DSize}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t))$ obtained from Theorem 6.4 is $\widetilde{O}((t^{0.753\ldots})^4) = \widetilde{O}(t^{3.012\ldots})$ which gives $\widetilde{O}((n^{1/3})^{3.012\ldots}) > n$.

2. The upper bound obtained on $\log(\mathsf{DSize}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t))$ via Equation 12 is $O((k\sqrt{t})^6) > n$.

Similarly, we can also get settings where our Theorem 6.5 gives nontrivial upper bounds on DNF size, while the bounds by Equation 11 and Equation 13 are trivial.

Corollary B.7. *Set k such that $k + 2^k = n^{1/3}$ and $t = n^{2/3}$. Then the function $\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t$ is a function from $\{0,1\}^n \to \{0,1\}$, and*

$$\log(\mathsf{DNFSize}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t)) = \widetilde{O}(n^{0.835\ldots}).$$

To see this, note that using Observation B.5 and Theorem 6.5 gives that

$$\log(\mathsf{DNFSize}(\mathsf{ADDRESS}_k \circ \mathsf{AND\text{-}OR\text{-}tree}_t)) = \widetilde{O}(k\sqrt{t}(t^{0.753\ldots}))$$

where the tilde hides a polynomial factor of $\log(t(k + 2^k)) = \log n$. On the other hand, Equation 11 only gives a bound of the form $\widetilde{O}((t^{0.753\ldots})^2) = \widetilde{O}(t^{1.506\ldots})$ which gives $\widetilde{O}((n^{2/3})^{1.506\ldots}) > n$. Using Equation 13 gives $\widetilde{O}((k\sqrt{t})^5) = \widetilde{O}(n^{5/3}) > n$.

In the above discussion we compared bounds on the logarithm of decision tree and logarithm of DNF size with those that follow from previous bounds, and ignored polylog n factors. This was enough for illustrating that our results may give nontrivial bounds while previous methods would only give trivial bounds. We would like to note that translating to decision tree size and DNF size our bounds may be exponentially smaller than those that follow from previous methods, even when previous methods give nontrivial bounds.

C Number of Non-constant Sub-functions vs. Sensitivity

Consider the addressing function $\mathsf{ADDRESS}_k$ defined in Definition B.3. In this section we omit the subscript k from notation. For ease of presentation, a subset of the indices of the input will be described as a subset of $\{x_1, \ldots, x_k, y_1, \ldots, y_{2^k}\}$.

Observe that the sensitivity of the $\mathsf{ADDRESS}$ function is $(k + 1)$. But observe that for any $x \in \{0,1\}^k$ if B is any subset of $\{x_1, \ldots, x_k, y_1, \ldots, y_{2^k}\}$ such that $\{x_1, \ldots, x_k\} \subset B$ and $B \cap \{y_1, \ldots, y_{2^k}\} \neq \emptyset$, then the function $\mathsf{ADDRESS}_{B,x}$ is not constant (recall the notation $f_{B,x}$ from Section 3). This is because for some setting of the address variables $x_1, \ldots, x_k$ the function value will be the value of the variable in $B \cap \{y_1, \ldots, y_{2^k}\}$. Thus, the number of possibly overlapping ℓ-size blocks yielding non-constant sub-functions for the $\mathsf{ADDRESS}$ function is the number of subsets B of size ℓ that contain all x-variables and at least one y-variable. It is easy to see that for any $\ell > k$, the number of such sets is equal to $\sum_{i=1}^{\ell-k} \binom{2^k}{i}$.

Thus, for the $\mathsf{ADDRESS}$ function and $\ell \geq \mathsf{s}(\mathsf{ADDRESS})$, for any $x \in \{0,1\}^{k+2^k}$ the number of possibly overlapping blocks of size ℓ yielding non-constant sub-functions is as large as $2^{\Omega(k(\ell-k))}$, where $k = \mathsf{s}(\mathsf{ADDRESS}) - 1$.

Thus, unlike our main results (theorems 1.5 and 2.3) for the block number and brick number, a similar upper bound on the number of possibly overlapping blocks yielding non-constant sub-functions is not possible in terms of sensitivity. In contrast, as we noted in Section 1.1, considering *disjoint* blocks for a given input x all three notions

- the maximum number of disjoint blocks with non-constant sub-functions $f_{B,x}$
- the maximum number of disjoint bricks for x
- the maximum number of disjoint minimal sensitive blocks for x

are equal to the block sensitivity of f at x.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il