Electronic Colloquium on Computational Complexity, Report No. 195 (2026)

Sumset Structure in Local Computation

Alexander Golovnev* Mohit Gurumukhani†

Abstract

We introduce and study a new model of decision trees that lies at the frontier of provable circuit lower bounds.

We study *ℓ-local decision trees*, in which each internal node queries an ℓ-local function of the input. We show that sufficiently strong lower bounds for ℓ-local decision trees would imply several breakthrough circuit lower bounds, including super-linear size lower bounds for log-depth circuits and improved bounds for unrestricted-depth circuits. Previously, such consequences were known to follow from strong lower bounds for depth-3 circuits, which has been the main prior approach in attempting to prove these results. Since local decision trees are strictly weaker than depth-3 circuits, this provides a formally easier route to the same circuit lower-bound consequences.

We prove essentially optimal lower bounds for a weaker variant that we call *oblivious ℓ-local decision trees*, where all nodes at the same depth query the same function. Our lower bounds follow from a new technique that uncovers sumset structure in local maps, and our hard functions are sumset dispersers, sumset condensers, and directional affine dispersers. Along the way, we give an explicit construction of a sumset condenser with small entropy loss.

As an additional contribution, we initiate the study of *degree-2 decision trees*, in which each query is a quadratic polynomial of the input. We show that proving lower bounds in this model is a natural stepping stone toward constructing dispersers for degree-2 variety sources, which imply improved circuit lower bounds for unrestricted depth circuits. We prove nearly maximal lower bounds for the oblivious variant of the model.

*Cornell University. This research is supported by the National Science Foundation CAREER award (grant CCF-2338730). Email: alexgolovnev@gmail.com.

†Cornell University. This research is supported by a Sloan Research Fellowship, the National Science Foundation CAREER Award (grant CCF-2045576), and the National Science Foundation Award CCF-2514586. Email: mgurumuk@cs.cornell.edu.

ISSN 1433-8092

Contents

1 Introduction

The central goal of circuit complexity is to prove lower bounds on the complexity of an explicit function in various models of computation. A counting argument shows that almost all functions require very large complexity in most computational models. Yet it is notoriously hard to find a polynomial-time computable function which exhibits a non-trivially large complexity. The lack of non-trivial lower bounds is used as an explanation for why progress has been stalled in other areas of complexity, such as proving $\mathsf{BPP} = \mathsf{P}$ or proving $\mathsf{P} \neq \mathsf{NP}$.

Directly working with various circuit models in the hope of proving lower bounds is often quite difficult since it is hard to analyze them. Therefore, one often reduces the task of proving lower bounds against various circuit models to another task such as: 1) proving lower bounds for other computational models which are seemingly easier to analyze such as depth-3 circuits or the number-on-forehead model; 2) constructing explicit pseudorandom objects such as rigid matrices or dispersers for variety sources. Despite intense efforts to: improve these reductions, prove better lower bounds for seemingly simpler models, and to construct pseudorandom objects, there has been very little progress in obtaining these lower bounds.

Motivated by this, we initiate the systematic study of *local decision trees*, a natural model of computation where proving strong lower bounds yields many strong circuit lower bounds.[1] Our model has rich connections to many other models of computation and we will show that proving lower bounds against local decision trees is an easier task: that lower bounds against simple circuit models as well as explicit constructions of pseudorandom objects will yield lower bounds against these trees. We will then prove nearly maximal lower bounds against a slightly weakened version of local decision trees by finding additive structure in local computation. The hard functions for which our lower bound will hold will be sumset dispersers and condensers, which either we already know explicitly how to construct or we construct in this work for our task.

Let us first define these trees. An *ℓ-local decision tree* over n variables is a binary tree where each internal node queries an ℓ-local function of the input and based on that branches left or right, until it reaches a leaf which is labeled with the outcome 0 or 1. We will prove strong lower bounds against "oblivious" versions of these decision trees where all nodes at the same level compute the exact same function. An *oblivious ℓ-local decision tree* T of depth d can also be characterized by an ℓ-local map $L : \{0,1\}^n \to \{0,1\}^d$ and a function $h : \{0,1\}^d \to \{0,1\}$ of arbitrary complexity such that $T(x) = h(L(x))$. We also naturally extend these definitions to functions that output m bits by letting each leaf be labeled by an outcome from $\{0,1\}^m$.

We additionally study degree-d decision trees—decision trees where at each node, the function queried is a degree-d multilinear polynomial (over $\mathbb{F}_2$) of the input bits. We prove maximal lower bounds against oblivious degree-2 decision trees using directional affine dispersers. We show that proving these lower bounds and similar maximal lower bounds against (non-oblivious) degree-2 decision trees are necessary in order to construct dispersers for variety sources and obtain improved circuit lower bounds.

1.1 Prior and Related work

We mention a related line of work that seeks to find algebraic structure in local computation: This arises in extractor theory, where the goal is to transform entropic distributions generated by local

[1] This model was first studied by Chistopolskaya and Podolskii and they focused on 2-local decision trees, and their connections to communication complexity [CP22].

computations into uniform distribution [DW12; Vio14; ACGLR22].

We now summarize several prior lower bounds for local decision trees that are immediate from prior work, or that have previously appeared in the literature.

We begin with a simple observation that for any function that depends on all n variables, the depth of an ℓ-local decision tree computing it is at least $\frac{n}{\ell}$. Therefore, to measure the non-triviality of a function, we introduce the following quantity, defined for both oblivious and non-oblivious local decision trees.

Definition 1.1 (Depth overhead)**.** *For a function $f : \{0,1\}^n \to \{0,1\}^m$, we define its (oblivious)* depth overhead *as*

$$\max_{1 \le \ell \le n} \frac{d_\ell(f)}{(n/\ell)} ,$$

where $d_\ell(f)$ is the minimum depth of an (oblivious) ℓ-local decision tree computing it.

This quantity measures the multiplicative overhead over the trivial bound of n/ℓ and we will use it as a benchmark to compare and contextualize results across different values of ℓ.[2]

Lower bounds against (non-oblivious) ℓ-local decision trees. We can obtain lower bounds against ℓ-local decision trees using the connection to depth-3 circuits mentioned previously. The best lower bound against Σ_3^ℓ against an explicit function is of size $2^{cn/\ell}$ where $c \simeq 1.282$ [PPSZ05] and only for $\ell = 2$, we have $2^{n-o(n)}$ lower bounds [PSZ00]. The explicit functions one can use in both cases are indicators of good codes. Also in the last case, the hard explicit function can be an affine disperser for min-entropy $o(n)$. These lower bounds directly yield depth lower bounds of size cn/ℓ and $n - o(n)$ for ℓ-local and 2-local decision trees, respectively.

Chistopolskaya and Podolskii studied the model of 2-local decision trees and showed that any 2-local decision tree computing Majority must have depth at least $n - o(n)$ [CP22].

Hence, the best depth overhead for ℓ-local decision trees is $O(1)$.

Lower bounds against oblivious ℓ-local decision trees. Oblivious ℓ-local decision trees were studied (under different terminology) in the works of Hrubeš and Rao, Lecomte et al., as well as Bessonov and Podolskii [HR15; LRT22; BP26]. Hrubeš and Rao showed that there exists an explicit function such that any oblivious $(1-\gamma)n$-local decision tree computing it, requires depth at least $n^{c\gamma}$ where $\gamma \in (0,1)$ is an arbitrary constant and $c \simeq 0.15$ is a universal constant [HR15]. Hence, this function has depth overhead of about $n^{0.15}$. Their proof strategy was inspired by Nechiporuk's lower bound on formula size [Nec66].

Lecomte et al. showed that any oblivious ℓ-local decision tree computing the Majority function requires depth at least $\frac{n}{\ell} \cdot \log(\ell)$ (provided $\ell \le n^{0.99}$), obtaining depth overhead of $\log(n)$ [LRT22]. They used information-theoretic arguments to obtain this lower bound. Using the connection to oblivious branching programs, they recovered the classical lower bound [AM86; BPRS90] that any constant-width oblivious branching program computing Majority requires length $\ge \Omega(n \log(n))$ using vastly different techniques.

Bessonov and Podolskii [BP26] proved that a special case of oblivious 3-local decision trees computing Majority also require depth $n - o(n)$.[3]

[2] Lecomte et al. also introduced this quantity as composition overhead, which is equivalent to depth overhead for oblivious local decision trees [LRT22].

[3] In this special case, the functions computed in the nodes of the oblivious decision tree are depth-2 decision trees themselves.

1.2 Our results

Our main conceptual contribution is that the task of proving lower bounds for many varied circuit models can be reduced to proving lower bounds against local decision trees. And for many such models, it is necessary to do so. We provide these connections in detail in Section 2. We view our strong lower bounds against oblivious local decision trees as progress towards the task of proving such strong lower bounds against local decision trees.

We obtain lower bounds in 3 different settings: 1) Against oblivious local decision trees outputting 1 bit. 2) Against oblivious local decision trees outputting many bits. 3) Against oblivious degree-2 decision trees outputting 1 bit. The hard functions for our lower bounds in these three settings are: sumset dispersers, sumset condensers, and directional affine dispersers, respectively. As far as we are aware, this is the first instance of a lower bound that uses the full powers of these pseudorandom objects to make progress towards frontier circuit lower bounds.[4] Our lower bounds will be obtained by finding sumset structure in local computation. Such a proof technique has not appeared in previous works and we hope it leads to interesting lower bounds against (non-oblivious) local decision trees and further circuit lower bounds.

We now present our results in each of the three settings below.

Single-output oblivious local decision trees. Our main result in this setting is as follows.

Theorem 1. *There exists a polynomial-time computable function* $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ *and a constant* $C > 0$ *such that any oblivious* ℓ*-local decision tree computing* Disp *has depth at least*

$$\frac{n^2}{n + C\ell^2 \log(n)} .$$

In particular, for $\ell = o(\sqrt{n/\log(n)})$*, any oblivious* ℓ*-local decision tree computing* Disp *has depth* $n - o(n)$*.*

Thus, our depth overhead here is $\sqrt{n/\log(n)}$, much larger than the depth overhead obtained in previous works by Hrubeš and Rao as well as Lecomte et al. Our explicit hard function Disp is a sumset disperser and has the property that for any $A, B \subseteq \{0,1\}^n$ with $|A|, |B| \geq \mathrm{poly}(n)$, it holds that $\mathsf{Disp}(A + B)$ is non-constant. Such a function Disp (with even stronger guarantees) was recently constructed by Li [Li23]. These dispersers strictly generalize affine dispersers which were used as the hard function in proving unrestricted-depth circuit lower bounds [DK11; FGHK16; LY22]. As far as we are aware, this is the first instance of a lower bound that uses the full power of a sumset disperser.

Multi-output oblivious local decision trees. We first mention that for functions that output m bits and whose image is $\{0,1\}^m$, we trivially obtain the lower bound of m on the depth of a local decision tree computing it. Our result will have the property that $d = \omega(m)$ and hence it will be non-trivial.

We now state our nearly maximal lower bounds in this setting.

[4] Directional affine dispersers and sumset dispersers have been used to obtain lower bounds against linear branching programs [GPT22; CL23; LZ24a; LZ24b].

Theorem 2. *For every $\ell \leq n/\operatorname{poly}(\log(n))$, there exists $m = \ell \cdot \operatorname{poly}(\log(n))$ and a polynomial-time computable function $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ such that any oblivious ℓ-local decision tree computing Cond has depth $n - o(n)$.*

Thus, our depth overhead here is $n/\operatorname{poly}(\log(n))$, which is nearly optimal. Our explicit hard function Cond is a condenser for sumset sources and has the property that for any $A, B \subseteq \{0,1\}^n$ where $|A| \geq |B| \geq 2^{\operatorname{poly}(\log(n))}$, the size of the image $\mathsf{Cond}(A+B)$ is at least $2^{m-o(m)}$.[5] Once again, as far as we are aware, this is the first instance of a lower bound that uses such pseudorandom properties of the hard function. To get a sense of parameters, we will have that $|A| \gg |B|$ and $m \simeq 0.99 \log(|A|)$.

We also emphasize that a much weaker version of our result for (non-oblivious) local decision trees, provided $d = \omega(m)$, will lead to breakthrough lower bounds against Valiant's model and hence against frontier circuit families: For instance, a lower bound of $0.01n$ on the depth for any growing $\ell = \omega(1)$ would suffice (see Section 2.2 for further details).

While we do have excellent extractors (whose output distribution is close to uniform) for sumset sources [CL22; Li23], the number of bits outputted by these constructions for (n, k_1, k_2) sumset sources (see Definition 3.18), when $k_1 \gg k_2$, is proportional to k_2 and not k_1. Therefore, as an additional result, we construct a sumset condenser that can get most of the entropy out of such sumset sources. See Theorem 5.1 for a formal statement. While constructing such a condenser, we needed to use chain rule for smooth min-entropy, a natural extension of the chain rule for min-entropy; we are not aware of an explicit reference for this and provide a self contained proof that we hope proves to be useful for future works (see Lemma 3.11 for details).

Oblivious degree-2 decision trees. In this setting, again, we obtain maximal lower bounds.

Theorem 3. *There exists a polynomial-time computable function $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ such that any oblivious degree-2 decision tree computing Disp has depth at least $n - O(\log(n))$.*

While there is no notion of depth overhead for these trees, we find it quite surprising that we can obtain such strong lower bounds, with such a small additive gap to n. Our hard function Disp here is a directional affine disperser which has the property that for all $a \neq 0^n$, the function $f_a(x) = \mathsf{Disp}(x) + \mathsf{Disp}(x+a)$ is an affine disperser for dimension $O(\log(n))$ (see Definition 3.16). Such a disperser (with even stronger guarantees) was recently constructed by Li and Zhong [LZ24b].

1.3 Proof overview

In this section, we provide an overview of each of our lower bound arguments as well as the construction of the sumset condenser.

1.3.1 Single-output oblivious local decision trees

We sketch the proof of Theorem 1. For clarity, we fix $\ell = n^{0.49}$ and prove that any ℓ-local decision tree computing a *sumset disperser* has depth $d \geq 0.99n$. At a high level, we use a separator-based argument to find sumset structure inside the local computation. As mentioned above, our

[5] These objects are also referred to as a disperser but in this work, we will only refer to dispersers as functions that have non-constant output image. Sumset condensers satisfy a stronger property but for our lower bounds, this property suffices. Moreover, our explicit construction of Cond will satisfy the standard properties of a condenser.

explicit hard function $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ is chosen so that for any $A, B \subseteq \{0,1\}^n$ with $|A|, |B| \geq \text{poly}(n)$, we have $\mathsf{Disp}(A+B) = \{0,1\}$. Given a low-depth tree T computing Disp, we will construct A and B that violate this property.

Let T be an oblivious ℓ-local decision tree of depth $d = 0.99n$ computing Disp. We can associate with T an ℓ-local map $L : \{0,1\}^n \to \{0,1\}^d$ and a function h such that

$$T(x) = h(L(x)) \qquad \text{for all } x \in \{0,1\}^n.$$

We will construct sets A, B with $|A|, |B| \geq \text{poly}(n)$ such that

$$L(a) = L(a+b) \qquad \text{for all } a \in A,\ b \in B.$$

Concluding the theorem from such A and B. Partition A into A_0 and A_1, where $a \in A_0$ if $h(L(a)) = 0$, and $a \in A_1$ otherwise. Without loss of generality, $|A_0| \geq |A|/2 \geq \text{poly}(n)$. Then for every $a_0 \in A_0$ and $b \in B$, $h(L(a_0+b)) = h(L(a_0)) = 0$, and, hence, $T(A_0+B) = h(L(A_0+B)) = \{0\}$, contradicting that T computes the sumset disperser Disp.

Constructing A and B. Let $\alpha = O(\log n)$. Order the n input variables by the number of functions L_i in which they appear, in non-decreasing order, and let $I \subseteq [n]$ be the set of the first α variables in this order. Since there are d many ℓ-local functions, the variables in I appear in at most $\frac{\alpha d \ell}{n} \leq \alpha\ell$ many functions L_i. In particular, there are at most $\alpha\ell^2 = o(n)$ variables that appear in some function L_i together with a variable from I. Let $S \subseteq [n]$ be the set of all such variables, note that $|S| = o(n)$. And let $R = [n] \setminus (I \cup S)$ be the set of the remaining variables, satisfying $|R| \geq n - \alpha - \alpha\ell^2 = n - o(n)$.

By construction, S *separates* I from R: no function L_i depends on variables from both I and R simultaneously. Let d_1 be the number of output functions L_i that depend only on variables from $I \cup S$, and let $Q_1 : \{0,1\}^{|I|+|S|} \to \{0,1\}^{d_1}$ be the corresponding submap of L. Let $d_2 = d - d_1$, and let $Q_2 : \{0,1\}^{|R|+|S|} \to \{0,1\}^{d_2}$ be the remaining submap, whose outputs depend only on variables from $R \cup S$. Hence, Q_1, Q_2 partition the output bits of L and we can write $L(x) = (Q_1(x_I, x_S), Q_2(x_I, x_S))$.

Fix the variables in S to $0^{|S|}$, and define the restricted maps

$$Q_1' = Q_1(\cdot, 0^{|S|}) \colon \{0,1\}^{|I|} \to \{0,1\}^{d_1},$$
$$Q_2' = Q_2(\cdot, 0^{|S|}) \colon \{0,1\}^{|R|} \to \{0,1\}^{d_2}.$$

Let $z^* \in \{0,1\}^{d_2}$ be a value with the largest preimage under Q_2', and let $Y(z^*) = \{\, y \in \{0,1\}^{|R|} : Q_2'(y) = z^* \,\}$. We pick any arbitrary $y^* \in Y(z^*)$ and define our sets A, B as follows: Let $A = A_S \times A_I \times A_R$ and $B = B_S \times B_I \times B_R$, where

$$A_S = \{0^{|S|}\}, \qquad A_I = \{0,1\}^{|I|}, \qquad A_R = \{y^*\};$$
$$B_S = \{0^{|S|}\}, \qquad B_I = \{0^{|I|}\}, \qquad B_R = Y(z^*) + y^*.$$

Verifying $L(a) = L(a+b)$. We claim that $L(a) = L(a+b)$ for all $a \in A$ and $b \in B$. Using the decomposition of L into Q_1 and Q_2 and the fact that $A_S = B_S = \{0^{|S|}\}$, it suffices to check the claim for Q_1' and Q_2'. For our choices of A and B, we indeed have

$$Q_1'(a+b) = Q_1'(a + 0^{|I|}) = Q_1'(a),$$
$$Q_2'(a+b) = Q_2'(y^* + (y + y^*)) = z^* = Q_2'(a)$$

for a $y \in Y(z^*)$.

Sizes of A and B. We now show A and B are large enough. First, $|A| \geq 2^{\alpha} \geq \text{poly}(n)$ by the choice of $\alpha = O(\log n)$. And second, $|B| = |Y(z^*)| \geq 2^{|R|-d_2} \geq 2^{|R|-d}$. From our assumption on $|R|$, we infer that $|B| \geq 2^{n-o(n)-d}$. As $d = 0.99n$, we conclude that $|B| \geq 2^{0.01n-o(n)} \geq \text{poly}(n)$, concluding our result.

1.3.2 Multi-output oblivious local decision trees

We sketch the proof of Theorem 2. For clarity, we fix $\ell = n^{0.99}$ and $m = \widetilde{O}(\ell)$, and we prove that any ℓ-local decision tree computing a *sumset condenser* $\mathsf{Cond}\colon \{0,1\}^n \to \{0,1\}^m$ has depth $d \geq 0.99n$.

We use the following property of a sumset condenser. For any $A, B \subset \{0,1\}^n$ with $|A| \geq 2^{\widetilde{\Omega}(\ell)}$ and $|B| \geq 2^{\text{poly}(\log(n))}$, it holds that $|\mathsf{Cond}(A+B)| \geq 2^{m-o(m)}$. We sketch our construction of an explicit sumset condenser Cond in Section 1.3.4

Let T be an oblivious ℓ-local decision tree of depth d computing Cond. As before, we associate with T an ℓ-local map $L : \{0,1\}^n \to \{0,1\}^d$ and a function h such that for all $x \in \{0,1\}^n$, $T(x) = h(L(x))$. We will find large sets A, B such that $L(A+B)$ has small image. This implies that $T(A+B) = h(L(A+B))$ also has small image, contradicting the condenser property.

Constructing A and B. Let $z^* \in \{0,1\}^d$ be a value with the largest preimage under L, and let $A = \{\, a \in \{0,1\}^n : L(a) = z^* \,\}$. We have that $|A| \geq 2^{n-d} = 2^{0.01n}$. On average, an input variable appears in at most $d\ell/n \leq \ell$ many functions L_i. For simplicity, assume that each variable appears in at most ℓ functions. In this case, we let B be the Hamming ball of radius $O(\text{poly}(\log(n))$.[6]

Concluding the theorem from such A and B. We claim that for any $y_0 \in A+B$, the Hamming distance between z^* and $L(y_0)$ is at most $\ell \cdot \text{poly}(\log(n))$. Showing this suffices to prove the theorem since then the output of $L(A+B)$ will lie within a ball of radius $r = \ell \cdot \text{poly}(\log(n))$ around z^*, showing that $|T(A+B)|$ is at most $n^r = 2^{r\log(n)} \ll 2^{m-o(m)}$, contradicting the condenser property.

Hamming distance between z^* and $L(A+B)$. We now show that for any $y_0 \in A+B$, the Hamming distance between z^* and $L(y_0)$ is at most $\ell \cdot \text{poly}(\log(n))$. As B is a Hamming ball of radius $\text{poly}(\log(n))$, there exists some $y \in A$ such that the Hamming distance between y and y_0 is at most $\text{poly}(\log(n))$. Since we assumed that each input variable appears in at most ℓ functions, $L(y)$ and $L(y_0)$ differ in at most $\ell \cdot \text{poly}(\log(n))$ positions, and so the Hamming distance between $z^* = L(y)$ and $L(y_0)$ is at most $\ell \cdot \text{poly}(\log(n))$ as desired.

1.3.3 Oblivious degree-2 decision trees

We sketch the proof of Theorem 3. As mentioned earlier, our hard function $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ will be a *directional affine disperser*: for every non-zero $a \in \{0,1\}^n$ and every affine subspace S of dimension at least $\Omega(\log(n))$, $\mathsf{Disp}(x+a) - \mathsf{Disp}(x)$ is not constant on S.

Let T be an oblivious degree-2 decision tree of depth $d = 0.99n$ computing Disp. We can write $T(x) = h(Q(x))$, where $Q\colon \{0,1\}^n \to \{0,1\}^d$ is a degree-2 map and $h : \{0,1\}^d \to \{0,1\}$ is a

[6] In the general case, we set B to be an appropriate subset of this Hamming ball.

function. We will find an affine subspace S of dimension at least $n - d \geq \Omega(\log(n))$ and a direction $a \in \{0,1\}^n$ such that $Q(x) = Q(x+a)$ for all $x \in S$. It will then follow that

$$T(x) = h(Q(x)) = h(Q(x+a)) = T(x+a) \qquad \text{for all } x \in S\ ,$$

contradicting the disperser property.

Constructing S and a. It remains to construct such an affine subspace S and direction a. Since $d < n$, there exist distinct $u, v \in \{0,1\}^n$ with $Q(u) = Q(v)$. We let $a = u + v$ and consider $L_a(x) = Q(x) + Q(x+a)$. As Q is a degree-2 map, L_a must be a degree-1 map. By our choice of u and v, we see that $L_a(u) = Q(u) + Q(v) = 0^d$ and hence 0^d lies in the image of the affine map L_a. Therefore, $S = L_a^{-1}(0^d)$ is a non-empty affine subspace of dimension at least $n - d$. Then, for all $x \in S$, we have $L_a(x) = Q(x) + Q(x+a) = 0^d$ and hence $Q(x) = Q(x+a)$ for all $x \in S$, as desired.

1.3.4 Constructing explicit sumset condenser

We here sketch the construction of our explicit sumset condenser Cond. Throughout this section, we assume the reader is familiar with notions of statistical distance, min-entropy, smooth min-entropy, sumset source, seeded extractor, and data processing inequality. See Section 3.3, Section 3.4, Section 3.7, and Section 3.8 for the relevant definitions.

We require a condenser $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ for (n, k_1, k_2) sumset sources where $k_1 = \frac{n}{\log(\log(n))}, k_2 = \text{poly}(\log(n)), m = 0.99k_1$ such that for any such source $\mathbf{X}$, $H^{\varepsilon}_{\infty}(\mathsf{Cond}(\mathbf{X})) \geq m - o(m)$ for $\varepsilon = 0.01$.

To construct our explicit sumset condenser Cond with these desired properties, we rely on previous constructions of excellent extractors $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^t$ for (n, k, k) sumset sources for any $k \geq \text{poly}(\log(n))$. This extractor will have the property that $(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X})) \approx_{n^{-0.01}} (\mathbf{X}_1, \mathbf{U}_t)$ where $t = k^{0.01}$ [CL22].

Ideally, given such Ext, we would have liked to output $\mathsf{sExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X}))$ where sExt is a seeded extractor which can output $0.99k_1$ bits; this is because by data processing inequality, $\mathsf{sExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X})) \approx_{\varepsilon} \mathsf{sExt}(\mathbf{X}_1, \mathbf{U}_t) \approx_{\varepsilon} \mathbf{U}_m$, concluding the construction. Such techniques have been deployed previously in constructions of two source extractors [Li16]. However, since we only have access to $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$, we are unable to do so. Therefore, we exploit the fact that $\mathbf{X}_2$ has much smaller entropy and construct a condenser.

To construct our condenser Cond, we use standard construction of a linear seeded extractor $\mathsf{LExt} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ for min-entropy $0.999k_1$ distributions where $m = 0.99k_1$. This has the property that $\mathsf{LExt}(\cdot, y)$ is a linear function of the input for all $y \in \{0,1\}^t$.[7] With this, our construction of Cond will be simply $\mathsf{Cond}(\mathbf{X}) = \mathsf{LExt}(\mathbf{X}, \mathsf{Ext}(\mathbf{X}))$.

Using linearity of the seeded extractor, we can write $\mathsf{LExt}(\mathbf{X}, \mathsf{Ext}(\mathbf{X})) = \mathsf{LExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X})) + \mathsf{LExt}(\mathbf{X}_2, \mathsf{Ext}(\mathbf{X}))$. As argued earlier, we will have that $\mathsf{LExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X}))$ is statistically close to $\mathbf{U}_m$. To finish our analysis, we observe that since $k_2, t \ll m$, the size of support of $\mathsf{LExt}(\mathbf{X}_2, \mathsf{Ext}(\mathbf{X}))$ is very small. Using chain rule for smooth min-entropy, we will be able to show that for most fixings of $\mathsf{LExt}(\mathbf{X}_2, \mathsf{Ext}(\mathbf{X}))$, the distribution $\mathsf{LExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X}))$ will have high smooth min-entropy. For every such "good" fixing z_2 of $\mathsf{LExt}(\mathbf{X}_2, \mathsf{Ext}(\mathbf{X}))$, our output distribution will be $z_2 + \mathsf{LExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X})) \,|\, (\mathsf{LExt}(\mathbf{X}_2, \mathsf{Ext}(\mathbf{X})) = z_2)$. As z_2 is a fixed string and

[7] Linear seeded extractors have proven to be incredibly helpful in constructions of seedless extractors for affine sources, and sumset sources.

$\mathsf{LExt}(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X})) \,|\, (\mathsf{LExt}(\mathbf{X}_2, \mathsf{Ext}(\mathbf{X})) = z_2)$ has high smooth min-entropy, we conclude that our overall output distribution will have high smooth min-entropy, as desired.

As mentioned earlier, we are not aware of an explicit reference for the chain rule for smooth min-entropy; hence we provide a self-contained proof of this. See Lemma 3.11 for further details.

Organization We explicitly map out the connections to various circuit models in Section 2. We include the necessary preliminaries in Section 3. In Section 4 we prove our lower bound for single-output oblivious local decision trees. In Section 5 we prove the lower bound for multi-output oblivious local decision trees, and in Section 6 we construct the required sumset condenser. Finally, in Section 7 we show the lower bound for oblivious degree-2 decision trees . We lay out many open directions in Section 8.

2 Connections to other circuit complexity models

2.1 Depth-3 circuits.

The closest connection to local decision trees is with Σ_3^ℓ circuits—these are $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{OR}_\ell$ circuits where ℓ is the bottom fan-in. Equivalently, these circuits compute an OR of ℓ-CNFs. We can easily see that if a function f can be computed by an ℓ-local decision tree of depth d, then it can also be computed by a Σ_3^ℓ circuit of size $2^{d+\ell}$. This means that lower bounds against these depth-3 circuits yield lower bounds against ℓ-local decision trees. For a lot of other circuit models, the task of proving non-trivial lower bounds can be reduced to proving near maximal lower bounds against depth-3 circuits. One of our main conceptual contributions is that all known reductions from other circuit models to depth-3 circuits can instead be made to ℓ-local decision trees where a size $\geq s$ lower bound requirement against depth-3 circuit often translates to that of proving depth $\geq \log(s)$ lower bound against ℓ-local decision trees. We will go over various computational models below specifying these exact connections.

We further believe that proving lower bounds against ℓ-local decision trees can be a much simpler task than proving lower bounds against Σ_3^ℓ circuits. For instance, for ℓ-local decision trees, it is trivial to see that computing parity requires depth n/ℓ and that this is tight. However, proving this for Σ_3^ℓ circuits was an open problem and only after extensive efforts, was the tight bound of $2^{n/\ell}$ obtained [PPZ99]. Another example illustrating this is that Chistopolskaya and Podolskii showed that a 2-local decision tree computing Majority must have depth at least $n - o(n)$ [CP22]. This is in contrast to the case of Σ_3^2 circuits where the size of optimal circuit computing Majority is $2^{n/2} \cdot \mathrm{poly}(n)$ [HJP95], yielding a trivial lower bound of $n/2 - o(n)$ on the depth of any 2-local decision tree computing Majority.

2.2 Valiant's model.

Valiant showed that to prove super-linear circuit lower bounds against log-depth circuits as well as against series-parallel circuits (these are still unconquered frontiers in circuit complexity), it suffices to prove strong lower bounds against a new model of computation that he introduced—Valiant's model [Val77]. He further showed that to prove lower bounds against his model, it suffices to prove strong lower bounds against Σ_3^ℓ circuits.

We observe that ℓ-local decision trees sandwich perfectly in between these depth-3 circuits and Valiant's model—that such depth-3 circuit lower bounds yield lower bounds against ℓ-local decision

trees; and that those exact lower bounds against ℓ-local decision trees yield the same lower bound against Valiant's model, yielding super-linear circuit lower bounds. We define Valiant's model and provide exact parameter translation in Section 3.2.

From these connections, we obtain the following consequences from lower bounds on local decision trees. First, finding an explicit function that requires depth $\Omega(n)$ against ℓ-local decision trees for any $\ell \in \omega(1)$ yields super-linear lower bounds against series-parallel circuits. And finding an explicit function that requires depth $\omega(\frac{n}{\log(\log(n))})$ against $\ell = n^{\varepsilon}$-local decision trees, for any constant $\varepsilon > 0$, yields super-linear lower bounds against log-depth circuits. In fact, for these lower bounds, it suffices to obtain lower bound against ℓ-local decision trees outputting many—say, m bits, provided that $m \leq o(d)$.

In fact, above, instead of proving lower bounds against ℓ-local decision trees, it suffices to prove all these lower bounds against semi-oblivious local decision trees: ℓ-local decision trees where all nodes at the same level are (potentially different) functions over the same set of ℓ variables (see Section 3.1 for formal treatment).

2.3 Unrestricted-depth circuit lower bounds.

The work [GKW21] showed that to improve the state of the art lower bounds for unrestricted-depth circuits (circuits with arbitrary fan-in 2 gates and no depth restriction) from current state of the art of $3.1n - o(n)$ to $3.9n - o(n)$, it suffices to obtain a lower bound of size $2^{n-o(n)}$ on Σ_3^{16} circuits. We observe from their proof that it suffices to obtain a size lower bound of $2^{n-o(n)}$ against 16-local decision trees, a simpler task. Also from earlier discussion, we have that $2^{n-o(n)}$ lower bounds on Σ_3^{16} circuits yield size lower bound of $2^{n-o(n)}$ against 16-local decision trees, demonstrating that the latter is an easier task.

2.4 Dispersers for quadratic varieties.

The work [GK16] showed that to improve these unrestricted-depth circuit lower bounds from $3.1n - o(n)$ to $3.11n$, it suffices to explicitly construct the following pseudorandom function—a disperser for degree-2 varieties of min-entropy $o(n)$ (see Definition 3.13). We observe that the task of proving lower bounds against degree-2 decision trees is an easier task than that of constructing a degree-2 variety disperser (see Section 3.5 for further details). In particular, any degree-2 decision tree computing such a disperser with min-entropy requirement $o(n)$ would require depth $n - o(n)$. Since constructing such dispersers seems to be beyond the reach of current techniques, we hope that the tools developed to obtain lower bounds against degree-2 decision trees can help inspire improved constructions of such dispersers.

2.5 Oblivious Branching programs.

An oblivious branching program is a layered branching programs where all nodes in the same layer read the same variables. It has been observed by previous works that if a function can be computed by a width-w length-t oblivious branching program, then it can be computed by an oblivious ℓ-local decision tree of depth $w \log(w) \cdot t/\ell$ [HR15; LRT22]. Hence, lower bounds against ℓ-local decision trees result in lower bounds against such branching programs.

3 Preliminaries

For a positive integer n, by $[n]$ we denote the set $\{1, \ldots, n\}$. All logarithms are base 2, i.e., $\log(2^n) = n$. Let $f\colon \{0,1\}^n \to \{0,1\}^m$ be a function. For a set $X \subseteq \{0,1\}^n$, $f(X)$ denotes the image of X under f, and for $y \in \{0,1\}^m$, $f^{-1}(y)$ denotes the preimage of y under f. We will use boldface font to denote random variables. We will sometimes abuse notation and interchangeably use the terms random variables, distributions, and sources. For a distribution $\mathbf{X}$, we write $\mathrm{Supp}(\mathbf{X})$ to denote its support. We will use $\mathbf{U}_n$ to denote the uniform distribution over $\{0,1\}^n$.

For an integer $\ell \geq 2$, an ℓ-uniform hypergraph is a hypergraph where each edge contains exactly ℓ distinct vertices. For a subset of vertices S of a hypergraph H over vertex set V, the neighborhood $\mathsf{Nbr}(S)$ is the set of all vertices adjacent to a vertex from S, $\mathsf{Nbr}(S) = \{v \in V : \exists s \in S, \exists e \in H, \text{ s.t. } v, s \in e\}$.

3.1 $\mathcal{F}$-decision trees

We first define the general notion of $\mathcal{F}$-decision trees and will specialize it to local and low-degree decision trees.

Definition 3.1 ($\mathcal{F}$-decision tree)**.** *Let $n, m \in \mathbb{N}$ and let $\mathcal{F}$ be a class of Boolean functions over n variables. An* $\mathcal{F}$ decision tree *$T : \{0,1\}^n \to \{0,1\}^m$ is a binary tree where each internal node is marked by a Boolean function $f \in \mathcal{F}$ over the n input variables with two outgoing edges marked with 0 or 1 (the outcome of the function), and each leaf node marked by an outcome from $\{0,1\}^m$.*

The depth of an $\mathcal{F}$-decision tree is the depth of the underlying binary tree.

We will study two specific instantiations of these trees: (i) *ℓ-local decision trees* where $\mathcal{F}$ is the family of all ℓ-local Boolean functions; and (ii) *degree-d decision trees* where $\mathcal{F}$ is the family of multivariate degree-d polynomials over $\mathbb{F}_2$.

We specialize this definition further to define the oblivious version of these trees.

Definition 3.2 (Oblivious $\mathcal{F}$-decision tree)**.** *Let $\mathcal{F}$ be a class of Boolean functions and let T be an $\mathcal{F}$-decision tree. We say T is an* oblivious $\mathcal{F}$-decision tree *if all nodes at the same level of the tree are marked by the exact same function from $\mathcal{F}$.*

We can characterize oblivious $\mathcal{F}$-decision trees using a composition-based definition as well. In particular, for any oblivious $\mathcal{F}$-decision tree $T : \{0,1\}^n \to \{0,1\}^m$ of depth d, we can associate an $\mathcal{F}$-map $P : \{0,1\}^n \to \{0,1\}^d$ (where each output function $P_i \in \mathcal{F}$) and a function $h : \{0,1\}^d \to \{0,1\}^m$ (of arbitrary complexity) such that $T(x) = h(P(x))$.

Here, at each node at level i, the query of the decision tree is the function P_i, and the label of each leaf is dictated by the function h.

We will also consider semi-oblivious ℓ-local decision trees defined as follows.

Definition 3.3 (Semi-Oblivious local decision tree)**.** *Let T be an ℓ-local decision tree. We say that T is a* semi-oblivious ℓ-local decision tree *if all nodes at level i are marked by (possibly different) ℓ-local functions of the same ℓ variables.*

3.2 Valiant's model

We define Valiant's model as follows:

Definition 3.4 (Valiant's model [Val77])**.** *For $\ell, m, n, t \in \mathbb{N}$, we say a function $f : \{0,1\}^n \to \{0,1\}^m$ is computable in Valiant's model with t intermediate bits and locality ℓ if the following holds. There exist t intermediate functions $h_1, \ldots, h_t$ where each $h_i \in \{0,1\}$ is a function of the previous bits $h_1, \ldots, h_{i-1}$ and ℓ input bits. Furthermore, for all $i \in [m]$, the output bit f_i is a function of the intermediate bits $h_1, \ldots, h_t$ and ℓ input bits.*

Valiant [Val77] showed that lower bounds in this model could be used to prove breakthrough circuit lower bounds. In particular, a lower bound of $t \geq \Omega(n)$ for any $\ell \in \omega(1)$ would imply a super-linear lower bound on the size of series-parallel circuits. Similarly, a lower bound of $t \geq \omega\left(\frac{n}{\log(\log(n))}\right)$ for $\ell = n^{\varepsilon}$ would imply a super-linear lower bound on the size of log-depth circuits.

Below we show that if f is computable in Valiant's model, then it is also computable by semi-oblivious local decision trees.

Claim 3.5. *Let $\ell, m, n, t \in \mathbb{N}$, and let $f : \{0,1\}^n \to \{0,1\}^m$ be computable by Valiant's model with t intermediate bits with locality ℓ. Then, there exists an ℓ-local semi-oblivious tree of depth $t + m$ computing f.*

Proof. Let the intermediate bits be $h_1, \ldots, h_t$. For $i \in [t]$, we let the nodes at level i compute the output of the intermediate bit h_i. This is possible since for a node at level i, the path from root to that node fixes the values of all previous intermediate bits, leaving h_i to be an ℓ-local function of the input. Then for $i \in [m]$, we let the nodes at level $t+i$ compute the output of bit f_i. Again, this is possible since after fixing the intermediate bits, f_i is an ℓ-local function of the input. Finally, we label each leaf with the path taken from level t to the leaf. □

This implies that for a function $f : \{0,1\}^n \to \{0,1\}^m$, if any semi-oblivious ℓ-local decision tree computing f has depth d, then it requires at least $d - m$ intermediate bits to be computed in Valiant's model.

In particular, for any $\ell = \omega(1)$, a lower bound of $d \geq \Omega(n)$ on the depth of semi-oblivious ℓ-local decision tree would lead to a super-linear lower bound for series-parallel circuits.

3.3 Statistical distance

Definition 3.6 (Statistical distance)**.** *Let $\mathbf{X}, \mathbf{Y} \sim \Omega$. The* statistical distance *between $\mathbf{X}$ and $\mathbf{Y}$ is defined as*

$$|\mathbf{X} - \mathbf{Y}| = \max_{S \subset \Omega} (\Pr[\mathbf{X} \in S] - \Pr[\mathbf{Y} \in S]) = \frac{1}{2} \sum_{s \in \Omega} |\Pr[\mathbf{X} = s] - \Pr[\mathbf{Y} = s]| \ .$$

If $|\mathbf{X} - \mathbf{Y}| \leq \varepsilon$, then we say that $\mathbf{X}$ is ε-close to $\mathbf{Y}$, and denote this by $\mathbf{X} \approx_{\varepsilon} \mathbf{Y}$. We will use the fact that statistical distance satisfies the triangle inequality. We will also use the following standard result regarding statistical distance.

Lemma 3.7 (Data processing inequality)**.** *For any random variables $\mathbf{X}, \mathbf{Y} \sim \Omega_1$ and any function $f : \Omega_1 \to \Omega_2$,*

$$|\mathbf{X} - \mathbf{Y}| \geq |f(\mathbf{X}) - f(\mathbf{Y})| \ .$$

3.4 Min-entropy

We will use the standard notion of min-entropy to measure the amount of randomness in a random variable.

Definition 3.8. *For a random variable $\mathbf{X} \sim \Omega$, we define its* min-entropy *as*

$$H_\infty(\mathbf{X}) = \min_{x \in \Omega} \log(1/\Pr[\mathbf{X} = x]) \ .$$

We say $\mathbf{X}$ is an (n, k) *source* if $\mathbf{X} \sim \{0,1\}^n$ and $H_\infty(\mathbf{X}) = k$. We will also use the following notion of smooth min-entropy. We say that $\mathbf{X}$ is a *flat* (n, k) *source* if $\mathbf{X}$ is a uniform distribution over a set $S \subseteq \{0,1\}^n$ of size $|S| = 2^k$ (provided that $2^k \in \mathbb{N}$).

Definition 3.9. *For a random variable $\mathbf{X} \sim \Omega$ and $\varepsilon > 0$, we define its* smooth min-entropy *with error ε as*

$$H^\varepsilon_\infty(\mathbf{X}) = \max_{\mathbf{Y} \sim \Omega : |\mathbf{X} - \mathbf{Y}| \leq \varepsilon} \{H_\infty(\mathbf{Y})\} \ .$$

We will use the chain rule for min-entropy.

Lemma 3.10 (Chain rule for min-entropy [MW97])**.** *For any random variables $\mathbf{X} \sim \Omega_X$ and $\mathbf{Y} \sim \Omega_Y$ and $\varepsilon > 0$, we have*

$$\Pr_{y \sim \mathbf{Y}}[H_\infty(\mathbf{X}|\mathbf{Y} = y) \geq H_\infty(\mathbf{X}) - \log(|\mathrm{Supp}(\mathbf{Y})|) - \log(1/\varepsilon)] \geq 1 - \varepsilon \ .$$

We will rely on the chain rule for smooth min-entropy. Since we are not aware of an explicit reference for the following lemma, we supply its proof using standard ideas.

Lemma 3.11 (Chain rule for smooth min-entropy)**.** *For any random variables $\mathbf{X} \sim \Omega_X$ and $\mathbf{Y} \sim \Omega_Y$ and $\varepsilon, \gamma, \delta > 0$, we have*

$$\Pr_{y \sim \mathbf{Y}}\left[H^\gamma_\infty(\mathbf{X}|\mathbf{Y} = y) \geq H^\varepsilon_\infty(\mathbf{X}) - \log(|\mathrm{Supp}(\mathbf{Y})|) - \log(1/\delta)\right] \geq 1 - (\varepsilon + \delta + (2\varepsilon/\gamma)) \ .$$

Proof. Let $\mathbf{X}' \sim \Omega_X$ be such that $H_\infty(\mathbf{X}') = H^\varepsilon_\infty(\mathbf{X})$ and $|\mathbf{X} - \mathbf{X}'| \leq \varepsilon$. Note that we can always pick such an X' satisfying $\mathrm{Supp}(\mathbf{X}) \subset \mathrm{Supp}(\mathbf{X}')$. We then define $\mathbf{Y}' \sim \Omega_Y$ such that for all $x \in \mathrm{Supp}(\mathbf{X})$ and $y \in \mathrm{Supp}(\mathbf{Y})$, $\Pr[\mathbf{Y}' = y|\mathbf{X}' = x] = \Pr[\mathbf{Y} = y|\mathbf{X} = x]$ Also for $x \in \mathrm{Supp}(\mathbf{X}') \setminus \mathrm{Supp}(\mathbf{X})$, we let $(\mathbf{Y}'|\mathbf{X}' = x)$ be uniformly distributed over $\mathrm{Supp}(\mathbf{Y})$. This finishes our description of the distributions $\mathbf{X}'$ and $\mathbf{Y}'$, satisfying $\mathrm{Supp}(\mathbf{X}) \subset \mathrm{Supp}(\mathbf{X}')$ and $\mathrm{Supp}(\mathbf{Y}') = \mathrm{Supp}(\mathbf{Y})$. We also observe that by definition of these random variables, $|(\mathbf{X}, \mathbf{Y}) - (\mathbf{X}', \mathbf{Y}')| \leq |\mathbf{X} - \mathbf{X}'| \leq \varepsilon$. From the data processing inequality (Lemma 3.7), we obtain that $|\mathbf{Y} - \mathbf{Y}'| \leq \varepsilon$.

Let $\mathsf{Bad} = \{y \in \mathrm{Supp}(\mathbf{Y}') : H_\infty(\mathbf{X}'|\mathbf{Y}' = y) \leq H_\infty(\mathbf{X}') - \log(|\mathrm{Supp}(\mathbf{Y}')|) - \log(1/\delta)\}$. Applying the chain rule for min-entropy (Lemma 3.10) to $\mathbf{X}'$ and $\mathbf{Y}'$, we infer that $\Pr[\mathbf{Y}' \in \mathsf{Bad}] \leq \delta$. Hence, we conclude that $\Pr_{y \in \mathbf{Y}}[y \in \mathsf{Bad}] \leq \varepsilon + \delta$.

Next, we define $\mathsf{Far} = \{y \in \mathrm{Supp}(\mathbf{Y}) : |(\mathbf{X}|\mathbf{Y} = y) - (\mathbf{X}'|\mathbf{Y}' = y)| \geq \gamma\}$. We claim that

$\mathbb{E}_{y\in\mathbf{Y}}\left[\left|\,(\mathbf{X}|\mathbf{Y}=y)-(\mathbf{X}'|\mathbf{Y}'=y)\,\right|\right]\le 2\varepsilon$. Indeed, we compute that

$$\begin{aligned}
&\mathbb{E}_{y\sim\mathbf{Y}}\left[\left|\,(\mathbf{X}|\mathbf{Y}=y)-(\mathbf{X}'|\mathbf{Y}'=y)\,\right|\right]\\
&\quad=\sum_{x\in\Omega_X,y\in\Omega_Y}\Pr[\mathbf{Y}=y]\cdot\frac{1}{2}\left|\Pr[\mathbf{X}=x|\mathbf{Y}=y]-\Pr[\mathbf{X}'=x|\mathbf{Y}'=y]\right|\\
&\quad\le\left|\mathbf{Y}-\mathbf{Y}'\right|+\frac{1}{2}\sum_{x\in\Omega_X,y\in\Omega_Y}\left|\Pr[\mathbf{Y}=y]\Pr[\mathbf{X}=x|\mathbf{Y}=y]-\Pr[\mathbf{Y}'=y]\Pr[\mathbf{X}'=x|\mathbf{Y}'=y]\right|\\
&\quad\le\varepsilon+\left|\,(\mathbf{X},\mathbf{Y})-(\mathbf{X}',\mathbf{Y}')\,\right|\\
&\quad=2\varepsilon\,.
\end{aligned}$$

Hence, the claim holds. We now apply Markov's inequality to infer that $\Pr_{y\sim\mathbf{Y}}[y\in\mathsf{Far}]\le\frac{2\varepsilon}{\gamma}$.

Therefore, $\Pr_{y\sim\mathbf{Y}}[y\in\mathsf{Bad}\cup\mathsf{Far}]\le\varepsilon+\delta+\frac{2\varepsilon}{\gamma}$. As $\mathrm{Supp}(\mathbf{Y}')=\mathrm{Supp}(\mathbf{Y})$ and $H_\infty(\mathbf{X}')=H^\varepsilon_\infty(\mathbf{X})$, we infer that for all $y\in\mathrm{Supp}(\mathbf{Y})\setminus(\mathsf{Bad}\cup\mathsf{Far})$, it holds that $(\mathbf{X}|\mathbf{Y}=y)$ has statistical distance at most γ from a distribution with min-entropy at least $H^\varepsilon_\infty(\mathbf{X})-\log(|\mathrm{Supp}(\mathbf{Y})|)-\log(1/\delta)$, as desired. □

3.5 Variety sources

Below we define degree-d variety sources over $\mathbb{F}_2$.

Definition 3.12 (Degree-d variety sources)**.** *For $d,n\in\mathbb{N}$, a degree-d variety source $\mathbf{X}\sim\{0,1\}^n$ is associated with a polynomial map $P=(p_1,\ldots,p_t)\colon\mathbb{F}_2^n\to\mathbb{F}_2^t$ for some $t\in\mathbb{N}$, where each p_i is a multilinear degree-d polynomial of the input. The source $\mathbf{X}$ is uniform over the set of common zeroes of this map: $V=\{x\in\mathbb{F}_2^n:P(x)=0^t\}$.*

We define dispersers for these sources as follows.

Definition 3.13 (Dispersers for degree-d variety sources)**.** *For $d,k,n\in\mathbb{N}$, a function $\mathsf{Disp}:\{0,1\}^n\to\{0,1\}$ is a disperser for degree-d variety sources with min-entropy k, if for all degree-d variety sources $\mathbf{X}$ of min-entropy at least k, it holds that $\mathsf{Disp}(\mathbf{X})=\{0,1\}$.*

In the work of [GK16], it was shown that dispersers for degree-2 variety sources for min-entropy $o(n)$ require unrestricted-depth circuits of size at least $3.11n$, which would improve upon the current state of the art lower bound of $3.1n-o(n)$ for an explicit function [LY22]. However, the current best dispersers for degree-2 variety sources require min-entropy at least $(1-\gamma)n$ for a small absolute constant $\gamma>0$ [LZ19].

We next show that dispersers for degree-t variety sources require large depth to be computed by a degree-t decision tree, thereby establishing that proving lower bounds for degree-t decision trees is an easier task than constructing said dispersers.

Claim 3.14. *Let $\mathsf{Disp}:\{0,1\}^n\to\{0,1\}$ be a disperser for degree-t variety sources with min-entropy requirement k. Then, any degree-t decision tree computing Disp requires depth at least $n-k+1$.*

Proof. Suppose there exists a degree-t decision tree T of depth $d\le n-k$ computing Disp. By an averaging argument, there exists a leaf node L such that at least 2^{n-d} inputs from $\mathbb{F}_2^n$ end up at that leaf. For each node v in the path from root to L, let p_v be the degree t polynomial associated with it. Let q_v be the polynomial that equals p_v if the path from root to L took the zero branch

at node v, and let $q_v = p_v + 1$ otherwise. We observe that the set of inputs that end up at L is exactly described by the variety where each of the polynomials q_v is set to 0. Hence, we have that T is constant over a variety of size at least $2^{n-d} \geq 2^k$, contradicting the fact that it is a disperser for such sources. □

3.6 Affine dispersers and directional affine dispersers

We define affine dispersers as follows.

Definition 3.15 (Affine Disperser)**.** *For $n, k \in \mathbb{N}$, a function $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ is an (n,k)* affine disperser *if for all affine subspaces S of dimension at least k, it holds that $\mathsf{Disp}(S) = \{0,1\}$.*

We will also require the following generalization of affine dispersers.

Definition 3.16 (Directional Affine disperser)**.** *For $n, k \in \mathbb{N}$, a function $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ is an (n,k)* directional affine disperser *if for all directions $a \in \{0,1\}^n \setminus \{0^n\}$, it holds that $\mathsf{Disp}(x) + \mathsf{Disp}(x+a)$ is an (n,k) affine disperser.*

We will utilize the following explicit construction of a directional affine disperser.

Theorem 3.17 ([LZ24b, Theorem 1.4])**.** *There exists a constant $C_0 > 1$ and a polynomial-time computable (n,k) directional affine disperser $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ for any $k \geq C_0 \log n$.*[8]

3.7 Sumsets and sumset sources

For $x, y \in \{0,1\}^n$, $x + y$ denotes the bitwise xor of x and y. For sets $A, B \subseteq \{0,1\}^n$, we define the sumset $A + B = \{a + b\colon a \in A, b \in B\}$. Similarly, for $x \in \{0,1\}^n$ and $A \subseteq \{0,1\}^n$, $x + A$ denotes the set $x + A = \{x + a\colon a \in A\}$.

We define sumset sources as follows.

Definition 3.18. *We say $\mathbf{X} \sim \{0,1\}^n$ is a (n, k_1, k_2)* sumset source *if there exist two independent sources $\mathbf{X}_1, \mathbf{X}_2 \sim \{0,1\}^n$ with $H_\infty(\mathbf{X}_1) = k_1, H_\infty(\mathbf{X}_2) = k_2$ such that $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$.*

We now define sumset dispersers.

Definition 3.19 (Sumset Disperser)**.** *For $n, k \in \mathbb{N}$, a function $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ is an (n,k)* sumset disperser *if for all (n,k,k) sumset sources $\mathbf{X}$, it holds that $\mathrm{Supp}(\mathsf{Disp}(\mathbf{X})) = \{0,1\}$.*

We will use the following explicit construction of a sumset disperser.

Theorem 3.20 ([Li23, Theorem 1.7])**.** *There exists a constant $C > 1$ and a polynomial-time computable (n,k) sumset disperser $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ for any $k \geq C \log n$.*[8]

We also need to define a stronger object—sumset extractors.

Definition 3.21 (Sumset Extractor)**.** *For $n, k \in \mathbb{N}$, a function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is an (n,k)* sumset extractor *with error ε if for all (n,k,k) sumset sources $\mathbf{X}$, it holds that $\mathsf{Ext}(\mathbf{X}) \approx_\varepsilon \mathbf{U}_m$. We furthermore say Ext is strong in the first source if for all sumset sources $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$, it holds that $(\mathbf{X}_1, \mathsf{Ext}(\mathbf{X})) \approx_\varepsilon (\mathbf{X}_1, \mathbf{U}_m)$.*

[8] They obtain a stronger variant of these objects, namely extractors; but we only need dispersers for our applications.

We will use the following construction of a sumset extractor.

Theorem 3.22 ([CL22]). *There exist universal constants $C > 1, \delta > 0, \gamma > 0$ such that for all $n, k, m \in \mathbb{N}$ where $k \geq \log^C(n)$ and $m = k^\delta$, there exists an explicit function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ that is an (n,k)-sumset extractor with error $n^{-\gamma}$ that is strong in the first source.*[9]

We lastly define a relaxation of extractors–condensers.

Definition 3.23 (Sumset condenser). *A function $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ is a $(k_1, k_2, k_{\mathsf{out}})$* sumset condenser *with error ε if for all (n, k_1, k_2) sumset sources $\mathbf{X}$, it holds that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq k_{\mathsf{out}}$.*

In Section 6, we will construct an explicit condenser for sumset sources.

3.8 Seeded extractors and linear seeded extractors

Definition 3.24 (Seeded extractor). *For $n, k, d, m \in \mathbb{N}$ and $\varepsilon > 0$, we say $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a* seeded extractor *for min-entropy k with error ε (or (k,ε)-seeded extractor in short) if for all (n,k) source $\mathbf{X}$:*

$$\mathsf{Ext}(\mathbf{X}, \mathbf{U}_d) \approx_\varepsilon \mathbf{U}_m \ .$$

We say d is the seed length of Ext. Furthermore, we say Ext is a linear seeded extractor *if for all $y \in \{0,1\}^d$, $\mathsf{Ext}(\cdot, y)$ is a linear map.*

We will utilize the following linear seeded extractor.

Theorem 3.25 ([Tre01; RRV02]). *For every $n, k, m \in \mathbb{N}$ and $\varepsilon > 0$, with $m \leq k \leq n$, there exists an explicit linear seeded extractor $\mathsf{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ for min-entropy k and error ε where $d = O\left(\frac{\log^2(n/\varepsilon)}{\log(k/m)}\right)$.*

4 Single-output oblivious local decision trees

In this section, we prove lower bounds for oblivious local decision trees with a single output ($m = 1$) using separator arguments. We show that any such tree computing a sumset disperser must have large depth, thereby establishing Theorem 1.

In Section 4.1, we will prove the following technical version of Theorem 1.

Theorem 4.1 (Technical version of Theorem 1). *Let $n, d, \ell, k \in \mathbb{N}$ be such that $d(1+\ell^2(k+1)/n) + k \leq n$. Let $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ be an (n,k) sumset disperser. Then any oblivious ℓ-local decision tree computing Disp requires depth at least $d+1$.*

Now we derive Theorem 1 by choosing appropriate parameters in Theorem 4.1 and instantiating it with the explicit sumset disperser construction from [Li23] (Theorem 3.20).

Theorem 1. *There exists a polynomial-time computable function $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ and a constant $C > 0$ such that any oblivious ℓ-local decision tree computing Disp has depth at least*

$$\frac{n^2}{n + C\ell^2 \log(n)} \ .$$

[9] It is not explicitly mentioned in [CL22] that the sumset extractor Ext is strong in the first source. However, this is immediate from inspecting their proof.

In particular, for $\ell = o(\sqrt{n/\log(n)})$, any oblivious ℓ-local decision tree computing Disp *has depth* $n - o(n)$.

Proof. Let $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ be the (n,k) sumset disperser from Theorem 3.20, where $k = C_0 \log n$. Let C be an arbitrary constant $C > C_0$. Note that for $\ell \geq n/\sqrt{C_0 \log(n)}$, the theorem statement holds trivially. Therefore, in the following we assume that $\ell < n/\sqrt{C_0 \log(n)}$. Let

$$d = \frac{n(n-k)}{n + (k+1) \cdot \ell^2} \,.$$

First we note that $d \geq 1$ for every $\ell < n/\sqrt{C_0 \log(n)}$ and all large enough n. Next we check that our choice of d satisfies the requirements of Theorem 4.1. Therefore, applying Theorem 4.1 with these parameter choices yields a lower bound of $d+1$ on the depth of any ℓ-local decision tree computing Disp.

To conclude the desired bound on the depth of ℓ-local decision trees, we note that for all $C > C_0$ and $k = C_0 \log(n)$, the following holds for all large enough values of n,

$$d = \frac{n(n-k)}{n + (k+1) \cdot \ell^2} > \frac{n^2}{n + C\ell^2 \log(n)} \,.$$

The second part of the theorem follows from the first one by observing that if $\ell = o(\sqrt{n/\log(n)})$, then $C\ell^2 \log(n) = o(n)$. □

4.1 Proof of Theorem 4.1

Our theorem follows from the following lemma, which identifies additive structure in arbitrary ℓ-local maps.

Lemma 4.2 (Sumsets in Local Maps)**.** *For all $d, \ell, \alpha \leq n$ and all ℓ-local maps $L : \{0,1\}^n \to \{0,1\}^d$, there exist sets $A, B \subseteq \{0,1\}^n$ such that for all $a \in A, b \in B$, we have that $L(a) = L(a+b)$. Furthermore, $|A| \geq 2^{\alpha}, |B| \geq 2^{n-d-\alpha d\ell^2/n}$.*

Proof. Let $L = (L_1, \ldots, L_d)$, where each $L_i\colon \{0,1\}^n \to \{0,1\}$ is an ℓ-local function. Consider an ℓ-uniform hypergraph H over vertex set $[n]$ with d edges defined as follows. For each $i \in [d]$, let $v_1, \ldots, v_\ell$ be the variables used in the function L_i (if L_i depends on $t < \ell$ variables, then we add arbitrary $\ell - t$ variables to the set). Then the hypergraph H contains the edge $(v_1, \ldots, v_\ell)$.

We claim there exists a set $I \subseteq [n]$ of size $|I| = \alpha$ such that $|\mathsf{Nbr}(I) \cup I| \leq \alpha d\ell^2/n$. For this, note that the expected number of edges a random vertex appears in is at most $d\ell/n$. Consider a random set of α vertices of the hypergraph. By linearity of expectation, the expected number of edges that these vertices appear in, is at most $\alpha d\ell/n$. This implies there exists a set I of size α such that the number of edges containing at least one vertex from I is at most $\alpha d\ell/n$. For this set I, it holds that $|\mathsf{Nbr}(I) \cup I| \leq \alpha d\ell^2/n$, as desired.

Let the separator set S be the set of all neighbors of I without the vertices from I, i.e., $S = \mathsf{Nbr}(I) \setminus I$. Let us denote the remaining vertices by $R = [n] \setminus (S \cup I)$. Let $\sigma = |S|$ and let $\rho = |R|$. We remark that $|R| \geq n - |\mathsf{Nbr}(I) \cup I| \geq n - \alpha d\ell^2/n$.

Note that by construction, no edge contains a vertex from I and a vertex from R simultaneously. Therefore, for every edge $e \in H$, either $e \subseteq I \cup S$ or $e \subseteq R \cup S$. Let $Q_1\colon \{0,1\}^{\alpha+\sigma} \to \{0,1\}^{d_1}$ be the set of functions in L that depend on the variables from $I \cup S$, and $Q_2\colon \{0,1\}^{\rho+\sigma} \to \{0,1\}^{d_2}$ be

the remaining functions in L (that depend on the variables from $R \cup S$) for some $d_1 + d_2 = d$. Now, up to a permutation of the coordinates, $L(x) = (Q_1(x_I, x_S), Q_2(x_R, x_S))$.

Using this decomposition of L, we will be able to define our desired sets A and B. To do that, we will fix the variables in the separator S to be 0^σ, and then carefully define our sets A, B. First, for each $z \in \{0,1\}^{d_2}$, define

$$Y(z) = \{y \in \{0,1\}^\rho : Q_2(y, 0^\sigma) = z\} .$$

Let $z^* = \operatorname{argmax}_z |Y(z)|$, the outcome with the largest preimage with respect to $Q_2(\cdot, 0^\sigma)$. Since z^* is the outcome with the largest preimage size, by an averaging argument, we obtain that $|Y(z^*)| \geq 2^{\rho - d_2} \geq 2^{\rho - d}$. Fix $y^* \in Y(z^*)$ to be arbitrary.

We are now ready to define our sets A, B. Let $A = A_S \times A_I \times A_R$ and $B = B_S \times B_I \times B_R$, where

$$\begin{aligned} A_S &= \{0^\sigma\} , & A_I &= \{0,1\}^\alpha , & A_R &= \{y^*\} ; \\ B_S &= \{0^\sigma\} , & B_I &= \{0^\alpha\} , & B_R &= Y(z^*) + y^* . \end{aligned}$$

We first prove lower bounds on the sizes of A and B, and later show that $L(a) = L(a+b)$ for all $a \in A$ and $b \in B$. By definitions of A, B, we see that $|A| = 2^\alpha$ and $|B| = |Y(z^*)| \geq 2^{\rho - d}$. From $\rho = |R| \geq n - |\mathsf{Nbr}(I) \cup I| \geq n - \alpha d \ell^2 / n$, we conclude that $|B| \geq 2^{n - d - \alpha d \ell^2 / n}$, as desired.

It remains to show that for all $a \in A, b \in B : L(a) = L(a+b)$. To do that, by the decomposition of L, it suffices to prove the corresponding claims for Q_1 and Q_2.

We first claim that for all $(a_I, a_S) \in A_I \times A_S$ and $(b_I, b_S) \in B_I \times B_S$, it holds that $Q_1(a_I, a_S) = Q_1(a_I + b_I, a_S + b_S)$. Indeed, since $A_S = B_S = \{0^\sigma\}$ and $B_I = \{0^\alpha\}$, we have that $a_I = a_I + b_I$, and $a_S = a_S + b_S$; and the claim follows.

Next, we claim that for all $(a_R, a_S) \in A_R \times A_S$ and $(b_R, b_S) \in B_R \times B_S$, it holds that $Q_2(a_R, a_S) = Q_2(a_R + b_R, a_S + b_S)$. Since $A_S = B_S = \{0^\sigma\}$ and $A_R = \{y^*\}$, it suffices to show that $Q_2(y^*, 0^\sigma) = Q_2(y^* + b_R, 0^\sigma)$. As $y^* \in Y(z^*)$, we have that $Q_2(y^*, 0^\sigma) = z^*$. Also, since $B_R = y^* + Y(z^*)$, $y^* + b_R \in Y(z^*)$ and hence, $Q_2(y^* + b_R, 0^\sigma) = z^* = Q_2(y^*, 0^\sigma)$ as desired.

With these two claims, we have that for all $a \in A, b \in B, L(a) = L(a+b)$ as desired. □

Equipped with Lemma 4.2, we are ready to prove Theorem 4.1.

Theorem 4.1 (Technical version of Theorem 1)**.** *Let $n, d, \ell, k \in \mathbb{N}$ be such that $d(1 + \ell^2(k+1)/n) + k \leq n$. Let $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ be an (n,k) sumset disperser. Then any oblivious ℓ-local decision tree computing Disp requires depth at least $d+1$.*

Proof. By way of contradiction, assume there exists an oblivious ℓ-local decision tree T of depth d computing Disp. Let T be such that $T(x) = h(L(x))$ where $L : \{0,1\}^n \to \{0,1\}^d$ is an ℓ-local map and $h : \{0,1\}^d \to \{0,1\}$.

We apply Lemma 4.2 to L with $\alpha = k+1$ to obtain $A, B \subseteq \{0,1\}^n$ such that $|A| \geq 2^\alpha, |B| \geq 2^{n - d - \alpha d \ell^2 / n}$ and for all $a \in A, b \in B$, it holds that $L(a) = L(a+b)$.

For $z \in \{0,1\}$, let $A_z = \{y \in A : h(L(y)) = z\}$. Without loss of generality, let $|A_0| \geq |A|/2$. Then, for all $a_0 \in A_0$ and $b \in B$, it holds that $L(a_0) = L(a_0 + b)$. Since $h(L(a_0)) = 0$, we infer that $h(L(a_0 + b)) = h(L(a_0)) = 0$. Thus, $T(A_0 + B) = h(L(A_0 + B)) = \{0\}$. However, as $|A_0| \geq 2^{\alpha - 1} \geq 2^k$ and $|B| \geq 2^{n - d - \alpha d \ell^2 / n} \geq 2^k$, we obtain a contradiction to the fact that T is a sumset disperser for min-entropy k. □

5 Multi-output oblivious local decision trees

In this section, we establish lower bounds for multi-output oblivious local decision trees by exploiting the fact that modifying a small number of input bits induces only a small change in the Hamming distance of the outputs of local maps. We use this observation to identify sumsets that any local map sends to a small number of outputs and, consequently, that are also mapped to a small number of outputs by any oblivious local decision tree. To prove Theorem 2, we then construct a function (specifically, a sumset condenser) for which such sumsets must be mapped to many distinct outputs.

In Section 6, we will construct the following condensers for sumset sources, which will serve as the hard function for our lower bound.

Theorem 5.1. *There exist universal constants $C' > C > 1, \gamma > 0$ with the following property. For every constant $\delta > 0$ and $n, m, k_1, k_2 \in \mathbb{N}$ such that $k_1 \geq k_2 \geq \log^C(n)$ and $m \leq (1-\delta)k_1$, there exists a polynomial-time computable function $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ that is a $(k_1, k_2, k_{\mathsf{out}})$ sumset condenser with error $n^{-\gamma}$, where $k_{\mathsf{out}} = m - \log^{C'}(n)$.*

In Section 5.1, we will prove the following technical version of Theorem 2.

Theorem 5.2 (Technical version of Theorem 2)**.** *Let n, m, ℓ, α, d be such that $2\ell d/n \leq \alpha \leq d$ and $m \leq n$. Let $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ be a $(k_1, k_2, k_{\mathsf{out}})$ sumset condenser with error $< \frac{1}{2}$, where*

$$k_1 = n - d,\ k_2 = (\alpha n/2d\ell)\log(2d\ell/\alpha),\ k_{\mathsf{out}} = \alpha(\log(d/\alpha) + 3)\ .$$

Then any oblivious ℓ-local decision tree computing Cond requires depth at least $d+1$.

We now prove Theorem 2 using the sumset condenser from Theorem 5.1 as the hard function in Theorem 5.2.

Theorem 2. *For every $\ell \leq n/\operatorname{poly}(\log(n))$, there exists $m = \ell \cdot \operatorname{poly}(\log(n))$ and a polynomial-time computable function $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ such that any oblivious ℓ-local decision tree computing Cond has depth $n - o(n)$.*

Proof. Let $C' > C > 1$ be the constants from Theorem 5.1. In the following, we will assume that $\ell \leq O(n/\log^{C'+3}(n))$. Let $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ be the $(k_1, k_2, k_{\mathsf{out}})$ sumset condenser from Theorem 5.1 for

$$k_1 = n/\log(\log(n)),\ k_2 = \log^C(n),\ k_{\mathsf{out}} = \ell \cdot \log^{C'+2}(n),\ m = 2k_{\mathsf{out}}\ .$$

First we check that the parameters satisfy the requirements of Theorem 5.1. Indeed, $k_1 \geq k_2 \geq \log^C(n)$, and $m \leq (1-\delta)k_1$ for every $\ell \leq O(n/\log^{C'+3}(n))$. Finally, $k_{\mathsf{out}} \leq m - \log^{C'}(n)$ holds from $k_{\mathsf{out}} \geq \log^{C'}(n)$.

We now apply Theorem 5.2 with $d = n - n/\log(\log(n))$ and $\alpha = \ell \cdot \log^C(n)$, and use Cond as the hard function. To see that we satisfy the requirements of Theorem 5.2, we note that $2\ell d/n \leq \alpha \leq d$ holds from $C' > C$, and $m \leq n$ for all $\ell \leq O(n/\log^{C'+3}(n))$. Moreover, $k_1 = n - d$, $k_2 < (\alpha n/2d\ell)\log(2d\ell/\alpha)$, and $k_{\mathsf{out}} > \alpha(\log(d/\alpha) + 3)$.

Therefore, the $(k_1, k_2, k_{\mathsf{out}})$ sumset condenser Cond requires oblivious ℓ-local decision trees of depth at least $d = n - n/\log\log n$. □

5.1 Proof of Theorem 5.2

In this subsection, we prove Theorem 5.2. To this end, we first prove the following lemma concerning the concentration of certain sumset sources when acted on by a local map.

Lemma 5.3 (Sumset concentration lemma)**.** *Let $n, d, \ell, \alpha \in \mathbb{N}$ be such that $2d\ell/n \leq \alpha \leq d$. Then, for any ℓ-local map $L : \{0,1\}^n \to \{0,1\}^d$, there exist sets $A, B \subseteq \{0,1\}^n$ with $|A| \geq 2^{n-d}, |B| \geq \binom{n}{\leq(\alpha n/2d\ell)}$ such that $|L(A+B)| \leq \binom{d}{\leq\alpha}$.*

Proof. By an averaging argument there exists $y_0 \in \{0,1\}^d$ such that $\left|L^{-1}(y_0)\right| \geq 2^{n-d}$. Let $A = L^{-1}(y_0)$. For $x \in \{0,1\}^n$, define $\mathsf{Nbr}(x) = \{i \in [d] : L_i \text{ depends on some } j \in [n] \text{ s.t. } x_j = 1\}$. Let $B = \{b \in \{0,1\}^n : |\mathsf{Nbr}(b)| \leq \alpha\}$.

We claim that $|B| \geq \binom{n}{\leq(\alpha n/2\ell d)}$. To this end, consider a bipartite graph with the left vertex set $[n]$, the right vertex set $[d]$, and an edge (j,i) in this bipartite graph if the ℓ-local function L_i depends on input j. Since L is an ℓ-local map, the total number of edges in this graph is at most $d\ell$. Hence, the average left degree in this graph is at most $d\ell/n$. For $w \in \mathbb{N}$, consider the quantity $\mathbb{E}_{x\in\{0,1\}^n,|x|\leq w}[|\mathsf{Nbr}(x)|]$, obtained by choosing a random subset of size at most w of the left vertices of G and computing the total number of neighbors on the right. Since the average left degree in G is at most $d\ell/n$, we infer that $\mathbb{E}_{x\in\{0,1\}^n,|x|\leq w}[|\mathsf{Nbr}(x)|] \leq wd\ell/n$. We now apply Markov's inequality to obtain that $\Pr_{x\in\{0,1\}^n,|x|\leq w}[|\mathsf{Nbr}(x)| \geq (4/3)wd\ell/n] \leq \frac{3}{4}$. Hence, $\Pr_{x\in\{0,1\}^n,|x|\leq w}[|\mathsf{Nbr}(x)| \leq (4/3)wd\ell/n] \geq \frac{1}{4}$. Setting $w = \frac{3\alpha n}{4d\ell}$, we conclude that $|B| \geq \frac{1}{4}\binom{n}{\leq(3\alpha n/4d\ell)} \geq \binom{n}{\leq(\alpha n/2d\ell)}$, as desired.

Lastly, we show that $|L(A+B)| \leq \binom{d}{\leq\alpha}$, proving our claim. Let $a \in A, b \in B$ be arbitrary. Using the definition of Nbr, we observe that $L(a+b)$ and $L(a)$ may only differ in output bits with indices in $\mathsf{Nbr}(b)$. Hence, the Hamming distance between $L(a+b)$ and $L(a) = y_0$ is at most $|\mathsf{Nbr}(b)|$. By our choice of B, we have that $|\mathsf{Nbr}(b)| \leq \alpha$. Since there are $\binom{d}{\leq\alpha}$ many strings at Hamming distance at most α from y_0, we conclude that $|L(A+B)| \leq \binom{d}{\leq\alpha}$, as desired. □

Equipped with Lemma 5.3, we are ready to prove Theorem 5.2.

Theorem 5.2 (Technical version of Theorem 2)**.** *Let n, m, ℓ, α, d be such that $2\ell d/n \leq \alpha \leq d$ and $m \leq n$. Let $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ be a $(k_1, k_2, k_{\mathsf{out}})$ sumset condenser with error $< \frac{1}{2}$, where*

$$k_1 = n - d,\ k_2 = (\alpha n/2d\ell)\log(2d\ell/\alpha),\ k_{\mathsf{out}} = \alpha(\log(d/\alpha) + 3)\ .$$

Then any oblivious ℓ-local decision tree computing Cond requires depth at least $d+1$.

Proof. Assume that there exists an oblivious ℓ-local decision tree T of depth d computing Cond. Let $T = h(L(\cdot))$ where $L : \{0,1\}^n \to \{0,1\}^d$ is an ℓ-local map and h is an arbitrary function.

We apply Lemma 5.3 to L to infer that there exist $A, B \subseteq \{0,1\}^n$ such that $|A| \geq 2^{n-d} = 2^{k_1}, |B| \geq \binom{n}{\leq\alpha n/2d\ell}$, and $|T(A+B)| \leq |L(A+B)| \leq 2^{\alpha(\log(d/\alpha)+2)}$. We note that $|B| \geq 2^{k_2}$ since

$$|B| \geq \binom{n}{\alpha n/2d\ell} \geq (2d\ell/\alpha)^{\alpha n/2d\ell} = 2^{k_2}\ .$$

This implies that the output of T on the sumset source $A+B$ is at statistical distance $\geq \frac{1}{2}$ from all distributions with min-entropy $\alpha(\log(d/\alpha)+2)+1 \leq \alpha(\log(d/\alpha)+3) = k_{\mathsf{out}}$. However, this contradicts the fact that T computes Cond. □

6 Construction of a sumset condenser

In this section, we prove Theorem 5.1, i.e., we construct a sumset condenser that serves as the hard function in Theorem 2.

Theorem 5.1. *There exist universal constants $C' > C > 1, \gamma > 0$ with the following property. For every constant $\delta > 0$ and $n, m, k_1, k_2 \in \mathbb{N}$ such that $k_1 \geq k_2 \geq \log^C(n)$ and $m \leq (1-\delta)k_1$, there exists a polynomial-time computable function $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ that is a $(k_1, k_2, k_{\mathsf{out}})$ sumset condenser with error $n^{-\gamma}$, where $k_{\mathsf{out}} = m - \log^{C'}(n)$.*

Proof. Let $C_{3.22}, \delta_{3.22}, \gamma_{3.22}$ be the universal constants from Theorem 3.22. We let our universal constant $C > 0$ be large enough constant such that $C \geq C_{3.22}$ and $C \geq 3/\delta_{3.22}$.

We let $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{m_0}$ be the sumset extractor from Theorem 3.22 for $(n, \log^C(n), \log^C(n))$ sumset sources, where $m_0 = (\log^C(n))^{\delta_{3.22}} \geq \log^3(n)$ by our choice of $C \geq 3/\delta_{3.22}$. Note that the choice $C \geq C_{3.22}$ guarantees that we satisfy the entropy requirement of Theorem 3.22.

We let $\mathsf{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be the linear seeded extractor from Theorem 3.25 for min-entropy k_1 with error $\varepsilon_0 = n^{-\gamma_{3.22}}$, where $d = O\left(\frac{\log^2(n/\varepsilon_0)}{\log(k_1/m)}\right) = O(\log^2(n))$. Note that our parameters $m \leq k_1 \leq n$ satisfy the requirements of Theorem 3.25.

We are now ready to present our construction of the condenser Cond. On input $x \in \{0,1\}^n$, we compute $y = \mathsf{Ext}(x) \in \{0,1\}^{m_0}$. We then take the prefix y_{pre} of y of length d, and output $\mathsf{LExt}(x, y_{\mathsf{pre}}) \in \{0,1\}^m$. We note that for all large enough n, the length of y is $m \geq \log^3(n) \geq d$ as $d = O(\log^2(n))$.

We now analyze our construction. Let $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$ be an arbitrary (n, k_1, k_2) sumset source. Without loss of generality we assume that $\mathbf{X}_2$ is an $(n, \log^C(n))$ flat source. Indeed, $\mathbf{X}_2$ is a convex combination of such flat sources (see, e.g., [Vad12, Lemma 6.10]). Then, $\mathbf{X}$ is a convex combination of sumset sources where the second source is a flat source with min-entropy exactly $\log^C(n)$. Now condensing from each such source, we obtain a condenser (with the exact same parameters) from the original source $\mathbf{X}$.

As LExt is a linear seeded extractor, our output distribution can be written as $\mathsf{LExt}(\mathbf{X}, \mathbf{Y}_{\mathsf{pre}}) = \mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}}) + \mathsf{LExt}(\mathbf{X}_2, \mathbf{Y}_{\mathsf{pre}})$. We first argue that $\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}})$ is close to uniform. As Ext is a strong two source extractor, we are guaranteed that $(\mathbf{X}_1, \mathbf{Y}) \approx_{\varepsilon_0} (\mathbf{X}_1, \mathbf{U}_{m_0})$ where $\varepsilon_0 = n^{-\gamma_{3.22}}$. Since $\mathbf{Y}_{\mathsf{pre}}$ is a prefix of $\mathbf{Y}$, using the data processing inequality (Lemma 3.7), we obtain that $(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}}) \approx_{\varepsilon_0} (\mathbf{X}_1, \mathbf{U}_d)$. From this, using the triangle inequality and the data processing inequality again (Lemma 3.7), we conclude that $\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}}) \approx_{2\varepsilon_0} \mathbf{U}_m$ (because $\mathbf{X}_1$ has min-entropy k_1 required by LExt).

We now show that adding $\mathsf{LExt}(\mathbf{X}_2, \mathbf{Y}_{\mathsf{pre}})$ to the nearly uniform distribution $\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}})$ does not make it lose much of entropy. Towards this, we first observe that as $\mathbf{X}_2$ is a flat source with min-entropy exactly $\log^C(n)$, we have that $|\mathrm{Supp}(\mathbf{X}_2)| = 2^{\log^C(n)}$. Using the fact that the domain of $\mathbf{Y}_{\mathsf{pre}}$ is $\{0,1\}^d$, we infer that $|\mathrm{Supp}(\mathsf{LExt}(\mathbf{X}_2, \mathbf{Y}_{\mathsf{pre}}))| \leq 2^{d+\log^C(n)}$.

We apply the chain rule for smooth min-entropy (Lemma 3.11) to random variables $\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}})$ and $\mathsf{LExt}(\mathbf{X}_2, \mathbf{Y}_{\mathsf{pre}})$ with parameters $\varepsilon_{3.11} = 2\varepsilon_0$, $\delta_{3.11} = \varepsilon_0$, and $\gamma_{3.11} = \sqrt{2\varepsilon_0}$ to infer that for at least $1 - 6\sqrt{\varepsilon_0}$ fraction of fixings $z_2 \sim \mathsf{LExt}(\mathbf{X}_2, \mathbf{Y}_{\mathsf{pre}})$, it holds that $\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}})|(\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}}) = z_2)$ has statistical distance at most $\sqrt{2\varepsilon_0}$ from a distribution with min-entropy at least $m - d - \log^C(n) - \log(1/\varepsilon_0) = m - d - \log^C(n) - \gamma_{3.22}\log(n)$. Conditioned on any such fixing, our output distribution becomes $\mathsf{LExt}(\mathbf{X}, \mathbf{Y}_{\mathsf{pre}}) = \mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}}) + z_2$

and so it also has statistical distance at most $\sqrt{2\varepsilon_0}$ from a distribution with min-entropy at least $m - d - \log^C(n) - \gamma_{3.22}\log(n)$.

As $d = O(\log^2(n))$ and $\gamma_{3.22}$ is a universal constant, we let C' be a large enough universal constant and γ to be a small enough universal constant so that for at least $1 - n^{-\gamma}/2$ fraction of fixings $z_2 \sim \mathsf{LExt}(\mathbf{X}_2, \mathbf{Y}_{\mathsf{pre}})$, it holds that $\mathsf{LExt}(\mathbf{X}, \mathbf{Y}_{\mathsf{pre}})|(\mathsf{LExt}(\mathbf{X}_1, \mathbf{Y}_{\mathsf{pre}}) = z_2)$ has statistical distance at most $n^{-\gamma}/2$ from a distribution with min-entropy at least $m - \log^{C'}(n)$. From this we conclude that $\mathsf{LExt}(\mathbf{X}, \mathbf{Y}_{\mathsf{pre}})$ has statistical distance at most $n^{-\gamma}$ from a distribution with min-entropy at least $m - \log^{C'}(n)$, as desired. □

7 Oblivious degree-2 decision trees

In this section, we prove maximal lower bounds against oblivious degree-2 decision trees, establishing Theorem 3.

Lemma 7.1. *Let* $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ *be an* (n,k) *directional affine disperser. Then any oblivious degree-2 decision tree computing* Disp *requires depth at least* $n - k + 1$.

Proof. Assume that there exists such a tree $T : \{0,1\}^n \to \{0,1\}$ of depth $d \leq n-k$ computing Disp. Let $T = h(Q(\cdot))$ where $Q\colon \{0,1\}^n \to \{0,1\}^d$ is a degree-2 map and $h : \{0,1\}^d \to \{0,1\}$ is an arbitrary function. As $d < n$, there exist $u, v \in \{0,1\}^n$ such that $u \neq v$ and $Q(u) = Q(v)$. Let $a = u + v$. Since $u \neq v$, we have that $a \neq 0$. Let $L\colon \{0,1\}^n \to \{0,1\}^d$ be defined as

$$L(x) = Q(x) + Q(x + a)\ .$$

First we note that $L(u) = Q(u) + Q(v) = 0$. As Q is a degree 2 map, L is a degree 1 map, i.e., an affine map. Since 0 lies in the image of the affine map L, there exists an affine subspace $U \subseteq \{0,1\}^n$ of dimension at least $n - d$ such that $L(U) = 0$. Hence, for all $x \in U$, it holds that $Q(x) + Q(x + a) = 0$, implying $Q(x) = Q(x + a)$. Therefore, for all $x \in U$, we obtain that $T(x) = h(Q(x)) = h(Q(x + a)) = T(x + a)$. However, this implies that for all $x \in U$, $T(x) + T(x + a) = 0$. As U is a subspace of dimension at least $n - d \geq n - (n - k) = k$, and $a \neq 0$, we get a contradiction. □

We instantiate Lemma 7.1 with the construction of directional affine dispersers from [LZ24b] (Theorem 3.17) to prove Theorem 3.

Theorem 3. *There exists a polynomial-time computable function* $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ *such that any oblivious degree-2 decision tree computing* Disp *has depth at least* $n - O(\log(n))$.

Proof. We let $\mathsf{Disp}\colon \{0,1\}^n \to \{0,1\}$ be the (n,k) directional affine disperser from Theorem 3.17 for $k = O(\log(n))$, and apply Lemma 7.1 to it. □

8 Conclusion and Open problems

We conclude by providing open problems for various models that we studied.

Oblivious decision trees.

- Find an explicit function requiring depth overhead of $\Omega(n)$ for oblivious local decision trees. Obtaining such a result for single-output functions would yield an improved lower bound for oblivious bounded-width branching programs.
- Prove strong lower bounds (larger than $0.01n$, obtained from variety dispersers of [LZ19]) for oblivious degree-d decision trees for $d \geq 3$. It is unclear how to directly extend our lower bound for $d = 2$ against directional affine dispersers to obtain even a non-trivial lower bound for $d = 3$.

Non-oblivious decision trees. Perhaps the main direction for future work is to extend our lower bounds to *non-oblivious* local decision trees, even with substantially weaker parameters.

- The most pressing open problem is to construct an explicit function which requires depth overhead of $\omega(1)$ for non-oblivious local decision trees. We hope our techniques regarding finding sumset structure in local computation can be generalized to obtain such a bound.
- Extend the lower bound of Theorem 1 or Theorem 2 (with $m = o(n)$ outputs) to the non-oblivious case. Even for constant $\ell = 16$, a size lower bound of $2^{n-o(n)}$ would give us a lower bound of $3.9n - o(n)$ against unbounded-depth circuits—the first major improvement in decades. As a stepping stone towards this, it is necessary to obtain a depth lower bound of $n - o(n)$ for $\ell = 16$.
- Extend the lower bound of Theorem 1 or Theorem 2 (with $m = o(d)$ outputs) to the non-oblivious setting. Even in the much weaker regime of $\ell = \omega(1)$ and $d = \Omega(n)$, this would imply a super-linear lower bound for series-parallel circuits. Extending this further to $\ell = n^{0.01}$ and $d = \omega(n/\log(\log(n)))$ would imply a super-linear lower bound for log-depth circuits. Either result would resolve a problem open for nearly 50 years [Val77].
- Extend the lower bound of Theorem 3 to the non-oblivious case. Dispersers for degree-2 variety sources with $o(n)$ dependence on min-entropy would imply a lower bound of $3.11n$ on the size of unbounded-depth circuits, improving the best known bound [LY22]. Such dispersers will also require depth $n - o(n)$ when computed by degree-2 decision trees. Hence, we view resolving this as a necessary step towards obtaining the improved circuit lower bound. Our maximal lower bounds for oblivious degree-2 decision trees resolve this problem in the oblivious setting.
- Show that any ℓ-local decision tree computing Majority requires depth at least $(n/\ell) \cdot \log(\ell)$. It has been conjectured that Σ_3^k circuits computing Majority (for $k \leq \sqrt{n \log(n)}$) require size at least $2^{(n/k)\log(k)}$ [HJP95]. Such a lower bound would yield the above depth lower bound for local decision trees. Hence, we view resolving this question as an important stepping stone towards resolving that conjecture. We note that Lecomte et al. showed that any *oblivious* ℓ-local decision tree computing Majority requires depth at least $(n/\ell) \cdot \log(\ell)$, resolving the question in the oblivious setting [LRT22].
- More broadly, it would be interesting to discover more structural properties of local decision trees and find connections to other areas. For instance, the Fourier-analytic structure

of degree-1 decision trees (parity decision trees) has found many interesting connections to learning theory, communication complexity and the log-rank conjecture (see, e.g., [KM93; MS24; GTW21]).

Acknowledgements

We are thankful to Eshan Chattopadhyay for many fruitful discussions on this topic.

AI Usage Statement

The article was written entirely by the authors, and all underlying ideas, arguments, and proofs are solely the authors' own. ChatGPT Pro was used exclusively for proofreading the final version of the paper.

References

[AM86] Noga Alon and Wolfgang Maass. "Meanders, Ramsey Theory and Lower Bounds for Branching Programs". In: *FOCS*. 1986 (cit. on p. 2).

[ACGLR22] Omar Alrabiah, Eshan Chattopadhyay, Jesse Goodman, Xin Li, and João Ribeiro. "Low-Degree Polynomials Extract From Local Sources". In: *ICALP*. 2022 (cit. on p. 2).

[BPRS90] László Babai, Pavel Pudlák, Vojtech Rödl, and Endre Szemerédi. "Lower Bounds to the Complexity of Symmetric Boolean Functions". In: *Theoretical Computer Science* (1990) (cit. on p. 2).

[BP26] Kirill Bessonov and Vladimir Podolskii. *Personal communication*. 2026 (cit. on p. 2).

[CL22] Eshan Chattopadhyay and Jyun-Jie Liao. "Extractors for sum of two sources". In: *STOC*. 2022 (cit. on pp. 4, 7, 15).

[CL23] Eshan Chattopadhyay and Jyun-Jie Liao. "Hardness Against Linear Branching Programs and More". In: *CCC*. 2023 (cit. on p. 3).

[CP22] Anastasiya Chistopolskaya and Vladimir V. Podolskii. "On the Decision Tree Complexity of Threshold Functions". In: *Theory of Computing Systems* (2022) (cit. on pp. 1, 2, 8).

[DW12] Anindya De and Thomas Watson. "Extractors and lower bounds for locally samplable sources". In: *ACM Transactions on Computation Theory* 4.1 (2012), pp. 1–21 (cit. on p. 2).

[DK11] Evgeny Demenkov and Alexander S. Kulikov. "An elementary proof of a $3n - o(n)$ lower bound on the circuit complexity of affine dispersers". In: *MFCS*. 2011 (cit. on p. 3).

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. "A better-than-3n lower bound for the circuit complexity of an explicit function". In: *FOCS*. 2016 (cit. on p. 3).

[GTW21] Uma Girish, Avishay Tal, and Kewen Wu. "Fourier Growth of Parity Decision Trees". In: *CCC*. 2021 (cit. on p. 23).

[GK16] Alexander Golovnev and Alexander S. Kulikov. "Weighted gate elimination: Boolean dispersers for quadratic varieties imply improved circuit lower bounds". In: *ITCS*. 2016 (cit. on pp. 9, 13).

[GKW21] Alexander Golovnev, Alexander S. Kulikov, and R. Ryan Williams. "Circuit Depth Reductions". In: *ITCS*. 2021 (cit. on p. 9).

[GPT22] Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. "Linear Branching Programs and Directional Affine Extractors". In: *CCC*. 2022 (cit. on p. 3).

[HJP95] Johan Håstad, Stasys Jukna, and Pavel Pudlák. "Top-Down Lower Bounds for Depth-Three Circuits". In: *Computational Complexity* (1995) (cit. on pp. 8, 22).

[HR15] Pavel Hrubes and Anup Rao. "Circuits with Medium Fan-In". In: *CCC*. 2015 (cit. on pp. 2, 9).

[KM93] Eyal Kushilevitz and Yishay Mansour. "Learning Decision Trees Using the Fourier Spectrum". In: *SIAM Journal of Computing* (1993) (cit. on p. 23).

[LRT22] Victor Lecomte, Prasanna Ramakrishnan, and Li-Yang Tan. "The Composition Complexity of Majority". In: *CCC*. 2022 (cit. on pp. 2, 9, 22).

[LZ19] Fu Li and David Zuckerman. "Improved Extractors for Recognizable and Algebraic Sources". In: *APPROX/RANDOM*. 2019 (cit. on pp. 13, 22).

[LY22] Jiatu Li and Tianqi Yang. "$3.1n - o(n)$ circuit lower bounds for explicit functions". In: *STOC*. 2022 (cit. on pp. 3, 13, 22).

[Li16] Xin Li. "Improved Two-Source Extractors, and Affine Extractors for Polylogarithmic Entropy". In: *FOCS*. 2016 (cit. on p. 7).

[Li23] Xin Li. "Two Source Extractors for Asymptotically Optimal Entropy, and (Many) More". In: *FOCS*. 2023 (cit. on pp. 3, 4, 14, 15).

[LZ24a] Xin Li and Yan Zhong. "Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs". In: *CCC*. 2024 (cit. on p. 3).

[LZ24b] Xin Li and Yan Zhong. "Two-Source and Affine Non-Malleable Extractors for Small Entropy". In: *ICALP*. 2024 (cit. on pp. 3, 4, 14, 21).

[MS24] Nikhil S. Mande and Swagato Sanyal. "On parity decision trees for Fourier-sparse Boolean functions". In: *ACM Transactions on Computation Theory* (2024) (cit. on p. 23).

[MW97] Ueli Maurer and Stefan Wolf. "Privacy Amplification Secure Against Active Adversaries". In: *CRYPTO*. 1997 (cit. on p. 12).

[Nec66] Eduard Ivanovich Nechiporuk. "On a Boolean function". In: *Doklady Akademii Nauk*. 1966 (cit. on p. 2).

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. "An improved exponential-time algorithm for k-SAT". In: *Journal of the ACM* (2005) (cit. on p. 2).

[PPZ99] Ramamohan Paturi, Pavel Pudlák, and Francis Zane. "Satisfiability Coding Lemma". In: *Chicago Journal of Theoretical Computer Science* (1999) (cit. on p. 8).

[PSZ00] Ramamohan Paturi, Michael E. Saks, and Francis Zane. "Exponential lower bounds for depth three Boolean circuits". In: *Computational Complexity* (2000) (cit. on p. 2).

[RRV02] Ran Raz, Omer Reingold, and Salil Vadhan. "Extracting all the Randomness and Reducing the Error in Trevisan's Extractors". In: *Journal of Computer and System Sciences* 65.1 (2002), pp. 97–128 (cit. on p. 15).

[Tre01] Luca Trevisan. "Extractors and pseudorandom generators". In: *Journal of the ACM* 48.4 (2001), pp. 860–879 (cit. on p. 15).

[Vad12] Salil P. Vadhan. "Pseudorandomness". In: *Foundations and Trends in Theoretical Computer Science* 7.1–3 (2012), pp. 1–336 (cit. on p. 20).

[Val77] Leslie G. Valiant. "Graph-Theoretic Arguments in Low-Level Complexity". In: *MFCS*. 1977 (cit. on pp. 8, 11, 22).

[Vio14] Emanuele Viola. "Extractors for circuit sources". In: *SIAM Journal on Computing* 43.2 (2014), pp. 655–672 (cit. on p. 2).

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il