Decision List Compression by Mild Random Restrictions

SHACHAR LOVETT, University of California, San Diego
KEWEN WU, University of California, Berkeley
JIAPENG ZHANG, University of Southern California

A decision list is an ordered list of rules. Each rule is specified by a term, which is a conjunction of literals, and a value. Given an input, the output of a decision list is the value corresponding to the first rule whose term is satisfied by the input. Decision lists generalize both CNFs and DNFs and have been studied both in complexity theory and in learning theory.

The size of a decision list is the number of rules, and its width is the maximal number of variables in a term. We prove that decision lists of small width can always be approximated by decision lists of small size, where we obtain sharp bounds for such approximation. This also resolves a conjecture of Gopalan, Meka, and Reingold (Computational Complexity, 2013) on DNF sparsification.

An ingredient in our proof is a new random restriction lemma, which allows to analyze how DNFs (and more generally, decision lists) simplify if a small fraction of the variables are fixed. This is in contrast to the more commonly used switching lemma, which requires most of the variables to be fixed.

CCS Concepts: • **Theory of computation** → *Pseudorandomness and derandomization*; *Complexity theory and logic*; • **Mathematics of computing** → *Combinatorics;*

Additional Key Words and Phrases: Decision lists, DNF sparsification, switching lemma

ACM Reference format:
Shachar Lovett, Kewen Wu, and Jiapeng Zhang. 2021. Decision List Compression by Mild Random Restrictions. *J. ACM* 68, 6, Article 45 (October 2021), 17 pages.
https://doi.org/10.1145/3485007

1 INTRODUCTION

Decision lists are a model to represent Boolean functions, first introduced by Rivest [34]. A decision list is given by a list of rules $(C_1, v_1), \ldots, (C_m, v_m)$, where $C_m \equiv \texttt{True}$ is the final default rule. A rule is composed of a condition, given by a term C_i, which is a conjunction of literals (variables or their negations), and an output value v_i in some set V. A decision list computes a function $f : \{0,1\}^n \to V$ as follows:

Shachar Lovett research supported by NSF award 1614023.
Authors' addresses: S. Lovett, University of California, San Diego, Computer Science & Engineering Department, 9500 Gilman Drive, Mail code 0404, La Jolla, CA 92093, USA; email: shachar.lovett@gmail.com; K. Wu, University of California, Berkeley, Department of Electrical Engineering and Computer Sciences, 387 Soda Hall, Mail code 1776, Berkeley, CA 94720, USA; email: shlw_kevin@hotmail.com; J. Zhang, University of Southern California, Los Angeles, Computer Science & Engineering Department, 941 Bloom Walk, Mail code SAL 104, Los Angeles, CA 90089, USA; email: jiapengz@usc.edu.

0004-5411/2021/10-ART45 $15.00
https://doi.org/10.1145/3485007

If $C_1(x) =$ **True then** output v_1,
else if $C_2(x) =$ **True then** output v_2,
$\ldots$,
else if $C_{m-1}(x) =$ **True then** output v_{m-1},
else output v_m.

Decision lists generalize both CNFs and DNFs. For example, a DNF is a decision list with $v_1 = \cdots = v_{m-1} = 1$ and $v_m = 0$, and a CNF is a decision list with $v_1 = \cdots = v_{m-1} = 0$ and $v_m = 1$. It can be shown that decision lists are a strict generalization of both DNFs and CNFs [25, 34]. Following Rivest's original work, decision lists have been studied both in complexity theory [3, 6, 8, 10, 17, 26, 38] and in learning theory [4, 9, 18, 21, 30, 40, 41].

Complexity measures of decision lists. There are two natural complexity measures of decision lists: *size* and *width*. Let $L = ((C_i, v_i))_{i\in[m]}$ be a decision list. Its *size* is the number of rules in it (namely m), and its *width* is the maximal number of variables in a term C_i.

Decision list approximation. A decision list L ε-approximates another decision list L' if the two agree on a $(1-\varepsilon)$ fraction of the inputs.[1] It is straightforward to see that small-size decision lists can be approximated by small-width decision lists, by removing rules of large width. Concretely, a decision list of size m can be ε-approximated by a decision list of width $w = \log(m/\varepsilon)$, simply by removing all rules with terms of width more than w. The reverse direction is the main focus of this work. We prove the following result, which provides sharp bounds on approximating small-width decision lists by small-size decision lists.

Theorem 1.1 (Main Result). *Let $w \geq 1, \varepsilon > 0$. Any width-$w$ decision list L can be ε-approximated by a decision list L' of width w and size $s = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$. Moreover, L' is a sub-decision list of L, obtained by keeping s rules in L and removing the rest. The bound on s is optimal up to the constant in the $O(w)$ term.*

The proof of Theorem 1.1 appears in Section 2. We note that the size bound can be simplified, depending on whether the required error ε is below or above 2^{-w}:

$$\left(2 + \frac{1}{w}\log\frac{1}{\varepsilon}\right)^{O(w)} = \begin{cases} 2^{O(w)} & \varepsilon \geq 2^{-w}, \\ \left(\frac{2}{w}\log\frac{1}{\varepsilon}\right)^{O(w)} & \varepsilon \leq 2^{-w}. \end{cases}$$

In both cases, the bound we obtain is sharp up to the constant in the $O(w)$ term. We give examples demonstrating this in Section 3. We provide applications of our result in Section 4, which include *DNF sparsification*, *junta theorem*, and *learning small-width DNFs*.

1.1 Random Restrictions

Random restrictions are an essential ingredient of the proof of Theorem 1.1. Håstad's switching lemma [5, 19, 32] is based on the fact that small-width DNFs simplify under random restrictions. More concretely, a random restriction that fixes a $1 - O(1/w)$ fraction of the inputs simplifies a width-w DNF to a small-depth decision tree. The idea of obtaining structural information about a function from information about the family of random restrictions of the function is a general principle that has been used in many other works (e.g., References [27, 28, 36, 37]). In this work, we study random restrictions where a small constant fraction of the variables is fixed.

A good example to keep in mind is the TRIBES function: a read-once DNF[2] with 2^w terms of width w on disjoint variables. The TRIBES function does not simplify significantly under a random

[1] To clarify, approximation is under the uniform distribution over inputs.
[2] A read-once DNF is a DNF where no variable appears more than once.

restriction, unless one fixes a $1 - O(1/w)$ fraction of the inputs. For example, if we randomly fix 50% of the inputs, then the TRIBES function simplifies with high probability to what is essentially a smaller TRIBES function (formally, a read-once DNF of width $\Omega(w)$). However, we show that this is in essence the worst possible example.

The following lemma is a special case of Lemma 2.12 applied to DNFs (the full lemma deals with decision lists). Given a DNF $f\colon \{0,1\}^n \to \{0,1\}$, let $\rho \in \{0,1,*\}^n$ be a restriction, and let $f \restriction_\rho$ be the restricted DNF. Clearly, some terms in f might become redundant in $f \restriction_\rho$. For example, they could be false, or they could be implied by other terms. A term that is not redundant is called *useful*. We show that after fixing even a small fraction of the variables (say, 1%), a width-w DNF simplifies to have at most $2^{O(w)}$ useful terms and hence cannot be "too complicated."

Lemma 1.2 (DNFs Simplify After Mild Random Restrictions). *Let f be a width-w DNF, and let $f \restriction_\rho$ be a restriction of f obtained by restricting each variable with probability β, where the restricted variables take values 0 and 1 with equal probability. Then the expected number of useful terms in $f \restriction_\rho$ is at most $(4/\beta)^w$.*

1.2 Proof Overview

We give a high-level overview of the proof of Theorem 1.1. Let $L = ((C_i, v_i))_{i\in[m]}$ be a decision list of width w and size m.

General Framework. Given a subset $J \subset [m]$, we denote by $L|_J$ the decision list restricted to the rules in J, where we delete the rest. Our goal is to find a small subset $J \subset [m]$ such that $L|_J$ approximates L. We say that a rule (C_i, v_i) of L is *hit* by an input x if $C_i(x) = 1$ and $C_j(x) = 0$ for $j < i$; in this case, $L(x) = v_i$. The main intuition underlying our approach is as follows:

If a rule is rarely hit by random inputs, then we can safely remove it.

Armed with this intuition, our approach is to choose J to be the set of rules with the highest probability of being hit. We show that to get an ε-approximation, it suffices to keep $(2+\frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$ rules that are most likely to be hit.

Our general approach follows that of Lovett and Zhang [28]. They combined two key results in the analysis of Boolean functions: *random restrictions* and *noise stability*. The main innovation in the current work is that we apply random restrictions that fix only a small fraction of the inputs; this is in contrast to the common use of random restrictions, such as in the proof of Håstad's switching lemma [19], where most variables are fixed. The ability to handle random restrictions that fix only a small fraction is what allows us to obtain improved bounds.

Mild random restrictions. An index $i \in [m]$ is said to be *useful* if there exists an assignment x such that the evaluation of $L(x)$ hits the ith rule (and hence outputs v_i). We denote the number of useful indices in L by $\#\mathrm{useful}(L)$. This notion is natural, as we can always discard rules if no assignment hits them. The main point is that restrictions can render some rules in a decision list useless. Let ρ be a random restriction that keeps each variable alive[3] with probability α. We show that on average, the restricted decision list $L \restriction_\rho$ has a small number of useful indices:

$$\mathop{\mathbb{E}}_{\rho}\left[\#\mathrm{useful}\left(L \restriction_\rho\right)\right] \le \left(\frac{4}{1-\alpha}\right)^w.$$

The proof is based on an encoding argument. Let ρ be a restriction for which $L \restriction_\rho$ has T useful indices. Let $t \in [T]$ be uniformly chosen. We construct a new restriction ρ' by further restricting

[3]We say a variable is alive if it is not fixed to 0 or 1 by the random restriction.

the variables in the tth useful rule so that this rule is satisfied. We show that given ρ' and some small additional information a, we can recover both ρ and t. As ρ' is obtained by fixing a few variables in ρ, the number of options for ρ' is not much larger compared to the number of options for ρ. Together, this shows that typically T cannot be too large.

Noise Stability. Since there is no guarantee about the value on each rule of the decision list, it is convenient to consider the following *index function*. Let $L = ((C_i, v_i))_{i\in[m]}$ be a decision list. The index function of L outputs for an input x the index i of the first term in L satisfied by x. Equivalently, $\mathsf{Ind}L$ is given by the decision list $\mathsf{Ind}L = ((C_i, i))_{i\in[m]}$.

We make two important definitions. What we *want* to analyze are the quantities

$$p_L(i) := \Pr_x[\mathsf{Ind}L(x) = i],$$

where x is taken from the uniform distribution of the input. In particular, we want to show that there is a small set of indices J such that $\sum_{i\in J} p_L(i) \geq 1 - \varepsilon$. What we *can* analyze using random restrictions are the quantities

$$q_L(\alpha, i) = \Pr_\rho\left[\text{index } i \text{ is useful in } L\restriction_\rho\right],$$

since it holds that

$$\sum_i q_L(\alpha, i) = \mathop{\mathbb{E}}_\rho\left[\#\text{useful}\left(L\restriction_\rho\right)\right] \leq \left(\frac{4}{1-\alpha}\right)^w.$$

We use noise stability to connect the two.[4]

Let $\beta = 1 - \alpha$. For any $x \in \{0,1\}^n$, sample a random y from the noisy distribution $\mathcal{N}_\beta(x)$ by taking $\Pr[y_i = x_i] = \frac{1+\beta}{2}$ independently for $i \in [n]$. Consider sampling $x \in \{0,1\}^n$ uniformly and $y \sim \mathcal{N}_\beta(x)$. We can equivalently sample the pair (x, y) by first sampling a common restriction ρ, where each variables stays alive with probability α, and then sampling its completion for x and y independently. Let

$$\mathsf{Stab}_L(\beta, i) := \Pr_{x,y}[\mathsf{Ind}L(x) = \mathsf{Ind}L(y) = i].$$

We show that $p_L(i)$ is upper bounded by a polynomial in $q_L(\alpha, i)$, by relating them to $\mathsf{Stab}_L(\beta, i)$:

$$\frac{p_L(i)^2}{q_L(1-\beta, i)} \leq \mathsf{Stab}_L(\beta, i) \leq p_L(i)^{\frac{2}{1+\beta}}.$$

The upper bound is proven by hypercontrativity, and the lower bound by a somewhat delicate conditioning and Jensen's inequality. This allows us to obtain that

$$p_L(i) \leq q_L(1-\beta, i)^{\frac{1+\beta}{2\beta}}.$$

Finally, we put everything together by optimizing the value of β.

Related works. We already discussed the works of Gopalan, Meka, and Reingold [15] and Lovett and Zhang [28], which gave weaker bounds for DNF sparsification than those in Theorem 1.1.

There have been previous works studying how small-width DNFs simplify under mild random restrictions that fix a small constant fraction of the variables. The terminology of mild random restriction originates from Reference [16]. Segerlind, Buss, and Impagliazzo's work [35], improved by Razborov [33], show that width-w DNFs simplify to a decision tree of depth $2^{O(w)}$. We obtain bounds on size (namely, number of useful terms) in Theorem 1.1, which are better than bounds on depth. However, we only bound the first moment (that is, expected number of useful terms), while Reference [33] bounds higher moments as well. So to some extent, the results are incomparable. We

[4]Noise stability is well studied for general Boolean functions. See Reference [31] for details.

believe that with some further work, one can improve our techniques to obtain bounds on higher moments as well (this was unnecessary for the current work). Finally, it is also worthwhile to mention the work by the authors and Alweiss [1], where mild random restrictions (of a somewhat different flavor) were used to obtain improved bounds for the sunflower lemma in combinatorics.

Paper Organization. In Section 2, we prove the upper bound on decision list compression. In Section 3, we give the lower bounds to show the tightness of our result. In Section 4, we provide applications of our main results in detail. In Section 5, we summarize the article and present several future directions.

2 UPPER BOUNDS

We start by make some definitions formal. We denote $[n] = \{1, 2, \ldots, n\}$, variables are $x_1, \ldots, x_n$, and literals are $x_1, \neg x_1, \ldots, x_n, \neg x_n$. A *term* is a conjunction of literals.

Definition 2.1 (Decision List). A width-w size-m decision list is a list $L = ((C_i, v_i))_{i\in[m]}$ of rules. A rule is a pair (C_i, v_i), where C_i is a term containing at most w literals and each v_i is a value in some finite set V. We assume $C_m \equiv \mathsf{True}$, and (C_m, v_m) is the final default rule.

For any $J \subseteq [m]$ with $m \in J$, we denote by $L|_J = ((C_j, v_j))_{j\in J}$ the restriction of L to the rules in J, where elements of J are taken in ascending order.

The evaluation of L given assignment x is to find the first index i such that $C_i(x) = 1$ and then to output $L(x) = v_i$. We make additional remarks for the decision list to avoid potential pitfalls.

- If $m \notin J$, then we will consider $L|_J$ invalid, as it does not have a default rule at the end.
- No variable appears in any single term more than once, which rules out $x_1 \wedge x_1$ and $x_1 \wedge \neg x_1$.

Our goal in this section is to prove the following theorem, which is the upper bound part in Theorem 1.1.

Theorem 2.2. *Let $L = ((C_i, v_i))_{i\in[m]}$ be a width-w decision list. Then for every $\varepsilon > 0$, there exists $J \subseteq [m], m \in J$ of size $|J| = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$ such that $\Pr_x\left[L(x) \neq L|_J(x)\right] \leq \varepsilon$.*

To clarify, the probability is over a uniformly chosen input $x \in \{0, 1\}^n$.

2.1 Useful Indices

Since there is no guarantee about the value on each rule of the decision list, it is convenient to consider the index function. Let $L = ((C_i, v_i))_{i\in[m]}$ be a decision list on n variables. The index function of L is a function $\mathsf{Ind}L\colon \{0, 1\}^n \to [m]$, given by

$$\mathsf{Ind}L(x) = \min\{i \in [m] \mid C_i(x) = 1\}.$$

Equivalently, $\mathsf{Ind}L$ is given by the decision list $\mathsf{Ind}L = ((C_i, i))_{i\in[m]}$. Using the index function, it suffices to discard some rules of L and show it still approximates the index function.

Claim 2.3. *Let $L = ((C_i, v_i))_{i\in[m]}$ be a decision list. Then for any $J \subseteq [m], m \in J$, we have*

$$\Pr\left[L(x) \neq L|_J(x)\right] \leq \Pr\left[\mathsf{Ind}L(x) \notin J\right].$$

Proof. This follows as if $\mathsf{Ind}L(x) = j \in J$, then $L(x) = L|_J(x) = v_j$. □

Obviously, if a rule of a decision list is covered by some previous rules, then we can safely remove it. For example, in $(x_1, 1), (x_1 \wedge x_2, 2)$ the second rule is useless. To make this more formal, we introduce the following notion of a *useful index*.

Definition 2.4 (Useful Index). Given size-m decision list L, an index $i \in [m]$ is said to be *useful* if there exists an assignment x such that $\mathsf{Ind}L(x) = i$. We denote by $\#\mathsf{useful}\,(L)$ the number of useful indices in L.

Example 2.5. Assume $L = ((x_1, a), (x_1 \wedge \neg x_2, b), (1, c), (x_1, d), (1, e))$. Then indices $1, 3$ are useful, but indices $2, 4, 5$ are not. So $\#\mathsf{useful}\,(L) = 2$.

The main intuition underlying our approach is that rules that are hardly hit by random inputs can be removed. Motivated by this, we define the *hit probability* of the ith term as

$$p_L(i) := \Pr\left[\mathsf{Ind}L(x) = i\right].$$

Claim 2.6. *For any size-m decision list L, we have $\sum_{i=1}^{m} p_L(i) = 1$.*

Proof. This follows as the events $[\mathsf{Ind}L(x) = i]$ are a partition of the probability space. □

The following is our main technical lemma.

Lemma 2.7. *Let $L = ((C_i, v_i))_{i\in[m]}$ be a width-w decision list. Sort $[m] = \{j_1, \ldots, j_m\}$ such that $p_L(j_1) \geq p_L(j_2) \geq \cdots \geq p_L(j_m)$. For any $\varepsilon > 0$, let*

$$t = \left(2 + \frac{1}{w}\log\frac{1}{\varepsilon}\right)^{O(w)}.$$

Then for $J = \{j_1, \ldots, j_t, m\}$ it holds that $\Pr\left[\mathsf{Ind}L(x) \notin J\right] \leq \varepsilon$.

The proof of Theorem 2.2 follows immediately, by combining Lemma 2.7 and Claim 2.3.

2.2 Random Restrictions and Encoding

A *restriction* on n variables is some $\rho \in \{0, 1, *\}^n$. We say $x \in \{0, 1\}^n$ is consistent with $\rho \in \{0, 1, *\}^n$ if $x_i = \rho_i$ holds for all $i \in \rho^{-1}(0) \cup \rho^{-1}(1)$. An (n, k)-random restriction is the uniform distribution over restrictions $\rho \in \{0, 1, *\}^n$ with exactly k stars, which we denote by $\mathscr{R}(n, k)$. An (n, α)-random restriction, which we denote by $\mathscr{U}(n, \alpha)$, assigns independently each bit of the restriction ρ to $0, 1, *$ with probabilities $\frac{1-\alpha}{2}, \frac{1-\alpha}{2}, \alpha$, respectively.

Given a decision list $L\colon \{0, 1\}^n \to V$, its restriction under ρ is $L\restriction_\rho\colon \{0, 1\}^{\rho^{-1}(*)} \to V$. Formally, let

$$I = \{i \in [m] \mid \exists x \in \{0, 1\}^n \text{ consistent with } \rho, \mathsf{Ind}L(x) = i\}$$

and sort $I = \{i_1, \ldots, i_{m'}\}$ such that $1 \leq i_1 < \cdots < i_{m'} \leq m$. Then $L\restriction_\rho = ((\tilde{C}_{i_j}, v_{i_j}))_{j\in[m']}$ where $\tilde{C}_{i_j}$ is C_{i_j} simplified by fixing $x_i = \rho_i$ for all $i \in \rho^{-1}(0) \cup \rho^{-1}(1)$. Pedantically, we set $\tilde{C}_{i_{m'}} = \mathsf{True}$.

Definition 2.8 (Useful Probability). Given size-m decision list L and $\alpha \in (0, 1)$, the useful probability of an index $i \in [m]$ is

$$q_L(\alpha, i) := \Pr_{\rho\sim\mathscr{U}(n,\alpha)}\left[\text{index } i \text{ is useful in } L\restriction_\rho\right].$$

Note that we assume L initially does not contain useless rules, so for any α and i, we always have $q_L(\alpha, i) > 0$. We also have the following simple fact regarding useful probability.

Claim 2.9. *For any size-m decision list L, we have $\sum_{i=1}^{m} q_L(\alpha, i) = \mathbb{E}_{\rho\sim\mathscr{U}(n,\alpha)}[\#\mathsf{useful}\,(L\restriction_\rho)]$.*

Proof. Let $1_{\rho,i}$ be the indicator of index i being useful in $L\restriction_\rho$. Then

$$\mathop{\mathbb{E}}_{\rho\sim\mathscr{U}(n,\alpha)}\left[\#\mathsf{useful}\left(L\restriction_\rho\right)\right] = \mathop{\mathbb{E}}_{\rho}\left[\sum_{i=1}^{m} 1_{\rho,i}\right] = \sum_{i=1}^{m}\mathop{\mathbb{E}}_{\rho}\left[1_{\rho,i}\right] = \sum_{i=1}^{m} q_L(\alpha, i).$$ □

Now we present an encoding/decoding scheme for a random restriction (see Algorithm 1 and Algorithm 2) and analyze the expectation in Claim 2.9 explicitly. Let $\alpha \in (0,1)$ be such that αn is an integer. Define:

$$\mathcal{U} := \left\{ (\rho, s) \middle| \rho \in \mathscr{R}(n, \alpha n), s \in \left\{1, \ldots, \#\text{useful}\left(L \restriction_\rho\right)\right\} \right\}$$

$$\mathcal{V} := \left\{ (\rho', a) \middle| \rho' \in \bigcup_{k=0}^{\min\{w, \alpha n\}} \mathscr{R}(n, \alpha n - k), a \in \{\textsc{Old}, \textsc{New}\}^w \right\}.$$

We define two deterministic algorithms $\mathsf{Enc}\colon \mathcal{U} \to \mathcal{V}$ and $\mathsf{Dec}\colon \mathsf{Enc}(\mathcal{U}) \subseteq \mathcal{V} \to \mathcal{U}$ such that $\mathsf{Dec}(\mathsf{Enc}(\rho, s)) = (\rho, s)$ holds for any $(\rho, s) \in \mathcal{U}$. The encoding algorithm (Algorithm 1) takes in a restriction ρ and an index s, and modifies things to satisfy the sth useful term. The decoding algorithm (Algorithm 2) looks at the first always satisfied term from the encoding output and then recovers ρ and s.

ALGORITHM 1: Encoding algorithm $\mathsf{Enc}(\rho, s)$

Input: restriction and index $(\rho, s) \in \mathcal{U}$
Output: restriction and string $(\rho', a) \in \mathcal{V}$

1 $I \leftarrow \left\{ i \mid i \text{ is a useful index in } L \restriction_\rho \right\}$
2 $j \leftarrow$ the sth element in I
3 $\rho' \leftarrow \rho, a \leftarrow \varnothing$
/* Assume $C_j = \bigwedge_{k=1}^{c} y_{j_k}, y_{j_k} \in \left\{x_{j_k}, \neg x_{j_k}\right\}, c \le w$ */
4 **for** $k = 1$ **to** c **do**
5 **if** $\rho(x_{j_k}) \in \{0, 1\}$ **then**
6 Append a with Old /* x_{j_k} is already set by ρ */
7 **else**
8 Append a with New /* x_{j_k} is newly set to satisfy this term */
9 **if** $y_{j_k} = x_{j_k}$ **then** Update $\rho'(x_{j_k}) \leftarrow 1$ **else** Update $\rho'(x_{j_k}) \leftarrow 0$
10 **end**
11 **end**
12 Complete a arbitrarily to length w
13 **return** (ρ', a)

The following claim proves the correctness of the encoding and decoding algorithms.

Claim 2.10. $\mathsf{Dec}(\mathsf{Enc}(\rho, s)) = (\rho, s)$ *holds for any* $(\rho, s) \in \mathcal{U}$.

Proof. Sort literals in each term of $L = ((C_i, v_i))_{i \in [m]}$ arbitrarily. To justify the correctness, let $(\rho', a) = \mathsf{Enc}(\rho, s)$, then we need to ensure:

- $\mathsf{Dec}(\rho', a)$ obtains the same j in line 1 as $\mathsf{Enc}(\rho, s)$ does in line 2:
  During $\mathsf{Enc}(\rho, s)$, index j is useful in $L \restriction_\rho$, thus setting unfixed variables to satisfy C_j will not make any term C_i for $i < j$ satisfied. Hence the first satisfied term in $L \restriction_{\rho'}$ is C_j.
- $\mathsf{Dec}(\rho', a)$ in line 8 obtains the correct ρ:
  Since each term is sorted in advance, and a encodes which variable in C_j is set by $\mathsf{Enc}(\rho, s)$ rather than ρ, the loop in $\mathsf{Dec}(\rho', a)$ will set these variables back to $*$ and recover ρ. □

Corollary 2.11. $|\mathcal{U}| \le |\mathcal{V}|$.

Proof. Enc is an injection from $\mathcal{U}$ to $\mathsf{Enc}(\mathcal{U}) \subset \mathcal{V}$. □

ALGORITHM 2: Decoding algorithm $\mathrm{Dec}(\rho', a)$

Input: restriction and string $(\rho', a) \in \mathrm{Enc}(\mathcal{U}) \subseteq \mathcal{V}$
Output: restriction and index $(\rho, s) \in \mathcal{U}$

$j \leftarrow$ index of the first satisfied term in $L \restriction_{\rho'}$
$\rho \leftarrow \rho'$
/* Assume $C_j = \bigwedge_{k=1}^{c} y_{j_k}, y_{j_k} \in \left\{x_{j_k}, \neg x_{j_k}\right\}, c \leq w$ */
for $k = 1$ **to** c **do**
  if $a_k = \textsc{New}$ **then** /* x_{j_k} was not set by ρ */
    Update $\rho(x_{j_k}) \leftarrow *$
  end
end
$I \leftarrow \left\{i \mid i \text{ is a useful index in } L \restriction_{\rho}\right\}$
$s \leftarrow$ rank of j in I
return (ρ, s)

Lemma 2.12. *Let L be a width-w decision list on n variables and let $\alpha \in (0,1)$. Then*

$$\mathop{\mathbb{E}}_{\rho \sim \mathscr{U}(n,\alpha)} \left[\#\text{useful}\left(L \restriction_{\rho}\right)\right] \leq \left(\frac{4}{1-\alpha}\right)^{w}.$$

Proof. We first prove the bound for $\rho \sim \mathscr{R}(n, \alpha n)$ and then increase the number of variables to infinity, by adding dummy variables. This proves the desired bound as for $n' \to \infty$, the restriction of $\mathscr{R}(n', \alpha n')$ to the first n variables converges to $\mathscr{U}(n, \alpha)$. We have

$$\mathop{\mathbb{E}}_{\rho \sim \mathscr{R}(n,\alpha n)} \left[\#\text{useful}\left(L \restriction_{\rho}\right)\right] = \frac{1}{|\mathscr{R}(n, \alpha n)|} \sum_{\rho \in \mathscr{R}(n,\alpha n)} \#\text{useful}\left(L \restriction_{\rho}\right)$$

$$= \frac{|\mathcal{U}|}{|\mathscr{R}(n, \alpha n)|} \leq \frac{|\mathcal{V}|}{|\mathscr{R}(n, \alpha n)|} \leq \frac{\left(\sum_{k=0}^{w} \binom{n}{\alpha n - k} 2^{(1-\alpha)n+k}\right) \cdot 2^{w}}{\binom{n}{\alpha n} 2^{(1-\alpha)n}}$$

$$\leq \frac{\left(\sum_{k=0}^{w} \binom{n}{\alpha n - k}\right) \cdot 4^{w}}{\binom{n}{\alpha n}} \leq \frac{\binom{n+w}{\alpha n} \cdot 4^{w}}{\binom{n}{\alpha n}} \leq \left(\frac{4}{1-\alpha}\right)^{w}.$$

□

2.3 Noise Stability

We use noise stability to connect $p_L(i)$ and $q_L(\alpha, i)$.

Definition 2.13 (Noisy Distribution). Given $x \in \{0,1\}^n$ and a noise parameter $\beta \in (0,1)$, we denote by $\mathcal{N}_\beta(x)$ the distribution over $y \in \{0,1\}^n$, where $\Pr[y_i = x_i] = \frac{1+\beta}{2}, \Pr[y_i \neq x_i] = \frac{1-\beta}{2}$ independently for all $i \in [n]$.

Definition 2.14 (Stability). Let $g\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. The β-stability of g is

$$\mathrm{Stab}_\beta(g) = \Pr_{x \in \{0,1\}^n, y \sim \mathcal{N}_\beta(x)} [g(x) = g(y) = 1].$$

The hypercontractive inequality [7] (see also Reference [31], p. 259) allows us to bound the stability of a Boolean function by the probability that it outputs 1.

Fact 2.15. *Let $g\colon \{0,1\}^n \to \{0,1\}$ and $\beta \in (0,1)$. Then* $\mathrm{Stab}_\beta(g) \leq (\Pr[g(x) = 1])^{\frac{2}{1+\beta}}$.

Next, we define index stability and relate it to useful probability $q_L(\cdot,\cdot)$ and hit probability $p_L(\cdot)$.

Definition 2.16 (Index Stability). Given a size-m decision list L on n variables, the β-stability of index $i \in [m]$ is

$$\mathrm{Stab}_L(\beta, i) := \Pr_{x \in \{0,1\}^n, y \sim \mathcal{N}_\beta(x)} \left[\mathrm{Ind}L(x) = \mathrm{Ind}L(y) = i\right].$$

Lemma 2.17 (Bridging Lemma). *Let L be a size-m width-w decision list on n variables. Then for any index $i \in [m]$ and $\beta \in (0,1)$, we have*

$$\frac{p_L(i)^2}{q_L(1-\beta, i)} \le \mathrm{Stab}_L(\beta, i) \le p_L(i)^{\frac{2}{1+\beta}}.$$

Proof. We first prove the upper bound. Let $g\colon \{0,1\}^n \to \{0,1\}$ be an indicator Boolean function for $\mathrm{Ind}L(x) = i$. Then using Fact 2.15, we have

$$\mathrm{Stab}_L(\beta, i) = \mathrm{Stab}_\beta(g) \le \left(\Pr\left[g(x) = 1\right]\right)^{\frac{2}{1+\beta}} = \left(\Pr\left[\mathrm{Ind}L(x) = i\right]\right)^{\frac{2}{1+\beta}} = p_L(i)^{\frac{2}{1+\beta}}.$$

We now turn to prove the lower bound. Let $\alpha = 1 - \beta$. Observe that we can sample (x, y) where $x \in \{0,1\}^n, y \sim \mathcal{N}_\beta(x)$ as follows:

- Sample restriction $\rho \sim \mathscr{U}(n, \alpha)$;
- Sample uniform $x' \in \{0,1\}^{\rho^{-1}(*)}$ and complete stars in ρ with it to obtain x;
- Sample uniform $y' \in \{0,1\}^{\rho^{-1}(*)}$ and complete stars in ρ with it to obtain y.

We thus have

$$\mathrm{Stab}_L(\beta, i) = \Pr_{\rho, x', y'} \left[\mathrm{Ind}L \restriction_\rho (x') = \mathrm{Ind}L \restriction_\rho (y') = i\right].$$

We now make a seemingly redundant, but surprisingly useful, conditioning. Let $\mathcal{E}(\rho, i)$ denote the event

$$\mathcal{E}(\rho, i) := \left[i \text{ is useful in } L \restriction_\rho\right].$$

Then we can equivalently write

$$\mathrm{Stab}_L(\beta, i) = \Pr_{\rho, x', y'} \left[\mathrm{Ind}L \restriction_\rho (x') = \mathrm{Ind}L \restriction_\rho (y') = i \wedge \mathcal{E}(\rho, i)\right].$$

For any fixed ρ, define

$$r_\rho(i) := \Pr_{x'} \left[\mathrm{Ind}L \restriction_\rho (x') = i\right].$$

Since x', y' are independent for any fixed restriction, we have

$$\begin{aligned}
\mathrm{Stab}_L(\beta, i) &= \Pr_\rho[\mathcal{E}(\rho, i)] \cdot \Pr_{\rho, x', y'} \left[\mathrm{Ind}L \restriction_\rho (x') = \mathrm{Ind}L \restriction_\rho (y') = i \middle| \mathcal{E}(\rho, i)\right] \\
&= q_L(\alpha, i) \cdot \mathop{\mathbb{E}}_\rho \left[r_\rho(i)^2 \middle| \mathcal{E}(\rho, i)\right] \\
&\ge q_L(\alpha, i) \cdot \left(\mathop{\mathbb{E}}_\rho \left[r_\rho(i) \middle| \mathcal{E}(\rho, i)\right]\right)^2 && \text{(Jensen's inequality)} \\
&= \frac{1}{q_L(\alpha, i)} \left(q_L(\alpha, i) \cdot \mathop{\mathbb{E}}_\rho \left[r_\rho(i) \middle| \mathcal{E}(\rho, i)\right]\right)^2
\end{aligned}$$

$$= \frac{1}{q_L(\alpha, i)} \left(\Pr_{\rho, x'} \left[\mathrm{Ind}L \restriction_\rho (x') = i \wedge \mathcal{E}(\rho, i) \right] \right)^2$$
$$= \frac{1}{q_L(\alpha, i)} \left(\Pr_{\rho, x'} \left[\mathrm{Ind}L \restriction_\rho (x') = i \right] \right)^2$$
$$= \frac{1}{q_L(\alpha, i)} \left(\Pr_x [\mathrm{Ind}L(x) = i] \right)^2 = \frac{p_L(i)^2}{q_L(\alpha, i)}. \qquad \square$$

By rearranging terms in Lemma 2.17, we obtain the following corollary.

Corollary 2.18. *Let L be a size-m width-w decision list. Then for any index $i \in [m]$ and $\beta \in (0, 1)$, we have*

$$p_L(i) \le q_L(1 - \beta, i)^{\frac{1+\beta}{2\beta}}.$$

As a remark, we note that Lemma 2.17 can be generalized to arbitrary Boolean functions with a similar proof.

Lemma 2.19. *Let $g\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function that is not identically zero. Set $|g| = \Pr[g(x) = 1]$. Then for any $\beta \in (0, 1)$, we have*

$$\frac{|g|^2}{\Pr\limits_{\rho \sim \mathscr{U}(n, 1-\beta)} [g \restriction_\rho \not\equiv 0]} \le \mathrm{Stab}_\beta(g) \le |g|^{\frac{2}{1+\beta}}.$$

2.4 Putting Everything Together

Now we put everything together and give the proof of Lemma 2.7.

Proof of Lemma 2.7. Recall that we sorted $[m] = \{j_1, \ldots, j_m\}$ such that $p_L(j_1) \ge p_L(j_2) \ge \cdots \ge p_L(j_m)$. Let $J = \{j_1, \ldots, j_t, m\}$ for t to be optimized later.

Next, let $\beta \in (0, 1)$ to be optimized later and set $\alpha = 1 - \beta$. Sort $[m] = \{i_1, \ldots, i_m\}$ such that $q_L(\alpha, i_1) \ge q_L(\alpha, i_2) \ge \cdots \ge q_L(\alpha, i_m)$. By Claim 2.9 and Lemma 2.12, we have

$$\sum_{k=1}^{m} q_L(\alpha, i_k) = \mathop{\mathbb{E}}_{\rho \sim \mathscr{U}(n, \alpha)} \left[\#\mathrm{useful}\left(L \restriction_\rho\right) \right] \le \left(\frac{4}{1 - \alpha} \right)^w = \left(\frac{4}{\beta} \right)^w.$$

Note that we have sorted q_L in decreasing order, so

$$q_L(\alpha, i_k) \le \frac{1}{k} \left(\frac{4}{\beta} \right)^w.$$

Observe that $j_1, \ldots, j_t$ have the largest hit probability, and apply Corollary 2.18, then

$$\sum_{j \notin J} p_L(j) \le \sum_{k=t+1}^{m} p_L(j_k) \le \sum_{k=t+1}^{m} p_L(i_k) \le \sum_{k=t+1}^{m} q_L(\alpha, i_k)^{\frac{1+\beta}{2\beta}}$$
$$\le \left(\frac{4}{\beta} \right)^{w \cdot \frac{1+\beta}{2\beta}} \sum_{k \ge t+1} \left(\frac{1}{k} \right)^{\frac{1+\beta}{2\beta}} \le \left(\frac{4}{\beta} \right)^{w \cdot \frac{1+\beta}{2\beta}} \int_t^{+\infty} \left(\frac{1}{x} \right)^{\frac{1+\beta}{2\beta}} \mathrm{d}\, x$$
$$= \left(\frac{4}{\beta} \right)^{w \cdot \frac{1+\beta}{2\beta}} \cdot \frac{2\beta}{1 - \beta} \cdot t^{-\frac{1-\beta}{2\beta}}. \qquad \text{(since } \beta \in (0, 1)\text{)}$$

If we restrict $\beta \le 1/2$ and choose

$$t = \left(\frac{1}{\varepsilon} \right)^{\frac{2\beta}{1-\beta}} \left(\frac{4}{\beta} \right)^{w \cdot \frac{1+\beta}{1-\beta}} \left(\frac{2\beta}{1 - \beta} \right)^{\frac{2\beta}{1-\beta}} \le 4 \left(\frac{1}{\varepsilon} \right)^{4\beta} \left(\frac{4}{\beta} \right)^{3w},$$

then

$$\Pr\left[\mathrm{Ind}L(x) \notin J\right] = \sum_{j \notin J} p_L(j) \leq \varepsilon.$$

Now we divide ε into two cases. Assume $\varepsilon = 2^{-\ell w}$. Then $2 + \frac{1}{w}\log\frac{1}{\varepsilon} \geq \max\{2, \ell\}$ and:

- If $\ell \leq 2$, then we set $\beta = 1/2$ and obtain $t = 2^{O(w)} = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$.
- If $\ell \geq 2$, then we set $\beta = 1/\ell$ and obtain $t = \ell^{O(w)} = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$. □

3 LOWER BOUNDS

In this section, we prove two lower bounds for decision list compression, which show that the bounds in Theorem 1.1 are tight up to constants.

Claim 3.1. *For any w, there is a width-w decision list $L\colon \{0,1\}^w \to \{0,1\}$ such that*

$$\Pr\left[L(x) \neq L'(x)\right] > 1/3$$

for any width-w decision list L' of size at most $2^w/100w$. Moreover, this holds with high probability for a random function $L\colon \{0,1\}^w \to \{0,1\}$.

Proof. Since any Boolean function on w variables can be expressed as some width-w decision list, there are 2^{2^w} possible L. However, for any fixed L', it can approximate at most

$$\binom{2^w}{2^w/3} \cdot 2^{2^w/3} \leq 2^{0.97 \cdot 2^w}$$

different Boolean functions within distance $1/3$; and for fixed size m, there are at most $(3^w \cdot 2)^m$ distinct size-m width-w decision lists. As small-size decision lists can be embedded in larger ones, when restricted to size at most $2^w/100w$, width-w decision lists only approximate at most

$$(3^w \cdot 2)^{\frac{2^w}{100w}} \cdot 2^{0.97 \cdot 2^w} < 2^{2^w}$$

different Boolean functions on w variables. □

Claim 3.2. *For any w and $n \geq 16w$, there is a width-w decision list $L\colon \{0,1\}^n \to \{0,1\}$ that is not equivalent to any width-w decision list L' of size smaller than $\binom{n}{w}/n^2$.*

Proof. Let $m = \binom{n}{w}$ and sort all $\binom{n}{w}$ subsets of $[n]$ with size w as $\{S_1, \ldots, S_m\}$ arbitrarily. For any $i \in [m]$, define $C_i = \bigwedge_{j \in S_i} x_j$. For any $v \in \{0,1\}^m$, let $L_v = ((C_1, v_1), \ldots, (C_m, v_m), (1, 0))$ be a size-$(m+1)$ width-w decision list.

As small-size decision lists can be embedded in larger ones, assume towards a contradiction that any L_v is equivalent to some size-(m/n^2) width-w decision list L'_v. Given L'_v, we can recover L_v by enumerating all assignments, since all rules in L_v are useful. Thus, by a counting argument, the number of possible L'_v is upper bounded by

$$\begin{aligned}
\left(2 \cdot \sum_{k=0}^{w} 2^k \binom{n}{k}\right)^{\binom{n}{w}/n^2} &\leq \left(2 \cdot 2^w \sum_{k=0}^{w} \binom{n}{k}\right)^{\binom{n}{w}/n^2} \leq \left(2^{w+1} \cdot (w+1) \cdot \binom{n}{w}\right)^{\binom{n}{w}/n^2} && \text{(since } n \geq 2w\text{)} \\
&\leq \left(2^{2(w+1)} \cdot \binom{n}{w}\right)^{\binom{n}{w}/n^2} \leq \binom{n}{w}^{2\binom{n}{w}/n^2} && \text{(since } n \geq 16w \text{ and } \tbinom{n}{w} \geq \left(\tfrac{n}{w}\right)^w\text{)} \\
&= m^{2m/n^2} < 2^m.
\end{aligned}$$

□

Now the general lower bound follows immediately.

Corollary 3.3. *For any $w \geq 16$ and $\varepsilon \in (0, 1/3]$, there is a width-w decision list L such that*

$$\Pr\left[L(x) \neq L'(x)\right] > \varepsilon$$

holds for any width-w decision list L' of size at most

$$\left(2 + \frac{1}{w}\log\frac{1}{\varepsilon}\right)^{w/13}.$$

Proof. For $\varepsilon > 2^{-17w}$, let L be the decision list in Claim 3.1. Then it cannot be approximated within $\varepsilon < 1/3$ by a decision list L' of size at most

$$\frac{2^w}{100w} \geq 19^{w/13} \geq \left(2 + \frac{1}{w}\log\frac{1}{\varepsilon}\right)^{w/13}.$$

For $\varepsilon \leq 2^{-17w}$, let L be the decision list in Claim 3.2 with $n = \lfloor \log(1/\varepsilon)\rfloor - 1 \geq 16w$. Since now $\varepsilon < 2^{-n}$, the desired L' must be equivalent to L. Thus it cannot be realized by a decision list L' of size at most

$$\begin{aligned}\frac{\binom{n}{w}}{n^2} &\geq \frac{(n/w)^w}{n^2} \geq \frac{(\log(1/\varepsilon)/2)^{w-2}}{w^w} = \frac{1}{w^2}\cdot\left(\frac{1}{2w}\log\frac{1}{\varepsilon}\right)^{w-2}\\ &\geq 2^{-w/2}\left(\frac{1}{2w}\log\frac{1}{\varepsilon}\right)^{w/2} \geq \left(2 + \frac{1}{w}\log\frac{1}{\varepsilon}\right)^{w/6}.\end{aligned}$$

□

4 APPLICATIONS

In this section, we discuss some applications of our main theorem (Theorem 1.1).

4.1 DNF Sparsification

The decision list compression problem is a natural generalization of the *DNF sparsification* problem, introduced by Gopalan, Meka, and Reingold [15] as a means to obtain pseudorandom generators fooling small-width DNFs. Their main structural result can be summarized as follows.

Theorem 4.1 ([15]). *Any width-w DNF can be ε-approximated by a DNF of width w and size $(w\log(1/\varepsilon))^{O(w)}$.*

They conjectured that a better bound is possible.

Conjecture 4.2 ([15]). *Any width-w DNF can be ε-approximated by a DNF of width w and size $s(w, \varepsilon)$, where:*

- Weak version: $s(w, \varepsilon) = c(\varepsilon)^w$ *for some function c.*
- Strong version: $s(w, \varepsilon) = (\log(1/\varepsilon))^{O(w)}$.

The weak version was resolved by Lovett and Zhang [28], where they showed that $c(\varepsilon) = (1/\varepsilon)^{O(1)}$ suffices. Our main result, Theorem 1.1, verifies the strong version of their conjecture (and in fact, proves a sharper bound than the one conjectured).

Corollary 4.3 (This Work). *Any width-w DNF can be ε-approximated by a DNF of width w and size $(2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)} \leq (\log(1/\varepsilon))^{O(w)}$.*

We remark that Corollary 4.3 is also tight up to the constant in the $O(w)$ term. The proof is very similar to the proof in Section 3 that Theorem 1.1 is tight. We sketch the proof here as follows:

- For $2^{-2w} < \varepsilon \leq 1/3$, Claim 3.1 shows that most functions $f\colon \{0,1\}^w \to \{0,1\}$ cannot be $(1/3)$-approximated by any decision list of width w and size $O(2^w/w)$. In particular, f cannot be approximated by a DNF of width w and size $O(2^w/w)$. Note that f can trivially be computed by a DNF of width w and size 2^w, and that $2^{\Omega(w)} = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{\Omega(w)}$ in this regime.

- For $\varepsilon \leq 2^{-2w}$, consider computing the Threshold-w function[5] on $\log(1/\varepsilon)$ variables, which amounts to approximation with any error $< \varepsilon$. This requires a width-w DNF of size $\binom{\log(1/\varepsilon)}{w} = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{\Omega(w)}$.

4.2 Junta Theorem

A k-junta is a function depending on at most k variables. Friedgut's junta theorem [13] shows that Boolean functions of small influence can be approximated by juntas. For the relevant definitions see, for example, Reference [31].

Theorem 4.4 (Friedgut's Junta Theorem [13]). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function with total influence I. Then for any $\varepsilon > 0$, f can be ε-approximated by a k-junta for $k = 2^{O(I/\varepsilon)}$.*

It is well known that width-w DNFs have total influence $I = O(w)$, which implies by Theorem 4.4 that width-w DNFs can be ε-approximated by $2^{O(w/\varepsilon)}$-juntas. Since a width-w size-s decision list is a (sw)-junta, as a corollary of Theorem 1.1, we improve the bound, and generalize it to decision lists.

Corollary 4.5 (This Work). *Any width-w decision list can be ε-approximated by a k-junta for $k = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$.*

This improves previous bounds, even when restricted to DNFs or CNFs. By combining the results in References [15, 28] one gets the bound $k = \min\{w\log(1/\varepsilon), 1/\varepsilon\}^{O(w)}$ for width-w DNFs or CNFs. It can be verified that our new result is indeed better; for example for $\varepsilon = w^{-w}$ we obtain $(\log w)^{O(w)}$ instead of $w^{O(w)}$. It is also worthwhile noting that the result of Reference [28], which obtained the bound $(1/\varepsilon)^{O(w)}$, can be extended to decision lists with minimal changes.

4.3 Learning Small-width DNFs

For convenience we work with Boolean functions $f\colon \{\pm1\}^n \to \{\pm1\}$ in this subsection, where -1 represents `True` and $+1$ represents `False`.[6]

A class of Boolean functions is said to be (ε, δ)-PAC learnable using q queries if there exists a learning algorithm that, given query access to an unknown function in the class, returns with probability $(1 - \delta)$ a function that ε-approximates the unknown function, while making at most q queries. In our context we consider membership queries, where the learning algorithm can query the value of the unknown function on any chosen input.

A celebrated result of Jacskson [20] shows that polynomial-size DNFs are efficiently PAC learnable under the uniform distribution using membership queries. However, the best upper bound for learning DNFs under uniform distribution without membership queries is quasi-polynomial in Reference [39]. Meanwhile, Learning DNFs under arbitrary distribution with membership queries is hard under certain cryptographic assumptions [2]. See also Reference [12] for more details.

Theorem 4.6 (Jackson's Harmonic Sieve [20]). *The class of n-variate DNFs of size s is (ε, δ)-PAC learnable under the uniform distribution with $q = \mathrm{poly}(s, n, 1/\varepsilon, \log(1/\delta))$ membership queries and time complexity $\mathrm{poly}(s, n, 1/\varepsilon, \log(1/\delta))$.*

Using Theorem 1.1, we can extend Jackson's result to small-width DNFs. Prior to our work, we are not aware of any result on learning small-width unbounded-size DNFs. Note that the DNF

[5] A Threshold-w function on n variables is a Boolean function that outputs 1 if and only if the number of ones in the n input variables is at least w.

[6] In the $\{0,1\}$ setting, 1 represents `True` and 0 represents `False`. To convert it into the $\{\pm1\}$ setting, map b to $(-1)^b$ for $b \in \{0,1\}$.

sparsification bound from References [15, 28] also works here, if we replace the bound on s with their corresponding bound.

Corollary 4.7 (This Work). *The class of n-variate DNFs of width w is (ε, δ)-PAC learnable under the uniform distribution with $q = \mathrm{poly}(s, n, 1/\varepsilon, \log(1/\delta))$ membership queries and time complexity $\mathrm{poly}(s, n, 1/\varepsilon, \log(1/\delta))$, where $s = (2 + \frac{1}{w}\log\frac{1}{\varepsilon})^{O(w)}$.*

Proof. Jackson's algorithm combines a weak learner based on Fourier analysis, and a boosting algorithm that converts this weak learner to a strong learner. Let $f(x)$ be the target DNF that we are trying to learn. The weak learner solves the following problem: given a distribution $\mathcal{D}$ on $\{\pm 1\}^n$, output a set S such that the parity $\chi_S(x) = \prod_{i\in S} x_i$ is correlated with f under the distribution $\mathcal{D}$, i.e., $|\mathbb{E}_{x\sim\mathcal{D}}[f(x)\chi_S(x)]|$ is large. Initially $\mathcal{D}$ is the uniform distribution, but the boosting algorithm keeps adapting $\mathcal{D}$ to focus on inputs where it made many mistakes.

In Jackson's algorithm, the existence of such S is shown by observing that for a size-s DNF, at least one of the terms must be $1/s$ correlated to the function; and each term's contribution can be attributed to the parities supported on it.

Assume now that $f(x)$ is a width-w DNF with too many terms, so we cannot apply the previous argument directly. Apply Theorem 1.1 with error γ (to be determined soon), to obtain an approximate width-w DNF $g(x)$ that γ-approximates $f(x)$, where g has at most $s = (2 + \frac{1}{w}\log\frac{1}{\gamma})^{O(w)}$ terms. Crucially, we obtain g by removing some of the terms in f, and hence $g(x) = \mathtt{True}$ implies $f(x) = \mathtt{True}$ for any input x. In particular, $\Pr_{x\sim\mathcal{D}}[f(x) = -1] \geq \Pr_{x\sim\mathcal{D}}[g(x) = -1]$.

Assume that we know that the distribution $\mathcal{D}$ is not too far from uniform. Concretely, assume that $\mathcal{D}(x) \leq K2^{-n}$ for some parameter K. This implies that

$$\begin{aligned}\Pr_{x\sim\mathcal{D}}[f(x) = -1] &= \sum_x \mathcal{D}(x)\cdot\frac{1-f(x)}{2}\\ &\leq \sum_x \mathcal{D}(x)\cdot\frac{1-g(x)}{2} + \sum_{x:f(x)\neq g(x)} K2^{-n}\\ &\leq \Pr_{x\sim\mathcal{D}}[g(x) = -1] + \gamma K.\end{aligned}$$

We choose $\gamma = 1/12K$ and assume that $\Pr_{x\sim\mathcal{D}}[f(x) = -1] \in [1/3, 2/3]$, since otherwise the constant 1 function correlates with f under $\mathcal{D}$. Thus $\Pr_{x\sim\mathcal{D}}[g(x) = -1] \in [1/4, 3/4]$.

Next, we apply the same argument as in Reference [20, Fact 8] to show that some parity χ_S is $\Omega(1/s)$-correlated with f. First, note that for any term C of g and any input x, $C(x) = \mathtt{True}$ implies $g(x) = f(x) = \mathtt{True}$. Thus we can pick a term C of g such that

$$\Pr_{x\sim\mathcal{D}}[C(x) = -1] \geq \frac{1}{s}\cdot\Pr_{x\sim\mathcal{D}}[g(x) = -1] \geq \frac{1}{4s}.$$

Therefore, we have $\mathbb{E}_{x\sim\mathcal{D}}[f(x)\cdot(1 - C(x))/2] = \Pr_{x\sim\mathcal{D}}[C(x) = -1] \geq 1/4s$. By renaming variables, assume without loss of generality that C is given by

$$C(x) = \left(\bigwedge_{i=1}^{k} x_i\right) \wedge \left(\bigwedge_{j=1}^{t} \neg x_{k+j}\right) = 1 - 2\left(\prod_{i=1}^{k}\frac{1-x_i}{2}\right)\left(\prod_{j=1}^{t}\frac{1+x_{k+j}}{2}\right),$$

which means, after expanding the product, $(1 - C(x))/2$ is an average of parities (or negated parities). Thus by average, there exists some $S \subseteq [k + t]$ such that

$$\left|\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[f(x)\cdot\chi_S(x)]\right| \geq \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[f(x)\cdot\frac{1-C(x)}{2}\right] \geq \frac{1}{4s}.$$

Finally, we need to bound K. It is known (see, for example, Reference [24, Fact 14]) that boosting algorithms can be restricted to have $K = \varepsilon^{-O(1)}$, which completes the proof. □

5 DISCUSSION AND OPEN PROBLEMS

We showed that a small-width decision list can be approximated by a small-width decision list of small size, and that our bound is optimal up to constants. Here we mention several future directions and open problems.

Upper bound DNF compression. Recall that our compression method is to discard rules that are rarely hit under uniform distribution. Applied on a DNF f, we approximate it by a DNF g that *lower bounds* it, in the sense that $g(x) \leq f(x)$ holds for all inputs x. It is natural to ask for the other direction – can we get an upper bound approximation h for f (namely, $h(x) \geq f(x)$ for all x) similar guarantees as the ones we showed for lower bound approximation?

Gopalan, Meka, and Reingold [15] construct both lower and upper bound approximation for width-w DNFs, where their approximating DNFs have width w and size $(w \log \frac{1}{\varepsilon})^{O(w)}$. They use this sandwiching bound to obtain improved pseudorandom generators fooling small-width DNFs and a faster deterministic approximate counting algorithm for DNFs.

However, Lovett and Zhang [28] show that if the upper bound approximation can be constructed to have width w and size $(\frac{1}{\varepsilon} \log w)^{O(w)}$, then the bound in the famous sunflower lemma [11] can be greatly improved. Now that [1] obtains the improved sunflower bound, this should not be considered a barrier any more.

Mansour's conjecture. Mansour's conjecture [29] considers the question of approximating DNFs by Fourier-sparse polynomials. We say a Boolean function $f\colon \{\pm 1\}^n \to \{\pm 1\}$ can be ε-approximated by a polynomial of sparsity t if there exists a polynomial $p\colon \{\pm 1\}^n \to \mathbb{R}$ with at most t monomials (in the Fourier basis) such that

$$\mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} \left[(f(x) - p(x))^2 \right] \leq \varepsilon.$$

Conjecture 5.1 (Mansour's Conjecture). *For any $\varepsilon \in (0,1)$, there exists some $c = c(\varepsilon) > 0$ such that any size-s DNF can be ε-approximated by a polynomial of sparsity $c^{\log s}$.*

Mansour's conjecture, if true, would give an efficient agnostic learning algorithm for DNFs [14, 14]. As discussed in References [15, 28], DNF compression result shows the following conjecture is equivalent to Conjecture 5.1.

Conjecture 5.2. *For any $\varepsilon \in (0,1)$, there exists some $c = c(\varepsilon) > 0$ such that any width-w DNF can be ε-approximated by a polynomial of sparsity c^w.*

Apart from Reference [23], which verifies Mansour's conjecture for random DNFs, the best known bound for Conjecture 5.2 is $(w \log \frac{1}{\varepsilon})^{O(w)}$ [29], which also gives the best known bound for Conjecture 5.1 by approximating a size-s DNF with a width-$O(\log(s/\varepsilon))$ DNF.

Learning small-width decision lists. To extend Jackson's learning algorithm to small-width DNFs, we rely on the fact that the DNF approximator we obtain has only one-sided error. This, however, is not necessarily true in the decision list setting. More importantly, it is not even known if small-size decision lists can be efficiently learned under the uniform distribution with membership queries. Nonetheless, we wonder if our decision list compression result could be helpful for developing learning algorithms for small-width decision lists.

Derandomization of the switching lemma. A recent work of Kelley [22], improving upon Reference [37], shows the random positions of $*$'s in the Håstad's switching lemma can be efficiently

derandomized, thus improving the seed length of pseudorandom generators fooling small-width DNFs. It is natural to ask if this can also be achieved for our analog of the switching lemma (Lemma 2.12).

ACKNOWLEDGMENTS

We thank Benjamin Rossman for invaluable discussions. We also thank Ryan Alweiss and the anonymous reviewers for helpful suggestions on an earlier version of this article.

REFERENCES

[1] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. 2020. Improved bounds for the sunflower lemma. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC'20)*, Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (Eds.). ACM, 624–630. https://doi.org/10.1145/3357713.3384234

[2] Dana Angluin and Michael Kharitonov. 1995. When won't membership queries help? *J. Comput. Syst. Sci.* 50, 2 (1995), 336–355. https://doi.org/10.1006/jcss.1995.1026

[3] Vikraman Arvind, Johannes Köbler, Sebastian Kuhnert, Gaurav Rattan, and Yadu Vasudev. 2015. On the isomorphism problem for decision trees and decision lists. *Theor. Comput. Sci.* 590 (2015), 38–54.

[4] Giulia Bagallo and David Haussler. 1990. Boolean feature discovery in empirical learning. *Mach. Learn.* 5, 1 (1990), 71–99.

[5] Paul Beame. 1994. *A Switching Lemma Primer*. Technical Report. Technical Report UW-CSE-95-07-01, Department of Computer Science.

[6] Avrim Blum. 1992. Rank-r decision trees are a subclass of r-decision lists. *Inform. Process. Lett.* 42, 4 (1992), 183–185.

[7] Aline Bonami. 1970. Étude des coefficients de Fourier des fonctions de $L^p(G)$. *Annal. L'inst. Fourier* 20, 2 (1970), 335–402. http://eudml.org/doc/74019.

[8] Arkadev Chattopadhyay, Meena Mahajan, Nikhil S. Mande, and Nitin Saurabh. 2020. Lower bounds for linear decision lists. (unpublished).

[9] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. 1989. A general lower bound on the number of examples needed for learning. *Inf. Comput.* 82, 3 (1989), 247–261.

[10] Thomas Eiter, Toshihide Ibaraki, and Kazuhisa Makino. 2002. Decision lists and related boolean functions. *Theor. Comput. Sci.* 270, 1–2 (2002), 493–524.

[11] Paul Erdős and Richard Rado. 1960. Intersection theorems for systems of sets. *J. Lond. Math. Soc.* 35, 1 (1960), 85–90.

[12] Vitaly Feldman et al. 2007. *Efficiency and Computational Limitations of Learning Algorithms*. Vol. 68.

[13] Ehud Friedgut. 1998. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica* 18, 1 (1998), 27–35.

[14] Parikshit Gopalan, Adam Kalai, and Adam R. Klivans. 2008. A query algorithm for agnostically learning DNF? In *Proceedings of the 21st Annual Conference on Learning Theory (COLT'08)*, Rocco A. Servedio and Tong Zhang (Eds.). Omnipress, 515–516. http://colt2008.cs.helsinki.fi/papers/Gopalan-open-question.pdf.

[15] Parikshit Gopalan, Raghu Meka, and Omer Reingold. 2013. DNF sparsification and a faster deterministic counting algorithm. *Comput. Complex.* 22, 2 (2013), 275–310. https://doi.org/10.1007/s00037-013-0068-6

[16] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil P. Vadhan. 2012. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*. IEEE Computer Society, 120–129. https://doi.org/10.1109/FOCS.2012.77

[17] David Guijarro, Victor Lavin, and Vijay Raghavan. 2001. Monotone term decision lists. *Theor. Comput. Sci.* 259, 1–2 (2001), 549–575.

[18] Thomas Hancock, Tao Jiang, Ming Li, and John Tromp. 1996. Lower bounds on learning decision lists and trees. *Inf. Comput.* 126, 2 (1996), 114–122.

[19] Johan Håstad. 1987. *Computational Limitations of Small-depth Circuits*. MIT Press, Cambridge, MA.

[20] Jeffrey C. Jackson. 1997. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *J. Comput. Syst. Sci.* 55, 3 (1997), 414–440.

[21] Michael Kearns, Ming Li, Leonard Pitt, and Leslie Valiant. 1987. On the learnability of boolean formulae. In *Proceedings of the 19th Annual ACM Conference on Theory of Computing*, Vol. 1987. Citeseer, 285–295.

[22] Zander Kelley. 2020. An improved derandomization of the switching lemma. *Electron. Colloquium Comput. Complex.* 27 (2020), 182.

[23] Adam R. Klivans, Homin K. Lee, and Andrew Wan. 2010. Mansour's conjecture is true for random DNF Formulas. In *Proceedings of the 23rd Conference on Learning Theory (COLT'10)*, Adam Tauman Kalai and Mehryar Mohri (Eds.). Omnipress, 368–380.

[24] Adam R. Klivans and Rocco A. Servedio. 2003. Boosting and hard-core set construction. *Mach. Learn.* 51, 3 (2003), 217–238.
[25] Ron Kohavi and Scott Benson. 1993. Research note on decision lists. *Mach. Learn.* 13, 1 (1993), 131–134.
[26] Matthias Krause. 2006. On the computational power of boolean decision lists. *Comput. Complex.* 14, 4 (2006), 362–375.
[27] Nathan Linial, Yishay Mansour, and Noam Nisan. 1993. Constant depth circuits, fourier transform, and learnability. *J. ACM* 40, 3 (1993), 607–620. https://doi.org/10.1145/174130.174138
[28] Shachar Lovett and Jiapeng Zhang. 2019. DNF sparsification beyond sunflowers. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*. 454–460. https://doi.org/10.1145/3313276.3316323
[29] Yishay Mansour. 1995. An O(nˆ(log log n)) learning algorithm for DNT under the uniform distribution. *J. Comput. Syst. Sci.* 50, 3 (1995), 543–550. https://doi.org/10.1006/jcss.1995.1043
[30] Ziv Nevo and Ran El-Yaniv. 2002. On online learning of decision lists. *J. Mach. Learn. Res.* 3, (Oct. 2002), 271–301.
[31] Ryan O'Donnell. 2014. *Analysis of Boolean Functions*. Cambridge University Press.
[32] Alexander A. Razborov. 1995. Bounded arithmetic and lower bounds in boolean complexity. In *Feasible Mathematics II*. Springer, 344–386.
[33] Alexander A. Razborov. 2015. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Ann. Math.* (2015), 415–472.
[34] Ronald L. Rivest. 1987. Learning decision lists. *Mach. Learn.* 2, 3 (1987), 229–246.
[35] Nathan Segerlind, Sam Buss, and Russell Impagliazzo. 2004. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM J. Comput.* 33, 5 (2004), 1171–1200.
[36] Avishay Tal. 2017. Tight bounds on the fourier spectrum of AC0. In *Proceedings of the 32nd Computational Complexity Conference (CCC'17)*, Ryan O'Donnell (Ed.), LIPIcs, Vol. 79. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 15:1–15:31. https://doi.org/10.4230/LIPIcs.CCC.2017.15
[37] Luca Trevisan and Tongke Xue. 2013. A derandomized switching lemma and an improved derandomization of AC0. In *Proceedings of the 28th Conference on Computational Complexity (CCC'13)*. IEEE Computer Society, 242–247. https://doi.org/10.1109/CCC.2013.32
[38] György Turán and Farrokh Vatan. 1997. Linear decision lists and partitioning algorithms for the construction of neural networks. In *Foundations of Computational Mathematics*. Springer, 414–423.
[39] Karsten A. Verbeurgt. 1990. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT'90)*, Mark A. Fulk and John Case (Eds.). Morgan Kaufmann, 314–326. http://dl.acm.org/citation.cfm?id=92659.
[40] Fulton Wang and Cynthia Rudin. 2015. Falling rule lists. In *Artificial Intelligence and Statistics*. 1013–1022.
[41] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Received March 2020; revised March 2021; accepted September 2021