Bounds for Hardness Condensation in the Query Model

Chandrima Kayal
Université Paris Cité, CNRS, IRIF, Paris, France

Rajat Mittal
Indian Institute of Technology Kanpur, India

Sai Soumya Nalli
Microsoft Research India, Bangalore, India

Manaswi Paraashar
Indian Institute of Technology Hyderabad, India

Karthikeya Polisetty
Indian Institute of Technology Madras, India

Jayalal Sarma
Indian Institute of Technology Madras, India

Nitin Saurabh
Indian Institute of Technology Hyderabad, India

Abstract

For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ with a complexity measure having value $k \ll n$, is it possible to restrict the function f to $\Theta(k)$ variables while keeping the complexity preserved at $\Theta(k)$? Instantiation of this question for the measure of circuit complexity of the Boolean function was shown to be related to circuit lower bounds (Buresh-Oppenheim and Santhanam, 2006). Variants of the above question were also shown to have connections to the log-rank conjecture in communication complexity (Hrubeš, 2024) and lower bounds in proof complexity (Razborov, 2016). In the context of communication and query complexity, this question was recently studied by Göös, Newman, Riazanov and Sokolov (2024). They showed, among other results, that query complexity cannot be condensed losslessly.

In this work, we show that there exists a Boolean function f such that any restriction of f to $O(\mathcal{M}(f))$ variables has $\mathcal{M}(\cdot)$-complexity at most $\tilde{O}(\mathcal{M}(f)^{2/3})$, where $\mathcal{M}$ is one of block sensitivity (bs), fractional block sensitivity (fbs), certificate complexity (C), deterministic query complexity (D), zero-error randomized query complexity (R_0), and AND (and OR)-decision tree query complexity. This improves upon the results of Göös, Newman, Riazanov, and Sokolov (2024) for D and R_0, and in particular answers their open question about the condensation of block sensitivity.

We complement the negative results on lossless condensation with positive results about lossy condensation. In particular, we show that for every Boolean function f there exists a restriction of f to $O(\mathcal{M}(f))$ variables such that its $\mathcal{M}(\cdot)$-complexity is at least $\Omega(\mathcal{M}(f)^{1/2})$, where $\mathcal{M} \in \{\mathsf{bs}, \mathsf{fbs}, \mathsf{C}, \mathsf{UC}_{\min}, \mathsf{UC}_1, \mathsf{UC}, \mathsf{D}, \widetilde{\deg}, \lambda\}$. In addition, we show lossy condensation for randomized and quantum query complexity with a slightly smaller exponent.

2012 ACM Subject Classification Theory of computation → Oracles and decision trees

Keywords and phrases Query Complexity, Decision Trees, Hardness Condensation

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.30

Related Version *Full Version*: https://arxiv.org/abs/2602.00754v2

Funding *Chandrima Kayal*: Funded by French PEPR integrated project EPiQ (ANR-22-PETQ-0007) and ANR Grant FLITTLA (ANR-21-CE48-0023).
Nitin Saurabh: Supported by ANRF ARG-MATRICS grant ANRF/ARGM/2025/003191/MTR.

Acknowledgements CK and NS would like to thank Meena Mahajan and Gaurav Sood for insightful discussions during the initial stages of the work. We would also like to thank anonymous reviewers whose careful reading and thoughtful comments greatly improved the presentation of this paper. We would also like to thank the audience in STCS seminar at TIFR, Mumbai for helpful discussions.

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 30; pp. 30:1–30:20

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The problem of proving circuit lower bounds is one of the most challenging problems in computational complexity theory. Buresh-Oppenheim and Santhanam [9] introduced the concept of *hardness condensation* as a way to prove better circuit lower bounds. Their idea was to design a procedure that takes as input a function f that is hard to compute and outputs another function f', possibly on fewer input bits, such that the hardness of f' with respect to its input length is greater than that of f. In particular, they showed hardness condensation for the complexity measure circuit size. A decade later, this strategy was successfully implemented by Razborov [27] to establish strong lower bounds in proof complexity. Since then, the technique of hardness condensation has found many applications in proof complexity [28, 29, 6, 14, 10, 12, 13].

Clearly, the paradigm of hardness condensation is quite general and one could ask the following question for any complexity measure $\mathcal{M}(\cdot)$:

> Given a function f on n variables with complexity $\mathcal{M}(f)$ (a growing function of n), can we transform f explicitly into another function f' on a smaller number of variables t such that $\mathcal{M}(f') = \Theta(\mathcal{M}(f))$ while also being maximal over all functions on t variables?

The aforementioned question has also been explored recently in the context of communication complexity and matrix ranks [17, 19, 16], and query complexity [16]. In communication complexity, we are given a $2^n \times 2^n$ Boolean matrix M and the task is to solve the following two-player game: Alice is given a row $x \in [2^n]$ of M, Bob is given a column $y \in [2^n]$ of M, and their goal is to compute $M(x, y)$ while minimizing communication between them. Let $\mathsf{cc}(M)$ denote the deterministic communication complexity of this game. We say that M *condenses* to k variables if M contains a $2^k \times 2^k$ submatrix N such that $\mathsf{cc}(N) = \Theta(k)$. Further, the deterministic communication complexity *condenses losslessly* if every M condenses to $\Theta(\mathsf{cc}(M))$ variables. The hardness condensation question then asks whether $\mathsf{cc}(\cdot)$ condenses losslessly. Apart from being an interesting question about witnessing intrinsic hardness, it also has connections to the famous log-rank conjecture [24]. In particular, the log-rank conjecture implies that there exists a universal constant $\alpha \geq 1$ such that every M condenses to $\Omega(\mathsf{cc}(M)^{1/\alpha})$ variables (see also [19, 16]). In a recent work, Hrubeš [19] proved this implication unconditionally by showing that every M condenses to $\Omega(\sqrt{\mathsf{cc}(M)})$ variables.

In the context of query complexity, it was explored recently by Göös, Newman, Riazanov and Sokolov [16]. They studied whether hardness condensation can be achieved using the simple and natural transformation – *restrictions* – that is, substituting variables by constants. Typically, to reduce the number of variables, one does *variable substitutions* and/or composes the variables of f with *small gadgets* (e.g., XOR, OR, Indexing functions).

Let $\mathcal{M}$ be any decision tree based complexity measure of Boolean functions (for example, sensitivity, block sensitivity, certificate complexity, query complexity, unambiguous certificate complexity, etc). We say that a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ *condenses* to k variables *with respect to* the measure $\mathcal{M}$, if there exists a restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = k$, such that the restricted function $f|_\rho$ on k variables has large $\mathcal{M}$, i.e., $\mathcal{M}(f|_\rho) = \Theta(k)$. We further say that f *condenses losslessly* with respect to $\mathcal{M}$ if it condenses to $\Theta(\mathcal{M}(f))$ variables. We also say that $\mathcal{M}$ condenses losslessly if every function f condenses losslessly w.r.t. $\mathcal{M}$.

For example, consider the measure of query complexity of Boolean functions, which is studied by [16]. For $f\colon \{0,1\}^n \to \{0,1\}$, the query complexity $\mathsf{D}(f)$ of f is the minimum number of queries made by a deterministic decision tree algorithm computing f in the worst-case over all inputs $x \in \{0,1\}^n$. Following the above definition (also from [16]), we say that f condenses to k variables with respect to the measure of deterministic query

complexity if there exists a partial assignment $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = k$, such that the restricted function $f|_\rho$ on k variables has large query complexity, i.e., $\mathsf{D}(f|_\rho) = \Theta(k)$. The hardness condensation question, w.r.t. $\mathsf{D}(\cdot)$, then asks if $\mathsf{D}(\cdot)$ condenses losslessly.

The authors of [16] answered the aforementioned question in the negative. That is, they showed a function f such that for every restriction ρ that fixes all but at most $O(\mathsf{D}(f))$ variables, the query complexity $\mathsf{D}(f|_\rho)$ of the restricted function $f|_\rho$ is $\tilde{O}(\mathsf{D}(f)^{3/4})$ (where $\tilde{O}$ hides polylog factors in $\mathsf{D}(f)$). In other words, deterministic (and even randomized) query complexity cannot be condensed losslessly in general. The authors wondered if there is a function that witnesses greater loss, i.e., for which the hardness is even less condensible. Observe that the best possible bound cannot be better than $O(\mathsf{D}(f)^{1/3})$. Indeed, it follows from the fact that every f condenses to $\deg(f)$ variables[1] and $\deg(f) \geq \mathsf{D}(f)^{1/3}$ [25]. In [16], the authors left open, among other things, the problem of closing this gap and verifying lossless condensation for other decision tree based measures like, block sensitivity, certificate complexity, unambiguous certificate complexity, etc.

In this article, we delve deeper into the question of hardness condensation for $\mathsf{D}(\cdot)$ and initiate the study of hardness condensation for almost all other decision tree based measures. In particular, we close the gap for D from both sides, showing that it is not possible to condense even with an exponent of $2/3 + \varepsilon$ (improving from $3/4$) and it is always possible to condense with an exponent of $1/2$ (improving from $1/3$). We show similar results for many other well-known complexity measures like block sensitivity, certificate complexity, fractional block sensitivity, AND-decision tree complexity, etc. (Theorems 1, 2, 3 and 5).

Unfortunately, except for approximate degree and spectral sensitivity, there are gaps in terms of what can and cannot be achieved for these complexity measures in terms of hardness condensation (see Table 1). These gaps leave many intriguing questions open; we hope that future research will be able to resolve these questions.

Before going into the details of our results and proof techniques, we present some possible applications of hardness condensation by restrictions.

Applications of Hardness Condensation by Restrictions

Log-Rank Conjecture. We begin with an application to the log-rank conjecture highlighted by [18]. Let the AND-decision tree query complexity $\mathsf{D}_\wedge(f)$ of a Boolean function f be the minimum number of AND queries to the input made by a deterministic query algorithm computing f in the worst case (see Definition 14).

For any $f\colon \{0,1\}^n \to \{0,1\}$, let $\mathsf{mon}(f)$ denote the number of monomials with non-zero coefficients in the unique multilinear polynomial representing f. Further, define the communication problem $f_\wedge\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as follows $f_\wedge(x,y) = f(x_1 \wedge y_1, \ldots, x_n \wedge y_n)$. It then follows that the rank of the communication matrix of $f_\wedge$ equals $\mathsf{mon}(f)$ [7]. Moreover, it is known that

$$\mathsf{cc}(f_\wedge) \leq 2\mathsf{D}_\wedge(f) = O(\log^5 \mathsf{mon}(f) \cdot \log n) = O(\log^5 \mathsf{rank}(f_\wedge) \cdot \log n),$$

where the second inequality was shown by [22]. That is, the log-rank conjecture holds for $f_\wedge$ as long as $\log\log \mathsf{mon}(f) = \Omega(\log\log n)$. In other words, the log-rank conjecture remains unsolved for $f_\wedge$ when f has a very sparse representation as a polynomial. To resolve this, [18] observed that a hardness condensation of the following form suffices.

[1] Consider a monomial $\mathfrak{m}$ of maximum degree $\deg(f)$ in the unique real polynomial representing f. Let ρ be a restriction that fixes all variables except the ones in $\mathfrak{m}$. Then $\mathsf{D}(f|_\rho) = \deg(f|_\rho) = \deg(f)$.

▶ **Conjecture** ([18]). *There exist universal constants $\alpha, \beta > 0$ such that for every $f\colon \{0,1\}^n \to \{0,1\}$ there exists a restriction ρ with $|\rho^{-1}(*)| = 2^{O(\log^{\alpha} \mathsf{mon}(f))}$ and $\mathsf{D}_{\wedge}(f|_\rho) = \Omega(\mathsf{D}_{\wedge}(f)^{\beta})$.*

Separations between complexity measures. We now illustrate an application of hardness condensation to witness improved separations between complexity measures. We illustrate the argument with a simple example to highlight the applicability of this approach.

Consider the modified Rubinstein function f given in Definition 23. It can be easily shown that $\mathsf{D}(f) = k^2$ and $\mathsf{bs}(f) = k^{1.5}$ (see also Lemma 26). Thus we have an example where $\mathsf{D}(f) = \Omega(\mathsf{bs}(f)^{4/3})$ and $\mathsf{bs}(f) = \Omega(\mathsf{s}(f)^{3/2})$. Using hardness condensation, we now amplify the gap between D and bs. We know from Theorem 1 that f witnesses the incondensability of bs with exponent $2/3$. That is, for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(\mathsf{bs}(f))$ we have $\mathsf{bs}(f|_\rho) = O(k) = O(\mathsf{bs}(f)^{2/3})$. Now consider a restriction γ with $|\gamma^{-1}(*)| = \Theta(\mathsf{bs}(f))$ such that $\mathsf{D}(f|_\gamma) = |\gamma^{-1}(*)| = \Theta(\mathsf{bs}(f))$. Such a restriction is easily seen to exist since $\mathsf{D}(f)$ is full to begin with. We then have by Theorem 1 that $\mathsf{bs}(f|_\gamma) = O(k) = O(\mathsf{bs}(f)^{2/3})$. We therefore have obtained a function $h := f|_\gamma$ such that $\mathsf{D}(h) = \Omega(\mathsf{bs}(h)^{3/2})$ improving over $4/3$ separation witnessed by f. In general, such an approach can be used to achieve better separations.

A candidate application of the approach outlined above could be the gap between D and R. The best known separation between D and R is quadratic [3]. Note that we know of a function that witness more than a quadratic gap between R and s [5]. The approach outlined above exploits this gap with respect to sensitivity to obtain improved separations between complexity measures.

We further note that the positive condensation result in Theorem 5 also shows a limit to amplification using the approach outlined above. For example, we have shown in Theorem 5 that for every f there exists a restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(\mathsf{bs}(f))$ such that $\mathsf{bs}(f|_\rho) = \Omega(\sqrt{\mathsf{bs}(f)})$. Therefore, our approach cannot show a gap of more than $1/2$ between D and bs.

1.1 Our Results

In this work, we further explore the phenomenon of hardness condensation in the setting of query complexity and other Boolean function parameters using restrictions. We begin by recalling that sensitivity and degree are two such measures that condense losslessly. As mentioned above, [16] shows that deterministic query complexity cannot be condensed losslessly. Our first result (proved in Section 3) shows that block sensitivity, fractional block sensitivity and certificate complexity cannot be condensed losslessly.

▶ **Theorem 1.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\mathsf{bs}(f) = \mathsf{fbs}(f) = \mathsf{C}(f) = n^{3/4}$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/4})$ we have $\mathsf{bs}(f|_\rho) \leq \mathsf{fbs}(f|_\rho) \leq \mathsf{C}(f|_\rho) = O(\sqrt{n}) = O(\mathsf{bs}(f)^{2/3})$.*

Our second result (proved in Section 4) shows that deterministic query complexity is even less condensible than shown in [16]. We also show that AND-decision tree complexity and 0-decision tree complexity cannot be condensed losslessly.

▶ **Theorem 2.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_0(f) \leq \mathsf{D}_{\wedge}(f) \leq \mathsf{D}(f) = O(n^{3/5})$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/5})$ we have $\mathsf{D}_0(f|_\rho) \leq \mathsf{D}_{\wedge}(f|_\rho) \leq \mathsf{D}(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_0(f)^{2/3})$.*

Our proof technique for Theorem 2 works even in the dual setting and therefore we also obtain the following incondensability of OR-decision tree complexity.

▶ **Theorem 3.** *There exists a Boolean function* $f\colon \{0,1\}^n \to \{0,1\}$ *with* $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_1(f) \leq \mathsf{D}_\vee(f) = O(n^{3/5})$*, such that for every restriction* $\rho\colon [n] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(n^{3/5})$ *we have* $\mathsf{D}_1(f|_\rho) \leq \mathsf{D}_\vee(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_1(f)^{2/3})$.

We further observe that the zero-error randomized query complexity of the example in Theorem 2 is also high $\tilde{\Omega}(n^{3/5})$, thereby obtaining a similar incondensability result for R_0. For a proof of the following Corollary, we refer to the full version.

▶ **Corollary 4.** *There exists a Boolean function* $f\colon \{0,1\}^n \to \{0,1\}$ *with* $\mathsf{R}_0(f) = \tilde{\Theta}(n^{3/5})$*, such that for every restriction* $\rho\colon [n] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(\mathsf{R}_0(f))$ *we have* $\mathsf{R}_0(f) = \tilde{O}(\mathsf{R}_0(f)^{2/3})$.

We note that Theorems 1 and 2 answer (or, partially answer) Problems 1 and 2 in [16]. Theorem 2 also improves upon (one of) the results in [16]. It is natural to wonder if the gaps in Theorems 1 and 2 are quantitatively optimal. That is, does there exist a function for which a measure is even less condensible than established here?

In particular, for deterministic query complexity, it asks if there exists a function f such that for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f))$, we have $\mathsf{D}(f|_\rho) = O(\mathsf{D}(f)^\alpha)$ where $\alpha < 2/3$? Observe that $\alpha \geq 1/3$, since $\mathsf{D}(f) \leq \deg(f)^3$ [25] and $\deg(f)$ condenses losslessly. Similarly, for block sensitivity, the exponent $\alpha \geq 1/4$, since $\mathsf{bs}(f) \leq \mathsf{s}(f)^4$ [20] and $\mathsf{s}(f)$ condenses losslessly.

Observe that the lower bound on the exponent α can be improved if conditionally based on other relationships between the parameters. For example, if $\mathsf{D}(f) \leq \deg(f)^2$, or $\mathsf{bs}(f) \leq \mathsf{s}(f)^2$, then such an improvement is possible. Our third result (proved in Section 5) establishes the same lower bound on the exponent α unconditionally.

▶ **Theorem 5.** *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a Boolean function and* $\mathcal{M} \in \{\mathsf{bs}, \mathsf{fbs}, \mathsf{C}, \mathsf{UC}_{\min}, \mathsf{UC}_1, \mathsf{UC}, \mathsf{D}, \widetilde{\deg}, \lambda, \mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$ *be a complexity measure. Then, the following holds:*

(a) *For* $\mathcal{M} \notin \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$*, there exists a restriction* $\rho \in \{0,1,*\}^n$ *with* $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ *such that* $\mathcal{M}(f|_\rho) = \Omega(\sqrt{\mathcal{M}(f)})$.

(b) *For* $\mathcal{M} \in \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}_E\}$*, there exists a restriction* $\rho \in \{0,1,*\}^n$ *with* $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ *such that* $\mathcal{M}(f|_\rho) = \Omega(\mathcal{M}(f)^{1/3})$.

(c) *For* $\mathcal{M} = \mathsf{Q}$*, there exists a restriction* $\rho \in \{0,1,*\}^n$ *with* $|\rho^{-1}(*)| = \Theta(\mathsf{Q}(f))$ *such that* $\mathsf{Q}(f|_\rho) = \Omega(\mathsf{Q}(f)^{1/4})$.

Table 1 lists the known bounds on condensation for many decision tree measures. Note that for approximate degree ($\widetilde{\deg}$) and spectral sensitivity (λ), we have tight bounds on lossy condensation that can be achieved. For the other measures, there is a gap in the negative result and the positive result. Thus, the understanding for many of them is not complete, leaving many interesting open questions.

1.2 Proof Ideas

In this subsection, we explain the ideas and techniques used to obtain our results.

Incondensability of bs and C. Observe that since $\mathsf{s} \leq \mathsf{bs} \leq \mathsf{C}$ and sensitivity condenses losselessly, a counter-example $f\colon \{0,1\}^n \to \{0,1\}$ to lossless condensation of block sensitivity must witness a gap between bs and s, i.e., $\mathsf{bs}(f) = \omega(\mathsf{s}(f))$. Simultaneously, there should also be a gap between the number of variables n and $\mathsf{bs}(f)$, so that every restriction ρ with $|\rho^{-1}(*)| = O(\mathsf{bs}(f))$ must set some variables to constants. Thus the Rubinstein function (and rather a modification of it to introduce a gap between n and bs) becomes a natural choice.

Table 1 Hardness condensation by variable restrictions.

- A "$*$" entry : we do not know whether the measure labeling the row condenses losslessly or not.
- An entry γ in the Negative Result column (labelled by measure $\mathcal{M}$) : $\exists f$ such that for every restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = O(\mathcal{M}(f))$ we have $\mathcal{M}(f|_\rho) = O(\mathcal{M}(f)^\gamma)$.
- An entry ξ in the Positive Result column (labelled by measure $\mathcal{M}$) : $\forall f$ there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ such that $\mathcal{M}(f|_\rho) = \Omega(\mathcal{M}(f)^\xi)$.

Complexity Measure	Negative Result	Positive Result
D	2/3 (Theorem 2)	1/2 (Theorem 5)
D_0, D_1, $\mathsf{D}_\wedge$, $\mathsf{D}_\vee$	2/3 (Theorems 2 & 3)	$*$
R_0	2/3 (Corollary 4)	1/3 (Theorem 5)
R	3/4 ([16])	1/3 (Theorem 5)
$\mathsf{C}, \mathsf{fbs}, \mathsf{bs}$	2/3 (Theorem 1)	1/2 (Theorem 5)
$\mathsf{UC}, \mathsf{UC}_{\min}$	$*$	1/2 (Theorem 5)
Q_E	$*$	1/3 (Theorem 5)
Q	1/2 [OR_n]	1/4 (Theorem 5)
$\widetilde{\deg}$, λ	1/2 [OR_n]	1/2 (Theorem 5)
$\mathsf{s}, \deg$	–	condenses losslessly (see also Lemma 36)

We then observe that the Rubinstein function also has a nice property that only a particular kind of inputs (all 0's) have high block sensitivity and certificate complexity. We use this structure to argue that the certificate complexity, and in turn the block sensitivity, must reduce after restriction.

Incondensability of D and its variants. Incondensability of D was shown by [16] using a cheat-sheet version of Tribes (AND of ORs). Our improved example is also a cheat-sheet version f_{CS} (Equation (1)) of Tribes f (Definition 28). However, we choose different arities for AND and ORs. In particular, we choose $f = \mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}}$. This turns out to be a crucial insight that helps design an improved query algorithm for $f_{\mathrm{CS}}|_\rho$; thereby obtaining an improvement.

Our query algorithm for $f_{\mathrm{CS}}|_\rho$, like in [16], has two parts. In the first part our algorithm finds a string $z \in \{0,1\}^{\log t}$, where t is the number of cheat-sheet cells, that the address part $f(x_1|_\rho)f(x_2|_\rho)\cdots f(x_{\log t}|_\rho)$ can take if the restricted function $f_{\mathrm{CS}}|_\rho$ were to evaluate to 1. In the second part, the algorithm simply reads the cell pointed to by z and verifies it.

In the first part, the algorithm makes queries with an aim to find each bit of $z = z_1 z_2 \cdots z_{\log t} \in \{0,1\}^{\log t}$, while satisfying the following property: either we know the value of $z_j = f(x_j|_\rho)$ correctly or there are no valid cheat-sheet cells with $z_j = 1$. For each $j \in [\log t]$, the query algorithm maintains the following two sets: (a) the set $\mathcal{O}_0$ of ORs in the j-th copy of f that can possibly evaluate to 0, and (b) the set $\mathcal{V}_1$ of cheat-sheet cells with $z_j = 1$ and not found to be inconsistent with the queries made so far. Note that initially $|\mathcal{O}_0| = k$ and $|\mathcal{V}_1| = t/2$. Now the algorithm makes queries in such a way that each query reduces either $|\mathcal{O}_0|$ by 1 or $|\mathcal{V}_1|$ by a factor of $(1 - \frac{1}{2\sqrt{k}})$. Finally when the algorithm stops, after making at most $O(k + \sqrt{k}\log t)$ queries, we either know the value of $f(x_j|_\rho)$ or more crucially are guaranteed to be in one of the following two cases: (i) $|\mathcal{O}_0| = O(\sqrt{k})$ or (ii) $|\mathcal{V}_1| = O(k)$. In case (i) the algorithm can query all remaining ORs in $\mathcal{O}_0$ to find out the value of $f(x_j|_\rho)$, while in case (ii) it uses $O(\log k)$ queries to find inconsistency to remove a cell from $\mathcal{V}_1$ (Claim 31), and finally the last remaining cheat-sheet cell in $\mathcal{V}_1$ is verified for 1-certificate that it claims to contain for $f(x_j|_\rho)$.

To find a query that allows us to reduce either $|\mathcal{O}_0|$ by 1 or $|\mathcal{V}_1|$ by a factor of $(1 - \frac{1}{2\sqrt{k}})$, we use the fact that $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}}) = \tilde{O}(k^{1.5})$ and thus there must be a claimed 1-certificate that is completely revealed in at least half the fraction of $\mathcal{V}_1$.

Lossy condensation. We complement our negative results on lossless condensation with positive results about lossy condensation. Our proofs follow an analysis of different cases based on whether $\mathsf{s}(f)$ is large or $\deg(f)$ is large or neither. In each case, we find a restriction that gives the desired lossy condensation. In the first two cases, since sensitivity and degree condense losslessly (Lemma 36), we easily find a restriction witnessing lossy condensation. The last and the interesting case is when neither of them is large; we then argue that it must be the case that block sensitivity is large, which then helps us find a restriction that witnesses lossy condensation. For example, consider the case of the deterministic query complexity. As alluded to before, the interesting case is when neither sensitivity nor degree is large, i.e., at least $\sqrt{\mathsf{D}(f)}$. Then, from $\mathsf{D}(f) \leq \deg(f)\mathsf{bs}(f)$ [25], it follows that $\mathsf{bs}(f) \geq \sqrt{\mathsf{D}(f)}$. Thus, we can consider the following restriction ρ based on an input z with the maximum block sensitivity: leave $\sqrt{\mathsf{D}(f)}$ disjoint (minimal) sensitive blocks unset while setting the rest of the variables according to z. Since $\mathsf{s}(f) < \sqrt{\mathsf{D}(f)}$, $|\rho^{-1}(*)| = O(\mathsf{D}(f))$. At the same time, by construction, we also have $\mathsf{D}(f|_\rho) \geq \mathsf{bs}(f|_\rho) \geq \sqrt{\mathsf{D}(f)}$.

2 Preliminaries

Notations: We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

We recall the definitions of the complexity measures we will use throughout. We refer the reader to the survey [8] for an introduction to the complexity of Boolean functions and complexity measures. Several additional complexity measures and their relationships can also be found in [5] and [2].

In the following $f\colon \{0,1\}^n \to \{0,1\}$ is a Boolean function on n variables. For an input $x \in \{0,1\}^n$ and $i \in [n]$, let x^i represent the string in $\{0,1\}^n$ that is obtained from x by flipping (or, negating) the i-th bit. We say that the i-th bit is *sensitive* at x if $f(x) \neq f(x^i)$.

▶ **Definition 6** (Sensitivity). *The* sensitivity *of f on an input x, $\mathsf{s}(f,x)$, is defined as the number of sensitive bits at x with respect to f. Then, the* sensitivity *of f, denoted $\mathsf{s}(f)$, is defined as $\mathsf{s}(f) = \max\{\mathsf{s}(f,x) \mid x \in \{0,1\}^n\}$.*

We also define 0-sensitivity *of f as $\mathsf{s}_0(f) = \max\{\mathsf{s}(f,x) \mid x \in \{0,1\}^n, f(x) = 0\}$ and* 1-sensitivity *of f as $\mathsf{s}_1(f) = \max\{\mathsf{s}(f,x) \mid x \in \{0,1\}^n, f(x) = 1\}$.*

Similarly for $B \subseteq [n]$, let x^B denote the string obtained from x by flipping the bits indexed by the block B. We say that the block B is *sensitive* at x if $f(x) \neq f(x^B)$.

▶ **Definition 7** (Block Sensitivity). *The* block sensitivity *of a function f on an input x, $\mathsf{bs}(f,x)$, is the maximum number of disjoint subsets $B_1, B_2, \ldots, B_r$ of $[n]$ such that for all $j \in [r]$, $f(x) \neq f(x^{B_j})$. Then, the* block sensitivity *of f, denoted $\mathsf{bs}(f)$, is $\max\{\mathsf{bs}(f,x) \mid x \in \{0,1\}^n\}$. Similarly, define $\mathsf{bs}_0(f) = \max\{\mathsf{bs}(f,x) \mid x \in \{0,1\}^n, f(x) = 0\}$ and $\mathsf{bs}_1(f) = \max\{\mathsf{bs}(f,x) \mid x \in \{0,1\}^n, f(x) = 1\}$.*

A sensitive block at x is called *minimal* if no proper subset of it as a sensitive block at x.

We also need the following generalization of block sensitivity based on linear programming.

▶ **Definition 8** (Fractional Block Sensitivity). *Let* $\mathcal{W}(f,x) := \{B \subseteq [n] \mid f(x^B) \neq f(x)\}$ *denote the set of all sensitive blocks for an input* $x \in \{0,1\}^n$*. The* fractional block sensitivity *of* f *at* x*, denoted* $\mathsf{fbs}(f,x)$*, is the value of the linear program,*

$$\begin{aligned} \textit{maximize} \quad & \sum_{B \in \mathcal{W}(f,x)} y_B, \\ \textit{subject to} \quad & \sum_{B\colon i \in B} y_B \leq 1, \quad \forall\, i \in [n], \\ & 0 \leq y_B \leq 1, \quad \forall\, B \in \mathcal{W}(f,x). \end{aligned}$$

The fractional block sensitivity $\mathsf{fbs}(f)$ *of* f *is then defined as* $\mathsf{fbs}(f) := \max_{x \in \{0,1\}^n} \mathsf{fbs}(f,x)$.

A *partial assignment* is a function $\rho\colon \{0,1\}^n \to \{0,1,*\}$ where the size of ρ is $n - |\rho^{-1}(*)|$. Let $S \subseteq [n]$ denote the set of indices where ρ is not $*$. We say that an input $x \in \{0,1\}^n$ is *consistent* with ρ iff $x_i = \rho(i)$ for all $i \in S$. For $b \in \{0,1\}$, a *b-certificate* for the Boolean function f is a partial assignment ρ such that for all inputs x and y consistent with ρ we have $f(x) = f(y) = b$.

▶ **Definition 9** (Certificate Complexity). *The* certificate complexity *of a function* f *on an input* x*, denoted* $\mathsf{C}(x,f)$*, is the size of the smallest* $f(x)$*-certificate that is consistent with* x*. The* certificate complexity *of* f*, denoted* $\mathsf{C}(f)$*, is* $\max_x\{\mathsf{C}(f,x)\}$*. Further, define* $\mathsf{C}_0(f) = \max_{x\colon f(x)=0}\{\mathsf{C}(f,x)\}$ *and* $\mathsf{C}_1(f) = \max_{x\colon f(x)=1}\{\mathsf{C}(f,x)\}$.

We now note the following relation between these measures.

▶ **Lemma 10** ([26, 31]). $s(f,x) \leq \mathsf{bs}(f,x) \leq \mathsf{fbs}(f,x) \leq \mathsf{C}(f,x) \leq \mathsf{bs}(f,x)\mathsf{s}(f)$.

We also require the following generalization of certificate complexity.

▶ **Definition 11** (Unambiguous Certificate Complexity). *Fix* $b \in \{0,1\}$*. A set* U *of partial assignments is said to form an* unambiguous *collection of b-certificates for* f *if*

- *each partial assignment in* U *is a b-certificate for* f*,*
- *for each* $x \in f^{-1}(b)$ *there is some* $p \in U$ *that is consistent with* x*, and*
- *no two partial assignments in* U *are consistent.*

We then define $\mathsf{UC}_b(f)$ *to be the minimum value of* $\max_{p\in U}|p|$ *over all choices of such collections* U*. Further, define* $\mathsf{UC}(f) := \max\{\mathsf{UC}_0(f), \mathsf{UC}_1(f)\}$ *and* $\mathsf{UC}_{\min}(f) := \min\{\mathsf{UC}_0(f), \mathsf{UC}_1(f)\}$.

We now define the query model of computation. In this model, the input bits can be accessed by queries to an input oracle and the complexity of computing a Boolean function f is the number of queries made to the oracle.

▶ **Definition 12** (Deterministic Decision Trees). *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a Boolean function. Let* A *be a deterministic algorithm that computes* f *by making queries to the bits of an unknown input* x*. The deterministic decision tree complexity (also known as query complexity) of* f*, denoted* $\mathsf{D}(f)$*, is* $\min_A\{\max_x \mathit{cost}(A,x)\}$*, where the minimum is over all deterministic algorithm computing* f *and* $\mathit{cost}(A,x)$ *is the number of queries made by* A *on the input* x*.*

We note the following relations, which are easy to observe.

▶ **Fact 13.** $\deg(f) \leq \mathsf{UC}_{\min}(f) \leq \mathsf{UC}_1(f) \leq \mathsf{UC}(f) \leq \mathsf{D}(f)$*, and* $\mathsf{bs}(f) \leq \mathsf{C}(f) \leq \mathsf{UC}(f)$.

We also need the following variants of the deterministic decision tree.

▶ **Definition 14** (AND-Decision Trees). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let A be a deterministic algorithm that computes f by querying* AND *of a subset $S \subseteq [n]$ of the bits of an unknown input x. Further, let $cost(A,x)$ be the number of queries made by A on the input x, and $cost(A) = \max_x cost(A,x)$. Then, the* AND*-decision tree complexity of f, denoted $\mathsf{D}_\wedge(f)$, is $\min_A\{cost(A)\}$, where the minimum is over all deterministic* AND*-decision tree algorithm computing f.*

We can similarly define OR-decision trees, which are allowed to query OR of a subset of input variables, and OR-decision tree complexity ($\mathsf{D}_\vee(f)$). We also need the notion of decision tree algorithms that minimize the number of 0-queries (or, 1-queries) while computing f.

▶ **Definition 15** (0-Decision Trees). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let A be a deterministic algorithm that computes f by making queries to the bits of an unknown input x. Further, let $cost(A,x)$ be the number of queries made by A on the input x that are answered as 0, and $cost(A) = \max_x cost(A,x)$. Then, the deterministic 0-decision tree complexity of f, denoted $\mathsf{D}_0(f)$, is $\min_A\{cost(A)\}$, where the minimum is over all deterministic 0-decision tree algorithm computing f.*

Similarly, we can define deterministic 1-decision tree complexity ($\mathsf{D}_1(f)$). The following relations between these variants (see also [23]) are easy to observe.

▶ **Fact 16.** *$\mathsf{D}_0(f) \le \mathsf{D}_\wedge(f) \le \mathsf{D}(f)$, and $\mathsf{D}_1(f) \le \mathsf{D}_\vee(f) \le \mathsf{D}(f)$.*

For the definitions of well-studied complexity measures, such as randomized decision trees (R), zero-error randomized decision tree (R_0) , exact quantum query complexity (Q_E) and bounded error quantum query complexity (Q), we refer the reader to [8].

We note the following relations that are easy to observe.

▶ **Fact 17.** *$\mathsf{Q}(f) \le \mathsf{R}(f) \le \mathsf{R}_0(f) \le \mathsf{D}(f)$, and $\mathsf{Q}(f) \le \mathsf{Q}_E(f) \le \mathsf{D}(f)$.*

A real multilinear polynomial $p(x_1,\ldots,x_n)$ is said to *represent* a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, if $p(x) = f(x)$ for all $x \in \{0,1\}^n$. It is easily seen that a Boolean function f has a unique multilinear polynomial p representing it. Let $\deg(f)$ denote the degree of this unique multilinear polynomial representing f. We also need the following notion of approximating polynomials.

▶ **Definition 18.** *We say that a polynomial p* approximates *a Boolean function f, if $|p(x) - f(x)| \le 1/3$ for all $x \in \{0,1\}^n$. Let $\widetilde{\deg}(f)$ denote the minimal possible degree of a real polynomial approximating f.*

We again note relations that are easy to observe.

▶ **Fact 19.** *$\widetilde{\deg}(f) \le \deg(f) \le \mathsf{D}(f)$, and $\widetilde{\deg}(f) \le \mathsf{R}(f)$.*

We also need the following measure that played the key role in the proof of the sensitivity conjecture [20].

▶ **Definition 20.** *The* sensitivity graph *of f, $G_f = (V,E)$, is the graph where $V = \{0,1\}^n$ and $E = \{(x,x^i) \mid f(x) \ne f(x^i), i \in [n], x \in V\}$. Let A_f be the adjacency matrix of the graph G_f. Then, the* spectral sensitivity *of f, denoted $\lambda(f)$, is the spectral norm $\|A_f\|$ of the adjacency matrix A_f.*

We also describe the (disjoint) composition of two Boolean functions.

▶ **Definition 21.** *For any two Boolean functions* $f\colon \{0,1\}^n \to \{0,1\}$ *and* $g\colon \{0,1\}^m \to \{0,1\}$, *define the* composed *function* $f \circ g\colon \{0,1\}^{nm} \to \{0,1\}$ *as follows*

$$f \circ g(x_{11}, \ldots, x_{1m}, \ldots\ldots, x_{n1}, \ldots, x_{nm}) = f(g(x_1), \ldots, g(x_n)),$$

where $x_i = (x_{i1}, \ldots, x_{im}) \in \{0,1\}^m$ *for* $i \in [n]$.

▶ **Definition 22** (Rubinstein's function [30]). *Let* $g\colon \{0,1\}^k \to \{0,1\}$ *be such that* $g(x) = 1$ *iff* x *contains two consecutive ones and the rest of the bits are zeroes. The Rubinstein's function, denoted* $\mathsf{RUB}\colon \{0,1\}^{k^2} \to \{0,1\}$, *is defined to be* $\mathsf{RUB} = \mathsf{OR}_k \circ g$.

We also define a modified version of the Rubinstein function, which will be the witness function for our negative results.

▶ **Definition 23** (Modified Rubinstein function). *Define* $g\colon \{0,1\}^k \to \{0,1\}$ *to be* 1 *iff the input contains* $\sqrt{k}$ *consecutive ones and the rest of the bits are zeroes. Then, define* $f\colon \{0,1\}^{k^2} \to \{0,1\}$ *to be the function* $(\mathsf{OR}_k \circ g)(x)$ *where* $x \in \{0,1\}^{k^2}$.

Our negative results for some of the query measures require cheatsheet functions.

▶ **Definition 24** (Cheat-sheet version of a function). *Fix* $t > 0$. *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a function such that the* description *of any (minimal) certificate takes at most* m *bits. We then define*

$$f_{\mathrm{CS}}\colon \{0,1\}^{n \log t + mt \log t} \to \{0,1\}$$

the cheat-sheet version of f *as follows. The input to* f_{CS} *is* $X = (x_1, \ldots, x_{\log t}, Y_0, \ldots, Y_{t-1})$ *where* $x_i \in \{0,1\}^n$ *are inputs to* $\log t$ *copies of* f, *and* $Y_j \in \{0,1\}^{m \log t}$ *are cheat-sheet cells.*

Define $f_{\mathrm{CS}}(X) = 1$ *iff the cheat-sheet cell* Y_ℓ, *given by the positive integer* ℓ *corresponding to the binary string* $f(x_1)f(x_2)\cdots f(x_{\log t})$, *describes* $\log t$ *certificates, one for each* x_i *using* m *bits. That is, it contains a* description *of* $f(x_i)$*-certificate for each* x_i.

For more details on the cheat-sheet framework, we refer to [1, 4].

3 Negative Results I: Incondensability of Block Sensitivity and Certificate Complexity

In this section, we prove Theorem 1, i.e., block sensitivity, fractional block sensitivity and certificate complexity do not condense losslessly.

▶ **Theorem 1.** *There exists a Boolean function* $f\colon \{0,1\}^n \to \{0,1\}$ *with* $\mathsf{bs}(f) = \mathsf{fbs}(f) = \mathsf{C}(f) = n^{3/4}$, *such that for every restriction* $\rho\colon [n] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(n^{3/4})$ *we have* $\mathsf{bs}(f|_\rho) \le \mathsf{fbs}(f|_\rho) \le \mathsf{C}(f|_\rho) = O(\sqrt{n}) = O(\mathsf{bs}(f)^{2/3})$.

In fact, we will use the same function, Modified Rubinstein function (Definition 23), for all three measures. More formally, we establish the following bounds for its complexity measures.

▶ **Theorem 25.** *There exists a Boolean function* $f\colon \{0,1\}^{k^2} \to \{0,1\}$ *with* $\mathsf{bs}(f) = \mathsf{fbs}(f) = \mathsf{C}(f) = k^{1.5}$, *such that for every partial assignment* $\rho\colon [k^2] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(k^{1.5})$ *we have* $\mathsf{bs}(f|_\rho) \le \mathsf{fbs}(f|_\rho) \le \mathsf{C}(f|_\rho) = O(k)$.

Proof. Since for every f and input x, $\mathsf{bs}(f,x) \le \mathsf{fbs}(f,x) \le \mathsf{C}(f,x)$, it suffices to show the following two properties hold,

- **(i)** $\mathsf{bs}(f) = \mathsf{C}(f) = k^{1.5}$, and
- **(ii)** for every partial assignment $\rho\colon [k^2] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(k^{1.5})$ we have $\mathsf{C}(f|_\rho) = O(k)$.

Let f be the modified Rubinstein function as given in Definition 23. Then, Item i follows from Lemma 26 and Item ii is shown in Lemma 27. Thus, the theorem follows. ◀

We now prove the two lemmas to complete the proof.

▶ **Lemma 26.** *Let f be the function defined in Definition 23. Then, $\mathsf{C}_0(f) = k^{1.5}$, $\mathsf{bs}_0(f) = k^{1.5}$, $\mathsf{bs}_1(f) = k$, and $\mathsf{C}_1(f) = k$.*

Proof. Recall the k^2 variate function $f = \mathsf{OR}_k \circ g$, where $g\colon \{0,1\}^k \to \{0,1\}$ evaluates to 1 iff the input to g contains $\sqrt{k}$ consecutive ones and the rest of the bits are zeroes. Let $x = (x_{11}, x_{12}, \ldots, x_{1k}, x_{21}, \ldots, x_{2k}, \ldots, x_{k1}, \ldots, x_{kk})$ be an input to f. For $1 \le i \le k$, the i-th copy of g is defined over the set of variables $\{x_{i1}, \ldots, x_{ik}\}$.

Clearly, $\mathsf{C}_1(f) = \mathsf{C}_1(g) \le k$ and $\mathsf{bs}_1(f) = \mathsf{bs}_1(g) \ge k$, which in turn implies that both are equal to k. Note a 0-certificate for f is a collection of 0-certificates, one for each copy of g. Thus, $\mathsf{C}_0(f) \le k \cdot \mathsf{C}_0(g)$. We now upper bound $\mathsf{C}_0(g)$. A 0-input $y \in \{0,1\}^k$ of g falls under (at least) one of the following types:

(1) It contains two 1-bits that are at least $\sqrt{k}$ apart. That is, there exists $i, j \in [k]$ such that $y_i = y_j = 1$ and $|i - j| \ge \sqrt{k}$. For such 0-inputs, the certificate complexity is 2.

(2) It contains three input bits y_i, y_j, y_ℓ such that $i < j < \ell$ and $y_i = 1, y_j = 0$, and $y_\ell = 1$. For such 0-inputs, the certificate complexity is at most 3.

(3) It either (I) contains the substring $01^\ell 0$, or (II) equals to $0^{k-\ell}1^\ell$ or $1^\ell 0^{k-\ell}$ for some ℓ such that $1 \le \ell < \sqrt{k}$. For 0-inputs of kind (I), the certificate complexity is at most 3, namely the first two bits (01) and the last bit (0) of the substring suffice as a 0-certificate. For 0-inputs of kind (II), just the two bits on the boundary suffices, $y_{k-\ell} = 0$ and $y_{k-\ell+1} = 1$ (or, $y_\ell = 1$ and $y_{\ell+1} = 0$). Therefore, the certificate complexity is again at most 3.

(4) The input $y = 0^k$. The certificate complexity for this input is $\sqrt{k}$.

We therefore have $\mathsf{C}_0(f) \le k \cdot \mathsf{C}_0(g) \le k^{1.5}$.

It is now sufficient to show that $\mathsf{bs}_0(f) \ge k^{1.5}$. Consider the input 0^{k^2}. It is easily seen that $\mathsf{bs}(f, 0^{k^2}) = k \cdot \mathsf{bs}(g, 0^k) = k\sqrt{k}$. ◀

▶ **Lemma 27.** *Let f be the function defined in Definition 23. Then, for every partial assignment $\rho\colon [k^2] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(k^{1.5})$, we have $\mathsf{C}(f|_\rho) = O(k)$.*

Proof. Recall $f = \mathsf{OR}_k \circ g$, where $g\colon \{0,1\}^k \to \{0,1\}$ evaluates to 1 iff the input to g contains $\sqrt{k}$ consecutive ones and the rest of the bits are zeroes. Let $x = (x_{11}, \ldots, x_{1k}, \ldots, x_{k1}, \ldots, x_{kk})$ be an input to f. For $1 \le i \le k$, let $g_i = g(x_{i1}, \ldots, x_{ik})$ be the i-th copy of g.

Since $\mathsf{C}_1(f) = k$, it suffices to show $\mathsf{C}_0(f|_\rho) = O(k)$ for every $\rho\colon [k^2] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(k^{1.5})$.

A 0-certificate for $f|_\rho$ comprises of at most k 0-certificates, one for each copy of $g|_\rho$ that remains non-zero. That is, $\mathsf{C}_0(f|_\rho) \le \sum_{i=1}^{k} \mathsf{C}_0(g_i|_\rho)$. We therefore now bound $\mathsf{C}_0(g_i|_\rho)$ when $g_i|_\rho$ is not equal to the constant function 0. We assume, wlog, that $g_1|_\rho$ is non-constant. Based on the restriction ρ, we have two cases to consider.

Case 1: ρ sets some variable of g_1 to 1. That is, there exists $1 \le j \le k$, such that $\rho(x_{1j}) = 1$. In this case, $\mathsf{C}_0(g_1|_\rho) \le 3$.

This is essentially because we only need to certify that the sequence of 1s in a 0-input to $g_1|_\rho$ does not satisfy the required property. In particular, any 0-input of $g_1|_\rho$ is of type (1), (2), or (3).

Case 2: ρ sets no variable of g_1 to 1. That is, ρ may set some variables in $\{x_{11}, \ldots, x_{1k}\}$ to 0 and leaves the rest unset. Let η_1 be the number of variables of $g_1|_\rho$. That is, $|\rho^{-1}(*) \cap \{x_{11}, \ldots, x_{1k}\}| = \eta_1$.

In such a case, observe that the all-zero input to $g_1|_\rho$ remains the hardest to certify. Thus, $\mathsf{C}_0(g_1|_\rho) = \mathsf{C}(g_1|_\rho, 0^{\eta_1}) \leq \eta_1/\sqrt{k}$. The inequality follows because there is a certificate that reveals a 0-bit for every consecutive $\sqrt{k}$ positions and, hence its size is at most $\eta_1/\sqrt{k}$.

Let $\eta_i = |\rho^{-1}(*) \cap \{x_{i1}, \ldots, x_{ik}\}|$ be the number of variables of $g_i|_\rho$ for $1 \leq i \leq k$. We are now ready to bound $\mathsf{C}_0(f|_\rho)$. Let t be the number of g_i's such that $g_i|_\rho$ falls in **Case 2**. Further assume, wlog, that these are $\{g_1, g_2, \ldots, g_t\}$. Then, we have

$$\begin{aligned}\mathsf{C}_0(f|_\rho) &\leq \sum_{i=1}^{k} \mathsf{C}_0(g_i|_\rho) \leq 3(k-t) + \sum_{i=1}^{t} \mathsf{C}_0(g_i|_\rho) \leq 3(k-t) + \sum_{i=1}^{t} \left(\eta_i/\sqrt{k}\right) \\ &\leq 3(k-t) + |\rho^{-1}(*)|/\sqrt{k} = O(k).\end{aligned}$$

◀

▶ Remark. We note that the modified Rubinstein function, even with different parameters, cannot give a better counter-example for lossless condensation than Theorem 25. For a proof, please check the full version.

4 Negative Results II: Incondensability of Decision Tree Complexity

In this section, we prove Theorem 2, i.e., query complexity and its variants do not condense losslessly. In particular, we show that D_0-query and $\mathsf{D}_\wedge$-query complexities do not condense losslessly and also present an improved example for the fact that decision tree query complexity does not condense losslessly. In fact, we will use the same function to establish all three claims.

▶ **Theorem 2.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_0(f) \leq \mathsf{D}_\wedge(f) \leq \mathsf{D}(f) = O(n^{3/5})$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/5})$ we have $\mathsf{D}_0(f|_\rho) \leq \mathsf{D}_\wedge(f|_\rho) \leq \mathsf{D}(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_0(f)^{2/3})$.*

It was shown in [16] that there exists a function h such that for every restriction ρ that fixes all but at most $O(\mathsf{D}(h))$ variables, the query complexity $\mathsf{D}(h|_\rho)$ of the restricted function $h|_\rho$ is $\widetilde{O}(\mathsf{D}(h)^{3/4})$. We give an improved example by exhibiting a function h' such that $\mathsf{D}(h'|_\rho) = \widetilde{O}(\mathsf{D}(h')^{2/3})$. Our improved example is a variant of the function considered in [16]. In particular, our function is also a cheat-sheet version of Tribes (see Definition 24 for cheatsheet functions), but we use different arities for the basic Tribes function as well as keep the number of cheat-sheet cells much lower. Furthermore, we will use the same function to show that D_0 and $\mathsf{D}_\wedge$ also do not condense losslessly.

We consider the following Tribes function as the base function.

▶ **Definition 28** (Tribes function)**.** *Define $f\colon \{0,1\}^{k^{1.5}} \to \{0,1\}$ to be the function $(\mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}})(x)$ where $x \in \{0,1\}^{k^{1.5}}$.*

We note a few well-known properties of the Tribes function.

▶ **Proposition 29.** *For the Tribes function f as defined in Definition 28, it is known that $\mathsf{C}_0(f) = \sqrt{k}$, $\mathsf{C}_1(f) = k$, $\mathsf{D}(f) = k\sqrt{k}$ and $\mathsf{D}_0(f) \geq k\sqrt{k} - k + 1$.*

Proof. Though the bounds on C_0, C_1, and D are well-known, the bound on D_0 is not often encountered. We sketch a proof here for completeness.

Consider the following (usual) adversary algorithm for the Tribes: On a query by a decision tree algorithm, the adversary responds with 0 if the OR containing the queried variable has variables that have not been queried yet. Otherwise it's the last variable to be queried in that OR, in such a case the adversary responds with 1 if there are other ORs whose output is not yet fixed. Finally, the last variable in the last OR is queried, which implies the lower bound. ◀

We consider the *cheat-sheet* version f_{CS} of the Tribes function f (Definition 28) where certificates in the cheat-sheet cells are described as follows. A cell Y_ℓ, where ℓ in binary is given by $\ell_1\ell_2\cdots\ell_{\log t} \in \{0,1\}^{\log t}$, describes a ℓ_i-certificate for x_i with respect to f for $i \in [\log t]$.

Describing 1-certificates: A 1-certificate for $f = \mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}}$ contains a 1-certificate for each copy of OR. For each 1-certificate, the description is given by the label of a variable which is set to 1. This requires $\log k^{1.5}$ bits per OR. Therefore, the full description of a 1-certificate for f requires $k(\log k^{1.5})$ bits.

Describing 0-certificates: A 0-certificate for $f = \mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}}$ contains a 0-certificate for one of the copies of OR. Hence, we describe a 0-certificate for f by just writing the label of a copy of OR. Thus, the full description of a 0-certificate for f requires $\log k$ bits.

Therefore, the cheat-sheet version f_{CS} of the Tribes function is

$$f_{\mathrm{CS}} \colon \{0,1\}^{k^{1.5}(\log t)+(t\log t)k(\log k^{1.5})} \to \{0,1\}, \tag{1}$$

where its input is $X = (x_1,\ldots,x_{\log t}, Y_0,\ldots,Y_{t-1})$, and $f_{\mathrm{CS}}(X) = 1$ iff the cheat-sheet cell $Y_\ell \in \{0,1\}^{(\log t)k(\log k^{1.5})}$, given by the positive integer ℓ corresponding to the binary string $f(x_1)f(x_2)\cdots f(x_{\log t})$, describes $f(x_i)$-certificate for each x_i as given by the description above. Note that 0-certificates are much smaller than 1-certificates, but we will use the same amount of space, $k(\log k^{1.5})$, for each, with the notation that in the case of 0-certificates first $\log k$ bits describe them.

We now show that for the choice of $t = k^{1.5}$, both $\mathsf{D}(f_{\mathrm{CS}})$ and $\mathsf{D}_0(f_{\mathrm{CS}})$ equals $\widetilde{\Theta}(k^{1.5})$, but for every restriction ρ that fixes all but at most $O(\mathsf{D}(f_{\mathrm{CS}}))$ variables, the query complexity $\mathsf{D}(f_{\mathrm{CS}}|_\rho)$ (and hence $\mathsf{D}_0(f_{\mathrm{CS}}|_\rho)$) of the restricted function $f_{\mathrm{CS}}|_\rho$ is $\widetilde{O}(k)$. We now prove each of these assertions one by one. In the following fix $t = k^{1.5}$. Then the number of variables for f_{CS} is $\Theta(k^{2.5}\log^2 k)$.

▶ **Lemma 30.** $\mathsf{D}(f_{\mathrm{CS}}) \geq t = k^{1.5}$ *and* $\mathsf{D}_0(f_{\mathrm{CS}}) \geq t - t^{2/3}$.

Proof. We will give an adversary argument for f_{CS} based on the adversary for f mentioned in Proposition 29.

On a query, the adversary for f_{CS} will respond as follows: If the queried variable is from the address part, i.e., $x_1,\ldots,x_{\log t}$, then the response is in accordance with the adversary of f given in Proposition 29. Otherwise, the queried variable is from the cheat-sheet cells, i.e., $Y_0,\ldots,Y_{t-1}$, and then the adversary responds with 0.

Now if an algorithm makes only $t-1$ queries and they are answered according to the adversary defined above, then the following observations hold:

- No $f(x_i)$ has been fixed yet,
- there is at least one cheat-sheet cell, say Y_ℓ, which has not been queried at all, and
- the number of 0-queries made until now is at least $t-1-k$.

We can therefore extend x_i's in a way that the address points to the cell Y_ℓ and, furthermore, the cell Y_ℓ can be set as we wish to give values 0 or 1 to contradict the algorithm's answer; thus finishing the adversary argument. ◀

We now present a claim that will be used to construct a query algorithm for f_{CS} as well as its restricted version.

▷ Claim 31. Given two distinct indices $\ell, p \in \{0,\ldots,t-1\}$, at least one of the cells Y_ℓ and Y_p can be discarded with $O(\log k)$ queries.

Proof. Since $\ell \neq p$, there exists $i \in [\log t]$ such that $\ell_i \neq p_i$. Assume, wlog, $\ell_i = 0$ and $p_i = 1$.

The algorithm queries Y_ℓ to know the label of OR, which is a supposed 0-certificate for $f(x_i)$. Let OR_j, $j \in [k]$, be the copy of OR within the i-th copy of f that Y_ℓ has listed as 0-certificate for x_i. Observe that Y_p is supposed to have a 1-certificate for OR_j. Recall a 1-certificate for OR_j is the label of a variable of OR_j. Query the variables in Y_p and the input x_i to verify the 1-certificate for OR_j. Now, if the claimed 1-certificate for OR_j is valid, then we can remove Y_ℓ; else, Y_p can be removed.

The total number of queries made is at most $O(\log k)$. ◁

We are now ready to show a nearly tight upper bound on $\mathsf{D}(f_{\mathrm{CS}})$. Using Claim 31, we prove that the bound of Lemma 30 is tight up to logarithmic factors. By a straightforward algorithm that solves each copy of f and then verifies the contents of the cheat-sheet cell pointed to by the evaluations of f, we get the following. In the full version of the paper, we give another algorithm for f_{CS} that is instructive when designing an algorithm for the restricted version.

▶ **Lemma 32.** $\mathsf{D}(f_{\mathrm{CS}}) = O(t \log k \ + \ k \log k \log t)$.

We now finish the last, but the most important, task of showing that indeed the decision tree complexity of the restricted function reduces; thereby finishing the counterexample for the hardness condensation of decision tree complexity and its variants.

▶ **Lemma 33.** *For every partial assignment ρ to the variables of f_{CS} with $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}})) = O(k^{1.5} \log k)$, we have $\mathsf{D}(f_{\mathrm{CS}}|_\rho) = O(k \log^2 k)$.*

Proof. Fix a partial assignment ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}})) = O(k^{1.5} \log k)$. Our query algorithm for $f_{\mathrm{CS}}|_\rho$ on an unknown input $X|_\rho = (x_1|_\rho, \ldots, x_{\log t}|_\rho, Y_0|_\rho, \ldots, Y_{t-1}|_\rho)$ has two parts like the unrestricted case.

In the first part we identify a cheat-sheet cell (to be verified later) by zeroing in on the string $z \in \{0,1\}^{\log t}$ that the address part $f(x_1|_\rho)f(x_2|_\rho)\cdots f(x_{\log t}|_\rho)$ can take if the restricted function $f_{\mathrm{CS}}|_\rho$ were to evaluate to 1.

In the second part, like the unrestricted case, we verify the certificates described in Y_z by querying the variables in $Y_z|_\rho$ and the corresponding certificate variables in $(x_1|_\rho, \ldots, x_{\log t}|_\rho)$. If the verification passes, the algorithm outputs 1, else it outputs 0. Since the algorithm needs to verify the contents of one cell, the total number of queries made in the second part is $O(k \log k \log t + \mathsf{C}(f) \log t) = O(k \log^2 k)$.

Algorithm for the first part. We now give the details of our algorithm for the first part. Recall that the address part of the input contains $\log t$ copies of the base function f. We will run (at most) $\log t$ many iterations of the algorithm. At the end of each iteration i, we will have a candidate value for $f(x_i|_\rho)$, i.e., either we know the value of $f(x_i|_\rho)$ or there are no valid cheat-sheet cells for $f(x_i|_\rho) = 1$. At the end of the $\log t$ iterations, there will only be one remaining (possibly valid) cheat-sheet cell to be verified. We now give details of an iteration of the algorithm.

Let c be a constant such that $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}})) \leq ck^{1.5} \log k$.

Details of a particular iteration. Fix an $i \in [\log t]$. At any given point of the iteration let $\mathcal{V}$ be the set of "alive" cheat-sheets (cheat-sheets which have not been found inconsistent) and $\mathcal{V}_1$ be the cheat-sheet cells in $\mathcal{V}$ where, supposedly, $f(x_i|_\rho) = 1$, (i.e., the i-th bit of the index of the cell is 1). On the other hand, let $\mathcal{O}_0$ denote the set of OR_j's ($j \in [k]$) in the i-th copy of f which could still possibly be 0. Initially, $|\mathcal{O}_0| = k$ and, in the first iteration, $|\mathcal{V}| = t$ and $|\mathcal{V}_1| = t/2$. Also, note that each cell in $\mathcal{V}_1$ is supposed to contain a 1-certificate for k many ORs.

In each iteration, the algorithm will have two stages, where the first stage (and arguably the more involved one) will get us to a situation that can be handled by previous techniques. More formally, after the first stage, either $|\mathcal{O}_0| \leq c\sqrt{k}\log k$ or $|\mathcal{V}_1| \leq 2k$.

The second stage addresses these two cases. In the first case, we can check all the alive OR's exhaustively. For the second case, using Claim 31, we will get a single cheat-sheet cell in $|\mathcal{V}_1|$ which is then verified to ascertain whether indeed it contains a 1-certificate for $f(x_i|_\rho)$.

We now describe the two stages in detail.

Stage 1: We can assume that $|\mathcal{O}_0| > c\sqrt{k}\log k$ and $|\mathcal{V}_1| > 2k$, otherwise we move to the second stage.

Looking at $\mathcal{V}_1$, for each $\mathsf{OR}_j \in \mathcal{O}_0$, we call an OR_j to be *revealed* if (supposed) 1-certificates for that particular OR_j is completely fixed by ρ (i.e., not a single $*$) in at least $|\mathcal{V}_1|/2$ many cells in $\mathcal{V}_1$. Notice that if there are no revealed OR's, then for more than $c\sqrt{k}\log k$ many OR's there are at least $|\mathcal{V}_1|/2 > k$ many $*$'s. This is a contradiction because the number of $*$'s, by our assumption, is at most $ck^{1.5}\log k$.

For a revealed j, there exists a *popular* variable, say $l \in [\sqrt{k}]$, which appears in at least $|\mathcal{V}_1|/2\sqrt{k}$ many cells in $\mathcal{V}_1$. We query $x_{i,j,l}$ from the input part x_i. If it is 1 then OR_j can be discarded from $\mathcal{O}_0$, otherwise it is 0 and at least $|\mathcal{V}_1|/2\sqrt{k}$ many cells in $\mathcal{V}_1$ can be thrown out.

Essentially, the first stage of the iteration can be described as: find a revealed OR and query the popular variable till $|\mathcal{O}_0| \leq c\sqrt{k}\log k$ or $|\mathcal{V}_1| \leq 2k$. The existence of such a revealed OR follows from the argument described above. We now bound the query complexity in this stage.

Notice that after every query either $|\mathcal{O}_0|$ reduces by 1 (popular variable being 1) or $|\mathcal{V}_1|$ reduces by a factor of $(1 - \frac{1}{2\sqrt{k}})$ (popular variable being 0). So the query complexity of the first stage is bounded by $O(k + \sqrt{k}\log t) = O(k)$.

Stage 2: We know that either $|\mathcal{O}_0| \leq c\sqrt{k}\log k$ or $|\mathcal{V}_1| \leq 2k$. As alluded to before, for the first case, we can query all the OR's in $\mathcal{O}_0$ and find the value of $f(x_i|_\rho)$. Since the arity of each OR is $\sqrt{k}$, this requires $O(k\log k)$ queries.

For the second case, use Claim 31 to reduce the size of $\mathcal{V}_1$ to one using $O(k\log k)$ many queries. Then, in the remaining cell, the claimed 1-certificate for $f(x_i|_\rho)$ can be verified by reading the entire certificate in $O(k\log k)$ queries. In other words, the second stage makes $O(k\log k)$ many queries.

Hence, the total number of queries made in both stages of the iteration is $O(k\log k)$. This finishes one iteration $i \in [\log t]$ of the algorithm.

Query complexity of the entire algorithm: Since there are $\log t$ many iterations, the total queries made in the first part of the algorithm are $O(k\log^2 k)$. After these $\log t$ iterations, in the second part, there is at most one possible valid cheat-sheet, which can be checked in $O(k\log^2 k)$ many queries (as described in the beginning). Therefore, the entire algorithm makes at most $O(k\log^2 k)$ queries.

Correctness: The correctness of the algorithm follows from the fact that a cheat-sheet cell is removed from the set of valid cells if and only if the cell is found to be inconsistent with respect to input $(x_1|_\rho, \ldots, x_{\log t}|_\rho)$ and f, or when we know the value of $f(x_i)$. Further, the only remaining valid cell is verified in the second part. ◀

As a corollary to Lemma 30, Lemma 32, Lemma 33 and Fact 16, we obtain an improved example for the incondensability of query complexity and also show incondensability of D_0 and $\mathsf{D}_\wedge$.

▶ **Theorem 34.** *Let f_{CS} be the Boolean function on $\Theta(k^{2.5}\log^2 k)$ variables as defined in Equation (1). Then, the following holds*

- $\Omega(k^{1.5}) = \mathsf{D}_0(f_{\mathrm{CS}}) \le \mathsf{D}_\wedge(f_{\mathrm{CS}}) \le \mathsf{D}(f_{\mathrm{CS}}) = O(k^{1.5}\log k)$, *and*
- *for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}}))$, we have $\mathsf{D}_0(f_{\mathrm{CS}}|_\rho) \le \mathsf{D}_\wedge(f_{\mathrm{CS}}|_\rho) \le \mathsf{D}(f_{\mathrm{CS}}|_\rho) = O(k\log^2 k) = \widetilde{O}(\mathsf{D}_0(f_{\mathrm{CS}})^{2/3})$.*

This completes the proof of Theorem 2. We take a moment to note that if we consider the function $h = \mathsf{OR}_k \circ \mathsf{AND}_{\sqrt{k}}$ and the cheat-sheet version h_{CS} of h, a dually similar lower bound and algorithm for h_{CS} and its restricted version exists, which implies the following theorem on the incondensability of OR-decision tree complexity.

▶ **Theorem 3.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_1(f) \le \mathsf{D}_\vee(f) = O(n^{3/5})$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/5})$ we have $\mathsf{D}_1(f|_\rho) \le \mathsf{D}_\vee(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_1(f)^{2/3})$.*

We further observe that the zero-error randomized query complexity $\mathsf{R}_0(f_{\mathrm{CS}})$ of f_{CS} is also high, i.e., $\Omega(k^{1.5})$, thus obtaining an improved example for the incondensability of R_0 (Corollary 4).

▶ **Theorem 35.** *Let f_{CS} be the Boolean function on $\Theta(k^{2.5}\log^2 k)$ variables as defined in Equation (1). Then, the following holds*

- $\mathsf{R}_0(f_{\mathrm{CS}}) = \tilde{\Theta}(k^{1.5})$, *and*
- *for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{R}_0(f_{\mathrm{CS}}))$, we have $\mathsf{R}_0(f_{\mathrm{CS}}|_\rho) = O(k\log^2 k) = \widetilde{O}(\mathsf{R}_0(f_{\mathrm{CS}})^{2/3})$.*

Proof. Since $\mathsf{R}_0 \le \mathsf{D}$, it suffices to show that $\mathsf{R}_0(f_{\mathrm{CS}}) = \Omega(k^{1.5})$ (Theorem 34 will then complete the proof). To prove the lower bound on $\mathsf{R}_0(f_{\mathrm{CS}})$, note that $\mathsf{R}_0(f) = \Omega(\mathsf{R}(f))$ and $\mathsf{R}(f) = \Theta(k^{1.5})$ [21]. See also [15, 4, 11]. We will show a reduction from f_{CS} to f.

Given an input x for f, the input of f_{CS} (say y) will consist of $\log t$ copies of x in the address part and all 0's in the cheat-sheet part. Whenever the R_0 algorithm for f_{CS} answers correctly (i.e., does not say "don't know"), it has queried a certificate for y (say C_y). Note that the number of cheat-sheets is more than the allowed number of queries to the R_0 algorithm. This implies that the certificate C_y hasn't queried at least one cheat-sheet cell at all.

We claim that this certificate C_y (that hasn't queried any element of a cheat-sheet cell) has a certificate for at least one address bit. To prove the claim by contradiction, if C_y does not have a certificate for x (i.e., does not certify any of the address bits), the address bits of y could be flipped (keeping it consistent with C_y) to point to the cheat-sheet cell which has not been queried. Since C_y does not contain any element of this cheatsheet cell, but an input consistent with C_y could point to this cheatsheet cell, we have a contradiction.

To summarize, the R_0 algorithm for f_{CS} gives a certificate for y which contains a certificate for at least one address bit. Since the certificate for any address bit is a certificate for x, this will finish the R_0 algorithm for f. ◀

5 Positive Results: Lossy Condensation

It is a natural question to ask if the gaps in Theorem 1 and Theorem 2 are optimal. In other words, does there exist a function for which hardness is even less condensable than shown in these theorems?

In particular, for deterministic query complexity, it asks if there exists a function f such that for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f))$, we have $\mathsf{D}(f|_\rho) = O(\mathsf{D}(f)^\alpha)$ where $\alpha < 2/3$? Observe that $\alpha \geq 1/3$, since $\mathsf{D}(f) \leq \deg(f)^3$ [25] and $\deg(f)$ condenses losslessly. Note that if the improved bound $\mathsf{D}(f) \leq \deg(f)^2$ were known to hold, then we would have a better lower bound for the exponent.

Similarly, for block sensitivity, the exponent $\alpha \geq 1/4$, since $\mathsf{bs}(f) \leq \mathsf{s}(f)^4$ [20] and $\mathsf{s}(f)$ condenses losslessly. Note again that if the improved bound $\mathsf{bs}(f) \leq \mathsf{s}(f)^2$ were known to hold, then we would have a better lower bound for the exponent.

In this section, we will establish an improved lower bound for the exponent for almost all decision tree-based complexity measures.

▶ **Theorem 5.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function and $\mathcal{M} \in \{\mathsf{bs}, \mathsf{fbs}, \mathsf{C}, \mathsf{UC}_{\min}, \mathsf{UC}_1, \mathsf{UC}, \mathsf{D}, \widetilde{\deg}, \lambda, \mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$ be a complexity measure. Then, the following holds:*

(a) *For $\mathcal{M} \notin \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ such that $\mathcal{M}(f|_\rho) = \Omega(\sqrt{\mathcal{M}(f)})$.*

(b) *For $\mathcal{M} \in \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}_E\}$, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ such that $\mathcal{M}(f|_\rho) = \Omega(\mathcal{M}(f)^{1/3})$.*

(c) *For $\mathcal{M} = \mathsf{Q}$, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathsf{Q}(f))$ such that $\mathsf{Q}(f|_\rho) = \Omega(\mathsf{Q}(f)^{1/4})$.*

The proof for Theorem 5 is obtained by noting that the two measures, sensitivity and degree, condense losslessly; and then exploiting known relationships between different complexity measures. The technical details (and complete proofs) are given in the full version.

The proof strategy outlined above works for almost all measures except Q, Q_E, $\widetilde{\deg}$ and λ. The best bounds for these measures are obtained by noting that the two measures, sensitivity and degree, indeed follow a slightly stricter form of hardness condensation. We present this stronger hardness condensation here (It shows that sensitivity and degree can be condensed to any parameter below itself). Again, the application of this stricter form (hardness condensation for Q, Q_E, $\widetilde{\deg}$ and λ) can be seen in the full version.

▶ **Lemma 36** (Stronger hardness condensation)**.** *Let $\mathcal{M} \in \{\mathsf{s}, \deg\}$ and $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let $k \leq \mathcal{M}(f)$. Then, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = k$ such that $\mathcal{M}(f|_\rho) = k$.*

Proof. We argue each case separately.

$\mathcal{M} = \mathsf{s}$. Let $a \in \{0,1\}^n$ be an input to f such that $\mathsf{s}(f,a) = \mathsf{s}(f)$. Given any integer $k \leq \mathsf{s}(f)$, consider a restriction ρ that leaves any k sensitive variables of a unset and sets the rest of the variables according to a. Clearly, $|\rho^{-1}(*)| = k$ and $\mathsf{s}(f|_\rho) = k$ as well.

$\mathcal{M} = \deg$. Let $k \leq \deg(f)$. Let p be the unique multilinear polynomial that represents f. Suppose p contains a monomial $\mathfrak{m}$ of degree k. Consider a restriction ρ that leaves variables in $\mathfrak{m}$ unset and sets the rest of the variables to 0. Clearly, $p|_\rho$ is the unique polynomial of degree k that represents $f|_\rho$ over k variables.

Now suppose p does not contain any monomial of degree k. Let $d > k$ be the smallest positive integer such that there exists a monomial $\mathfrak{m}$ of degree d in p, and there does not exist a monomial of degree $i \in [k, d-1]$ in p. Consider a restriction ρ that leaves any k variables in $\mathfrak{m}$ unset, sets the rest of $d-k$ variables in $\mathfrak{m}$ to 1, and sets the rest of the variables not in $\mathfrak{m}$ to 0. The resulting monomial in the restriction (from $\mathfrak{m}$) will have a non-zero coefficient and cannot be cancelled by any other restricted monomial from the polynomial p. So, such a restriction ρ will give us a polynomial of degree k that represents $f|_\rho$ over k variables. ◀

We show the following additional result on Fourier sparsity (see full version for details).

▶ **Theorem 37.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function with Fourier sparsity k. Then there exists a partial assignment ρ on $O(k^2)$ variables such that the Fourier sparsity of the restricted function f_ρ is k.*

Although the Fourier sparsity of an n-bit Boolean function can be as large as 2^n, Theorem 37 is non-trivial in regimes where sparsity is $O(\sqrt{n})$. In particular, many natural Boolean functions (e.g., parity and juntas) have Fourier sparsity $k \leq \sqrt{n}$, and hence admit condensation to $O(k^2)$ variables without loss of sparsity. An interesting open question suggested by our result is whether *logarithmic Fourier sparsity* admits condensation.

6 Conclusion

Given that hardness condensation has been pretty useful in proving sharper lower bounds in circuit complexity theory and proof complexity, a natural question to ask is: "how much" of hardness condensation is possible for the complexity measures related to decision tree complexity? Taking cue from Göös, Newman, Riazanov and Sokolov [16]; we initiate the study for almost all established complexity measures and prove both negative and positive results about condensation (see Table 1).

In terms of negative results, we show that lossless condensation is not possible for block sensitivity, certificate complexity, AND-decision tree query complexity, and several other complexity measures. We also prove that there exist variable restrictions that yield polynomial condensation gaps for different complexity measures. However, it remains open whether there exist functions that exhibit even stronger condensation barriers than what is established in this work (see Table 1). We list several other open questions for future work.

(1) An important unresolved question is whether deterministic communication complexity condenses losslessly. Since the AND-decision tree complexity does not condense (Theorem 2), is it possible to show that the answer is negative (as conjectured by [16]) by lifting with 2-bit AND gadget?

(2) Closing several gaps between positive and negative condensation for different measures listed in Table 1 remains an interesting question to explore. An intriguing one is whether unambiguous certificate complexity condenses losslessly (see also [16, 19]).

(3) Another avenue to explore is condensation with respect to projections or other similar restricted reductions. Typically, such modifications are used in applications in proof complexity (see, e.g., [28, 29, 6, 14, 10, 12, 13]).

(4) As mentioned in the introduction (Section 1, see also [18]), "cross condensation" between Boolean function parameters is another interesting variant of hardness condensation to explore. For example, if there is a Boolean function f such that for any restriction ρ, on $O(s(f))$ many variables $\mathsf{bs}(f_\rho) = \omega(\sqrt{\mathsf{bs}(f)})$, then f must have a super-quadratic gap between sensitivity and block sensitivity.

References

1 S. Aaronson, S. Ben-David, and R. Kothari. Separations in query complexity using cheat sheets. In *STOC 2016*, pages 863–876, 2016. doi:10.1145/2897518.2897644.

2 S. Aaronson, S. Ben-David, R. Kothari, S. Rao, and A. Tal. Degree vs. approximate degree and quantum implications of Huang's sensitivity theorem. In *STOC 2021*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.

3 A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, and J. Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM*, 64(5):32:1–32:24, 2017. doi:10.1145/3106234.

4 A. Ambainis, M. Kokainis, and R. Kothari. Nearly optimal separations between communication (or query) complexity and partitions. In *CCC 2016*, volume 50, pages 4:1–4:14, 2016. doi:10.4230/LIPIcs.CCC.2016.4.

5 S. Ben-David, P. Hatami, and A. Tal. Low-sensitivity functions from unambiguous certificates. In *ITCS*, volume 67, pages 28:1–28:23, 2017. doi:10.4230/LIPIcs.ITCS.2017.28.

6 C. Berkholz and J. Nordström. Supercritical space-width trade-offs for resolution. *SIAM J. Comput.*, 49(1):98–118, 2020. doi:10.1137/16M1109072.

7 H. Buhrman and R. de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity 2001*, pages 120–130, 2001. doi:10.1109/CCC.2001.933879.

8 H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

9 J. Buresh-Oppenheim and R. Santhanam. Making hard problems harder. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006)*, pages 73–87, 2006. doi:10.1109/CCC.2006.26.

10 S. Buss and N. Thapen. A simple supercritical tradeoff between size and height in resolution. *Electron. Colloquium Comput. Complex.*, TR24-001, 2024. URL: https://eccc.weizmann.ac.il/report/2024/001.

11 S. Chakraborty, C. Kayal, R. Mittal, M. Paraashar, S. Sanyal, and N. Saurabh. On the Composition of Randomized Query Complexity and Approximate Degree. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275, pages 63:1–63:23, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.63.

12 A. Chattopadhyay and P. Dvorak. Super-critical trade-offs in resolution over parities via lifting. *Electron. Colloquium Comput. Complex.*, TR24-132, 2024. URL: https://eccc.weizmann.ac.il/report/2024/132.

13 S. F. de Rezende, N. Fleming, D. A. Janett, J. Nordström, and S. Pang. Truly supercritical trade-offs for resolution, cutting planes, monotone circuits, and weisfeiler-leman. *CoRR*, abs/2411.14267, 2024. doi:10.48550/arXiv.2411.14267.

14 N. Fleming, T. Pitassi, and R. Robere. Extremely deep proofs. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215, pages 70:1–70:23, 2022. doi:10.4230/LIPIcs.ITCS.2022.70.

15 M. Göös, T. S. Jayram, T. Pitassi, and T. Watson. Randomized communication versus partition number. *ACM Trans. Comput. Theory*, 10(1), 2018. doi:10.1145/3170711.

16 M. Göös, I. Newman, A. Riazanov, and D. Sokolov. Hardness condensation by restriction. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 2016–2027, 2024. doi:10.1145/3618260.3649711.

17 L. Hambardzumyan, H. Hatami, and P. Hatami. A counter-example to the probabilistic universal graph conjecture via randomized communication complexity. *Discret. Appl. Math.*, 322:117–122, 2022. doi:10.1016/J.DAM.2022.07.023.

18 G. Hart. *Condensing Hardness in Boolean Functions*, 2025. Undergraduate Research Report, University of Rochester.

19 P. Hrubes. Hard submatrices for non-negative rank and communication complexity. In *39th Computational Complexity Conference, CCC 2024*, volume 300, pages 13:1–13:12, 2024. doi:10.4230/LIPIcs.CCC.2024.13.

20 H. Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190(3):949–955, 2019. doi:10.4007/annals.2019.190.3.6.

21 R. Jain and H. Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010*, pages 247–258, 2010. doi:10.1109/CCC.2010.31.

22 A. Knop, S. Lovett, S. McGuire, and W. Yuan. Log-rank and lifting for and-functions. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 197–208, 2021. doi:10.1145/3406325.3450999.

23 B. Loff and S. Mukhopadhyay. Lifting theorems for equality. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126, pages 50:1–50:19, 2019. doi:10.4230/LIPIcs.STACS.2019.50.

24 L. Lovász and M. Saks. Lattices, möbius functions and communication complexity. In *29th Annual Symposium on Foundations of Computer Science*, pages 81–90, 1988. doi:10.1109/SFCS.1988.21924.

25 G. Midrijanis. Exact quantum query complexity for total Boolean functions. *arXiv:quant-ph/0403168v2*, 2004.

26 N. Nisan. Crew prams and decision trees. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 327–335, 1989.

27 A. A. Razborov. A new kind of tradeoffs in propositional proof complexity. *J. ACM*, 63(2):16:1–16:14, 2016. doi:10.1145/2858790.

28 A. A. Razborov. On the width of semialgebraic proofs and algorithms. *Math. Oper. Res.*, 42(4):1106–1134, 2017. doi:10.1287/MOOR.2016.0840.

29 A. A. Razborov. On space and depth in resolution. *Comput. Complex.*, 27(3):511–559, 2018. doi:10.1007/S00037-017-0163-1.

30 D. Rubinstein. Sensitivity vs. block sensitivity of Boolean functions. *Combinatorica*, 15(2):297–299, 1995. doi:10.1007/BF01200762.

31 A. Tal. Properties and applications of boolean function composition. In *Innovations in Theoretical Computer Science, ITCS '13*, pages 441–454. ACM, 2013. doi:10.1145/2422436.2422485.