

Sharper bounds on the Fourier concentration of DNFs

Victor Lecomte

Stanford

vlecomte@stanford.edu

Li-Yang Tan

Stanford

liyang@cs.stanford.edu

October 19, 2021

Abstract

In 1992 Mansour proved that every size- s DNF formula is Fourier-concentrated on $s^{O(\log \log s)}$ coefficients. We improve this to $s^{O(\log \log k)}$ where k is the *read number* of the DNF. Since k is always at most s , our bound matches Mansour's for all DNFs and strengthens it for small-read ones. The previous best bound for read- k DNFs was $s^{O(k^{3/2})}$. For k up to $\tilde{\Theta}(\log \log s)$, we further improve our bound to the optimal $\text{poly}(s)$; previously no such bound was known for any $k = \omega_s(1)$.

Our techniques involve new connections between the term structure of a DNF, viewed as a set system, and its Fourier spectrum.

1 Introduction

The relationships between combinatorial and analytic measures of Boolean function complexity is the subject of much study. A classic result of this flavor is Mansour’s theorem [Man92], which shows that every size- s DNF formula is Fourier-concentrated on $s^{O(\log \log s)}$ coefficients (that is, it is well-approximated by a polynomial with $s^{O(\log \log s)}$ monomials). More precisely:

Mansour’s theorem. *For every size- s DNF f and every ε , the Fourier spectrum of f is ε -concentrated on $(s/\varepsilon)^{O(\log \log(s/\varepsilon) \log(1/\varepsilon))}$ coefficients.*

However, Mansour conjectured that this bound was not tight, and that the correct bound was actually polynomial in s .

Mansour’s conjecture. *For every size- s DNF f and every ε , the Fourier spectrum of f is ε -concentrated on $s^{O_\varepsilon(1)}$ coefficients.*

Our main result is a sharpening of Mansour’s theorem that takes the *read number* of the DNF into account. We say that a DNF is *read- k* if every variable occurs in at most k of its terms.

Theorem 1. *For every size- s read- k DNF f and every ε , the Fourier spectrum of f is ε -concentrated on $(s/\varepsilon)^{O(\log \log k \log(1/\varepsilon))}$ coefficients.*

Since k is always at most s , our bound matches Mansour’s for all DNFs (indeed, it slightly improves the dependence on ε) and strengthens it for small-read ones. The dependence on k in Theorem 1 is a doubly-exponential improvement of the previous best bound of $s^{O(k^{3/2})}$ for read- k DNFs [ST19]; this was in turn an exponential improvement of an $s^{O(16^k)}$ bound by Klivans, Lee, and Wan [KLW10], who gave the first nontrivial bounds for $k \geq 2$.

For small values of k , we further improve our bound to the optimal $\text{poly}(s)$:

Theorem 2. *For every size- s DNF f with read up to $\tilde{\Theta}(\log \log s)$ and for every ε , the Fourier spectrum of f is ε -concentrated on $(s/\varepsilon)^{O(\log(1/\varepsilon))}$ coefficients.*

Previously no $\text{poly}(s)$ bound was known for any $k = \omega_s(1)$ (even for constant ε).

Regarding the dependence on ε in these bounds, Mansour showed a lower bound $s^{\tilde{\Omega}(\log(1/\varepsilon))}$ on the sparsity of any polynomial that ε -approximates TRIBES, a read-once DNF formula.

Theorems 1 and 2 immediately yield faster membership query algorithms for agnostically learning small-read DNF formulas under the uniform distribution. This is via a powerful technique of Gopalan, Kalai, and Klivans [GKK08], showing that if every function in a concept class over $\{\pm 1\}^n$ can be ε -approximated by t -sparse polynomials, then it can be agnostically learned in time $\text{poly}(n, t, 1/\varepsilon)$. As this implication is blackbox and by now standard, we do not elaborate further.

1.1 Other related work

Recent work of Kelman, Kindler, Lifshitz, Minzer, and Safra [KKL⁺20] proves that every boolean function f is Fourier concentrated on $\mathbb{I}(f)^{O_\varepsilon(\mathbb{I}(f))}$ coefficients, where $\mathbb{I}(f)$ is the total influence of f . Since $\mathbb{I}(f) \leq O(\log s)$ for size- s DNFs f , this result also recovers Mansour’s bound as a special case and strengthens it for DNFs with small total influence (modulo the dependence on ε).

This result is incomparable to Theorems 1 and 2. On one hand it is more general, applicable to all functions rather than just DNF formulas. On the other hand, there are small-read DNFs that saturate the $\mathbb{I}(f) \leq O(\log s)$ bound (e.g. TRIBES is a read-once DNF with total influence $\Theta(\log s)$).

1.2 Our techniques

It is well known that small-size DNFs are well-approximated by small-width¹ DNFs (see [Fact 5](#) in the preliminaries), so in this discussion and most of the paper, we focus on the “width” version of the question: that is, showing Fourier concentration for a width- w DNF f .

Our main conceptual contribution is to bound Fourier coefficients $\widehat{f}(S)$ by a quantity that depends on how S relates to the term structure of f as a DNF: we bound $|\widehat{f}(S)|$ by the probability over a random input x that S can be “covered” by the terms that x satisfies² in f (that is, the probability that each variable of S occurs in some satisfied term). Let us call this probability the *cover probability* of S . We use this bound on $|\widehat{f}(S)|$ twice, to prove the two main ingredients of our proof: a Fourier 1-norm bound ([Lemma 20](#)) and a 2-norm bound ([Lemma 23](#)).

The next three headings describe what happens in Sections 3 through 5.

The 1-norm bound. The first ingredient is a sharpening of Mansour’s [\[Man92\]](#) bound on the Fourier 1-norm due to low-degree monomials. The broad structure of the proof in [\[Man92\]](#) is to first show that f is concentrated up to degree $O(w)$, then to show that the Fourier 1-norm up to that degree is at most $w^{O(w)}$, which gives concentration on the same number of coefficients. As Mansour himself showed, this bound on the 1-norm is tight, even for read-once DNFs like TRIBES. Therefore, $w^{\Theta(w)}$ seemed to be the end of the story for 1-norm-based methods.

It turns out that we can make Mansour’s Fourier 1-norm bound more precise by splitting monomials x^S into groups \mathcal{S}_u based on (roughly) the size u of a minimal union of terms that includes S . We show a bound of $\binom{u}{O(w)}$ on the Fourier 1-norm due to \mathcal{S}_u . Note that this is a strict improvement on Mansour’s bound: the minimal cover of a set S of size $O(w)$ can involve at most $|S|$ terms and therefore will have total size at most $w|S| = O(w^2)$, in which case our bound $\binom{u}{O(w)}$ matches Mansour’s bound $w^{O(w)}$. On the other hand, for $u \ll w^2$, our bound $\binom{u}{O(w)}$ is much smaller than $w^{O(w)}$.

We prove this bound by tweaking Razborov’s [\[Raz95\]](#) proof of Håstad’s switching lemma [\[Hås87\]](#) to take this cover size u into account during the encoding phase. We first relate the absolute value of the Fourier coefficient $|\widehat{f}(S)|$ to the probability that a random restriction to S has decision tree depth $|S|$, then use Razborov’s encoding to show that this probability is small. Then, instead of separately identifying each variable of S by encoding their position within a term (which costs $\log w$ bits per variable), we encode their positions all together within the union of the terms they appear in (which costs $\binom{u}{|S|}$ in total).

The 2-norm bound. For the second ingredient, we give concrete bounds on the absolute value of the *individual* Fourier coefficients (as opposed to bounding their sum). Indeed, if the Fourier 1-norm due to a family of sets is $\leq M$ and, for each S in that family, $|\widehat{f}(S)| \leq \delta$, then the total Fourier weight due to that family is at most $M\delta$. In particular, if $M\delta \ll 1$, then we can simply discard that family. For $S \in \mathcal{S}_u$, we can bound $|\widehat{f}(S)|$ by (roughly) 2^{-u} times the number of ways to cover S by terms of f .

Concluding using small read. The read of f allows us to bound the number of ways a set can be covered by terms. We do this in two regimes:

¹the *width* of a DNF is the maximal length number of variables queried in a single term

²we say x “satisfies” some term if the values given by x make the term output true

- In general, if the read of f is k , then it is easy to see that any set S can only be (minimally) covered in $(k+1)^{|S|}$ ways. Thus we can bound the 2-norm due to family \mathcal{S}_u by $M\delta \leq \binom{u}{O(w)} 2^{-u} (k+1)^{O(w)}$, which is negligibly small for $u = \omega(w \log k)$. This means we can cut off at $u = O(w \log k)$, getting 1-norm at most $\binom{O(w \log k)}{O(w)} = (\log k)^{O(w)}$ for the remaining coefficients, and thus concentration on $(\log k)^{O(w)}$ coefficients ([Theorem 26](#)).
- If the read of f is small enough ($k \leq \frac{\log w}{\log \log w}$), then we can improve on the trivial $(k+1)^{|S|}$ bound by using a combinatorial result of [\[ST19\]](#) which states that the expected number of satisfied terms of an unbiased read- k DNF is $O(k)$. This allows us to cut off as low as $u = O(w)$, giving concentration on $2^{O(w)}$ coefficients, and thus proving Mansour’s conjecture for that entire family of functions ([Theorem 30](#)).

We note that this choice of coefficients (keeping only $\hat{f}(S)$ for sets S that are contained in a small union of terms) follows almost exactly the approach suggested by Lovett and Zhang [\[LZ19\]](#) for proving Mansour’s conjecture, although we did not end up using their sparsification result.

2 Preliminaries

In this section we define Boolean functions, the Fourier spectrum, and DNFs along with their complexity metrics (size, width, read). We also recall some facts that are used in Mansour’s original proof [\[Man92\]](#). For an in-depth treatment, see [\[O’D14, O’D12\]](#).

2.1 Boolean functions and Fourier analysis

We view Boolean functions as functions $f : \{-1, 1\}^n \rightarrow \mathbb{R}$, where an input of -1 represents “true” and an input of 1 represents “false”. However, for output values, we will use 1 for “true” and -1 for “false”, as usual.

This choice of input values makes the Fourier spectrum of f more convenient to define. For $S \subseteq [n]$, let

$$\hat{f}(S) = \mathbb{E}_{x \in \{-1, 1\}^n} [f(x) x^S]$$

where $x^S := \prod_{i \in S} x_i$. Any Boolean function is uniquely represented as a multilinear polynomial, where the coefficients are exactly the values $\hat{f}(S)$, which we call *Fourier coefficients*:

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) x^S.$$

We say f is ε -concentrated on a family $\mathcal{S} \subseteq 2^{[n]}$ if $\sum_{S \notin \mathcal{S}} \hat{f}(S)^2 \leq \varepsilon$, and we say that f is ε -concentrated on M coefficients if there is such an \mathcal{S} with $|\mathcal{S}| \leq M$. We define the *Fourier p -norm* of f as

$$\left(\sum_{S \subseteq [n]} |\hat{f}(S)|^p \right)^{1/p},$$

and the special case $p = 1$ has the following property:

Fact 3 (Exercise 3.16 in [O’D14]). Let $M = \sum_{S \subseteq [n]} |\hat{f}(S)|$ be the Fourier 1-norm of f , then f is ε -concentrated on M^2/ε coefficients.

Finally, we say a function $g : \{-1, 1\}^n \rightarrow \{0, 1\}$ ε -approximates a function $f : \{-1, 1\}^n \rightarrow \{0, 1\}$ if they differ on at most an ε fraction of inputs, that is,

$$\Pr_{x \in \{-1, 1\}^n} [f(x) \neq g(x)] \leq \varepsilon.$$

Fact 4 (Exercise 3.17 in [O’D14]). If g ε_1 -approximates f and is ε_2 -concentrated on a family \mathcal{S} , then f is $2(\varepsilon_1 + \varepsilon_2)$ -concentrated on \mathcal{S} .

2.2 DNFs

A function $f : \{-1, 1\}^n \rightarrow \{0, 1\}$ is a DNF if it can be represented as an OR of ANDs of the input variables. Each AND is called a *term*, and the number of terms is called the *size*. We write a size- s DNF as $f = T_1 \vee \dots \vee T_s$, and by abuse of notation, we frequently use T_j to represent the set of variables in the j^{th} term.

A DNF has *width* w if each of its terms queries at most w variables (i.e. $|T_j| \leq w$ for all j), and *read* k if each variable occurs in at most k terms. We only use w and k as upper bounds (except in [Theorem 27](#), where it is explicitly stated), which justifies us occasionally assuming “large enough w ” or “large enough k ”. A small-size DNF can be approximated by a small-width DNF:

Fact 5. Let f be a size- s DNF. Then there is a DNF g of width $\log(s/\varepsilon)$ that ε -approximates f . In fact, g is simply obtained from f by dropping some terms from f , so g ’s size and read are both at most f ’s.

This means that to prove [Theorem 1](#) and [Theorem 2](#), it is enough to prove the corresponding statements for width w : respectively, that width- w read- k DNFs are ε -concentrated on $2^{O(w \log \log k \log 1/\varepsilon)}$ coefficients ([Theorem 26](#)), and that width- w DNFs with read up to $\Omega(\log w)$ are ε -concentrated on $2^{O(w \log 1/\varepsilon)}$ coefficients ([Theorem 30](#)).

2.3 Restrictions and Håstad’s switching lemma

Let $S \subseteq [n]$ be a set of variables, let $\bar{S} = [n] \setminus S$, and let $x_{\bar{S}} \in \{-1, 1\}^{\bar{S}}$ be an assignment to only the variables in \bar{S} . Then the *restriction* of f to S at $x_{\bar{S}}$ is the function $f_{S|x_{\bar{S}}} : \{-1, 1\}^S \rightarrow \mathbb{R}$ that maps $x_S \in \{-1, 1\}^S$ to $f(x_S \circ x_{\bar{S}})$, where \circ denotes the act of combining vectors x_S and $x_{\bar{S}}$ into a vector of $\{-1, 1\}^{S \cup \bar{S}} = \{-1, 1\}^n$. We call the variables of S “free” and the variables of \bar{S} “fixed”.

Let $\text{DT}(f)$ denote f ’s *decision tree depth*: the smallest depth of a decision tree computing f exactly. Håstad’s switching lemma [[Hås87](#)] states that a random restriction of f (where both S and $x_{\bar{S}}$ are chosen randomly) is unlikely to have high decision tree depth. It is used twice in the proof of Mansour’s theorem [[Man92](#)]: once to show that f is concentrated on low-degree monomials ([Fact 6](#) below), and once to show that after removing the high-degree monomials, the Fourier 1-norm is low.

Fact 6. There is a constant $C > 1$ such that any width- w DNF f is ε -concentrated up to degree $Cw \log 1/\varepsilon$.

In the next section, we prove a variant of Håstad’s switching lemma in order to improve on the second application. Since the proof format of switching lemmas is quite unusual and complex, it is definitely helpful to be familiar with the proof of the original version beforehand. An excellent pedagogical presentation of the proof can be found in [[O’D09](#)].

2.4 Miscellaneous

We use \log to denote the base-2 logarithm (though the base will rarely matter). For a finite set S , we define $\binom{S}{k}$ to be the family of subsets of S that have size k . Finally, for a finite alphabet A , we use A^* to denote the set of strings over A , and we write the empty string as $()$.

3 Cover sizes and the switching lemma

As we mentioned before, the proof of Mansour’s theorem works by first proving concentration on degree up to $O(w \log 1/\varepsilon)$, then showing that the Fourier 1-norm due to monomials of degree at most d is at most $w^{O(d)}$. In this paper, we use the first part as is, and focus on improving the second part: the $w^{O(d)}$ bound on the Fourier 1-norm for degree at most d . The only fact we will ever use about the Fourier spectrum of f is the following lemma, which bounds the absolute value $|\widehat{f}(S)|$ of the Fourier coefficient of S by the probability that a random restriction to S has decision tree depth $|S|$.

Lemma 7. *Let $S \subseteq [n]$. Then $|\widehat{f}(S)| \leq \Pr_{x_{\bar{S}} \in \{-1,1\}^{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = |S|]$.*

Proof. First, observe that for any Boolean function $g : \{-1,1\}^m \rightarrow \{0,1\}$, if $\text{DT}(g) < m$, then $\widehat{g}([m]) = 0$ (indeed, the degree of g is at most $\text{DT}(g)$), and in all cases $|\widehat{g}([m])| \leq 1$. Because of this, $\widehat{g}([m]) \leq \mathbf{1}[\text{DT}(g) = m]$, and applying this to $g := f_{S|x_{\bar{S}}}$, we get

$$|\widehat{f}(S)| = \left| \mathbb{E}_{x_{\bar{S}}}[\widehat{f_{S|x_{\bar{S}}}}(S)] \right| \leq \mathbb{E}_{x_{\bar{S}}} \left[|\widehat{f_{S|x_{\bar{S}}}}(S)| \right] \leq \Pr_{x_{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = |S|]. \quad \square$$

Remark 8. For intuition, we think it is helpful to mentally replace the probability in [Lemma 7](#) with the “cover probability” of S which we mentioned in [Section 1.2](#): the probability over a random input x that every variable of S is present in at least one term that x satisfies. More broadly, the notion of “cover by terms” will be a key player throughout this whole paper. A formal link between these two probabilities is that when $\text{DT}(f_{S|x_{\bar{S}}}) = |S|$, it is possible to assign the variables of S to make sure that they are all involved in at least one satisfied term.³ This directly implies the following fact (although we will not use in this paper).

Fact 9. *Let $S \subseteq [n]$. Then $|\widehat{f}(S)| \leq 2^{|S|} \Pr_{x \in \{-1,1\}^n}[S \text{ is covered by satisfied terms}]$.*

For pedagogical reasons, we first reprove Mansour’s bound on the Fourier 1-norm at degree d as a warmup, but using our [Lemma 7](#). We will then slightly tweak the proof so that it tells us more about which coefficients contribute the most to the 1-norm (within a given degree d).

We start by summing up [Lemma 7](#) over all sets S of size d , and get a bound on the Fourier 1-norm at degree d that depends on the number of restrictions of f to d variables that require full decision tree depth.

Lemma 10. $\sum_{S:|S|=d} |\widehat{f}(S)| \leq 2^{-(n-d)} \times \#\{(S, x_{\bar{S}}) : |S| = d \wedge \text{DT}(f_{S|x_{\bar{S}}}) = d\}$.

Proof. By [Lemma 7](#), $\sum_{S:|S|=d} |\widehat{f}(S)| \leq \sum_{S:|S|=d} \Pr_{x_{\bar{S}} \in \{-1,1\}^{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = d]$, and we can rewrite each probability in this sum as the average $2^{-(n-d)} \times \#\{x_{\bar{S}} : \text{DT}(f_{S|x_{\bar{S}}}) = d\}$. \square

³If f is monotone, then it is easy to see: the fact that $\text{DT}(f_{S|x_{\bar{S}}}) = |S|$ shows that all variables of S are present in some term that is alive in $f_{S|x_{\bar{S}}}$, and by assigning all variables of S to true, we can satisfy all those terms. For f non-monotone it is only slightly subtler.

This now allows us to reprove Mansour’s bound on the 1-norm, using Håstad’s switching lemma [Hås87]. In the following proof, the sets of free variables correspond exactly to the sets whose Fourier coefficients we are trying to bound, rather than random selections of a $\Theta(1/w)$ fraction of the variables, as is usual. This allows us to avoid using facts about how random restrictions affect the Fourier spectrum, and will later allow us to take into account properties of the specific sets S we are encoding.

Lemma 11. $\sum_{S:|S|=d} |\widehat{f}(S)| \leq \binom{wd}{d} 2^{2d} = w^{O(d)}$.

Proof. By Lemma 10, we only need to show

$$\#\{(S, x_{\bar{S}}) : |S| = d \wedge \text{DT}(f_{S|x_{\bar{S}}}) = d\} \leq 2^{n+d} \times \binom{wd}{d}.$$

We will follow Razborov’s [Raz95] version of the proof of the switching lemma. This version uses an encoding argument, which consists of an encoding algorithm and a decoding algorithm. The encoding algorithm will take as inputs a set S of size d and an assignment $x_{\bar{S}}$ of the remaining variables such that $\text{DT}(f_{S|x_{\bar{S}}}) = d$ (one of the couples $(S, x_{\bar{S}})$ that we want to show are rare). It will produce as outputs an assignment $x \in \{-1, 1\}^n$ of *all* the variables, an element $\sigma \in \binom{[wd]}{d}$ (indicating a d -size subset of $[wd]$), and a d -bit binary string $a \in \{-1, 1\}^d$. The decoding algorithm will uniquely recover S and $x_{\bar{S}}$ from (x, σ, a) , showing that the encoding is injective, and therefore that there are only

$$\underbrace{2^n}_{\text{number of } x\text{'s}} \times \underbrace{\binom{wd}{d}}_{\text{number of } \sigma\text{'s}} \times \underbrace{2^d}_{\text{number of } a\text{'s}}$$

possible values of $(S, x_{\bar{S}})$ such that $|S| = d$ and $\text{DT}(f_{S|x_{\bar{S}}}) = d$, as desired.

The encoding algorithm Algorithm 1 will identify a number of terms T_{j_1}, \dots, T_{j_l} which together contain every variable of S , and complete the partial assignment $x_{\bar{S}}$ in a way that makes those terms easy to find (with some “hints” stored in a). It then suffices to encode which variables of the union $T_{j_1} \cup \dots \cup T_{j_l}$ belong to S , which is the role of the second output σ . Note that this is different from Razborov’s encoding, instead of separately encoding the position of each variable within its term, we encode the positions of all of the variables at once, within the *union* of all of the terms T_{j_1}, \dots, T_{j_l} .

More precisely (but still in words), Algorithm 1 does the following:

- Initialize S' to S . S' will represent the set of free variables.
- Initialize two partial assignments $x^{\text{sat}}, x^{\text{dt}}$ to $x_{\bar{S}}$. x^{sat} will be the string sent to the encoder, and is progressively made to satisfy the terms T_{j_1}, \dots, T_{j_l} (which will be identified later). x^{dt} , on the other hand, will progressively assign the variables of S in a way that maximizes the decision tree depth.
- Initialize a Boolean string a , which will contain the set of changes that the decoder needs to make to the variables of S in order to go from x^{sat} to x^{dt} during the decoding.
- Initialize a string c , which will contain the union of the variables of T_{j_1}, \dots, T_{j_l} , *in the order that they appear* (i.e. c starts with the variables of T_{j_1} , then the variables of $T_{j_2} \setminus T_{j_1}$, etc.).
- While S' is not empty (i.e. there are free variables in x^{sat} and x^{dt}):

- Let T_j be the first term not fixed by x^{dt} (we will call j_1, \dots, j_l the successive values that j takes in this loop).
 - Let S_j be the set of free variables in T_j .
 - Let $x_{S_j}^{\text{sat}}$ be the assignment to the variables of S_j that make T_j satisfied.
 - Let $x_{S_j}^{\text{dt}}$ be the assignment to the variables of S_j that maximizes the remaining decision tree depth of f (after it is restricted by $x^{\text{sat}} \circ x_{S_j}^{\text{sat}}$).
 - Extend x^{sat} with $x_{S_j}^{\text{sat}}$ and x^{dt} with $x_{S_j}^{\text{dt}}$.
 - Remove the variables of S_j from S' .
 - Add $x_{S_j}^{\text{dt}}$ to string a .
 - Add to c the variables of T_j that were not contained in the previously identified terms.
- Let σ be the set of indices within c where the variables of S are located (σ is a subset of $[wd]$).
 - Return x^{sat} , σ , and a .

The decoding algorithm will progressively re-identify terms T_{j_1}, \dots, T_{j_l} by looking at the first satisfied term in f , and progressively replacing the assignments $x_{S_j}^{\text{sat}}$ (which satisfy T_j) by the assignments $x_{S_j}^{\text{dt}}$ (which maximize decision tree depth). It will identify which variables of T_j are part of S by using σ . Crucially, string c in the encoding algorithm, which represents the union $T_{j_1} \cup \dots \cup T_{j_l}$, follows the order of the terms, so even though at the r^{th} step the decoding algorithm knows only terms T_{j_1}, \dots, T_{j_r} , it can reconstruct the first $|T_{j_1} \cup \dots \cup T_{j_r}|$ characters of c , and therefore correctly recover the set S_r using σ .

More precisely (but still in words), Algorithm 2 does the following:

- Initialize the set of variables S to an empty set.
- Initialize string c to an empty string. This string replicates string c from the decoding algorithm, and will take the exact same sequence of values as the algorithm progresses.
- While $|S| < d$ (we have not found all the variables yet):
 - Let T_j be the first term satisfied by x .
 - Add to c all the variables of T_j that have not been added to it in previous iterations.
 - Within the variables newly added to c , look at the one whose indices within c are in σ , and call this set of variables S_j . This will be the same as set S_j in the encoding algorithm.
 - Replace x 's assignment of the variables of S_j by the values $a_{|S|+1}, \dots, a_{|S|+|S_j|}$. This replaces $x_{S_j}^{\text{sat}}$ by $x_{S_j}^{\text{dt}}$ within x (where $x_{S_j}^{\text{sat}}$ and $x_{S_j}^{\text{dt}}$ are defined in the encoding algorithm).
 - Extend S with S_j .
- Return S and the values of x on \bar{S} .

Algorithm 1: Encode($S, x_{\bar{S}}$)

```
 $S' \leftarrow S$ 
 $x^{\text{sat}}, x^{\text{dt}} \leftarrow x_{\bar{S}}$ 
 $a \leftarrow () \in \{-1, 1\}^*$  // empty Boolean string
 $c \leftarrow () \in [n]^*$  // empty string of variables
while  $S' \neq \emptyset$  do
   $j \leftarrow \min\{j : T_j(x^{\text{dt}}) \neq 0\}$  //  $T_j$  is the first term unfixed by  $x^{\text{dt}}$ 
   $S_j \leftarrow T_j \cap S'$  // the set of variables unfixed in  $T_j$ 
   $x_{S_j}^{\text{sat}} \leftarrow$  the assignment of  $S_j$  such that  $T_j(x_{S_j}^{\text{dt}} \circ x_{S_j}^{\text{sat}}) \equiv 1$ 
   $x_{S_j}^{\text{dt}} \leftarrow$  any assignment of  $S_j$  such that  $\text{DT}(f_{(S' \setminus S_j)}(x^{\text{dt}} \circ x_{S_j}^{\text{dt}})) = |S' \setminus S_j|$ 
   $x^{\text{sat}} \leftarrow x^{\text{sat}} \circ x_{S_j}^{\text{sat}}$ 
   $x^{\text{dt}} \leftarrow x^{\text{dt}} \circ x_{S_j}^{\text{dt}}$ 
   $S' \leftarrow S' \setminus S_j = S' \setminus T_j$ 
  Append  $a$  with  $x_{S_j}^{\text{dt}}$ 
  Append  $c$  with all variables of  $T_j$  that are not yet in  $c$ 
end
 $\sigma \leftarrow \{k \in |c| : c_k \in S\}$  // the positions of the variables of  $S$  within  $c$ 
return  $(x^{\text{sat}}, \sigma, a)$ 
```

Definition 12. Let $j_1 < \dots < j_l$ be the successive values taken by j in Algorithm 1.

Claim 13. The encoding algorithm runs successfully, and in particular,

- (i) at the start of each run of the while loop, x^{sat} and x^{dt} are both assignments of all variables except S' , and $\text{DT}(f_{S'|x^{\text{dt}}}) = |S'|$;
- (ii) $x_{S_j}^{\text{dt}}$ always exists;
- (iii) $S = S_{j_1} \cup \dots \cup S_{j_l}$;
- (iv) c contains all variables of S ;
- (v) $|c| \leq wd$.

Proof. (i) Clear by induction and the choice of $x_{S_j}^{\text{dt}}$.

(ii) By (i), $\text{DT}(f_{S'|x^{\text{dt}}}) = |S'|$, and there is always a way to assign $|S_j|$ variables without decreasing the decision tree depth by more than $|S_j|$.

(iii) Clear since $S_{j_r} \subseteq S$ for all $r \in [l]$ and by the end of the algorithm, $\emptyset = S' = S \setminus S_{j_1} \setminus \dots \setminus S_{j_l}$.

(iv) By (iii), $S = S_{j_1} \cup \dots \cup S_{j_l} \subseteq T_{j_1} \cup \dots \cup T_{j_l}$, and c is constructed to contain exactly the variables of $T_{j_1} \cup \dots \cup T_{j_l}$.

(v) Each T_{j_r} contains at least one variable of S , so $l < |S| = d$, and each term has at most w variables, so $|c| = |T_{j_1} \cup \dots \cup T_{j_l}| \leq wd$. \square

Algorithm 2: Decode(x, σ, a)

```
 $S \leftarrow \emptyset$   
 $c \leftarrow () \in [n]^*$   
while  $|S| < d$  do  
   $j \leftarrow \min\{j : T_j(x) = 1\}$  //  $T_j$  is the first term satisfied by  $x$   
  Append  $c$  with all variables of  $T_j$  that are not yet in  $c$   
   $S_j \leftarrow \{c_k : k \in \sigma \wedge k \leq |c|\} \setminus S$   
  Replace  $x$ 's assignment of  $S_j$  by the values  $a_{|S|+1}, \dots, a_{|S|+|S_j|}$   
   $S \leftarrow S \cup S_j$   
end  
return  $(S, x|_{\bar{S}})$ 
```

Claim 14. *The successive values that j and S_j take in $\text{Decode}(\text{Encode}(S, x_{\bar{S}}))$ are exactly j_1, \dots, j_l and S_{j_1}, \dots, S_{j_l} (i.e. the same values they took in $\text{Encode}(S, x_{\bar{S}})$).*

Proof. We will show by induction that when the r^{th} run of the while loop starts, we have

- $S = S_{j_1} \cup \dots \cup S_{j_{r-1}}$;
- c contains the variables of the union $T_{j_1} \cup \dots \cup T_{j_{r-1}}$;
- $x = x_{\bar{S}} \circ x_{S_{j_1}}^{\text{dt}} \circ \dots \circ x_{S_{j_{r-1}}}^{\text{dt}} \circ x_{S_{j_r}}^{\text{sat}} \circ \dots \circ x_{S_{j_l}}^{\text{sat}}$.

If this is the case r , then we argue that the first term satisfied by x at the start of the r^{th} run is T_{j_r} . Indeed,

- $x' := x_{\bar{S}} \circ x_{S_{j_1}}^{\text{dt}} \circ \dots \circ x_{S_{j_{r-1}}}^{\text{dt}}$ is exactly the value of x^{dt} in the r^{th} run of the while loop in $\text{Encode}(S, x_{\bar{S}})$;
- by **Claim 13(ii)**, f is undecided by x' , so $f_{(S_{j_r} \cup \dots \cup S_{j_l})|x'}$ has no satisfied term, and in particular, by the definition of T_{j_r} , all terms before T_{j_r} are *unsatisfied* by x' ;
- $x_{S_{j_r}}^{\text{sat}}$ is defined to satisfy T_{j_r} .

Therefore, j will be assigned to j_r , c will be appropriately extended, which will allow the algorithm to correctly recover S_{j_r} , add it to S , and replace the values of x on S_{j_r} by the values $x_{S_{j_r}}^{\text{dt}}$ stored in a , proving the inductive hypothesis for $r + 1$. \square

As a consequence, the first output of Decode is indeed $S_{j_1} \cup \dots \cup S_{j_r} = S$, and the second output must be $x_{\bar{S}}$, since the values of the variables outside of S were never modified by either algorithm. Therefore, for all valid $(S, x_{\bar{S}})$, we have $\text{Decode}(\text{Encode}(S, x_{\bar{S}})) = (S, x_{\bar{S}})$, which concludes the proof of **Lemma 11**. \square

The main cost in the above lemma is the $\binom{wd}{d}$ factor, where the wd comes from the fact that covering a set S of size d by terms of f can take up to d terms, for a total size wd . This suggests that one might get savings if we know that a set S is “typically” covered by terms whose union has size much less than wd . This motivates the following definitions.

Definition 15 ($\text{cover}(S, x_{\bar{S}}), u(S, x_{\bar{S}})$). Given $(S, x_{\bar{S}})$ such that $\text{DT}(f_{S|x_{\bar{S}}}) = |S|$, let $\text{cover}(S, x_{\bar{S}}) := \{j_1, \dots, j_l\}$ and $u(S, x_{\bar{S}}) := |T_{j_1} \cup \dots \cup T_{j_l}|$ where j_1, \dots, j_l are the successive values of j obtained when running $\text{Encode}(S, x_{\bar{S}})$ (just like in [Definition 12](#)).

Fact 16. For any $(S, x_{\bar{S}})$ such that $\text{DT}(f_{S|x_{\bar{S}}}) = |S|$,

- (i) $S \subseteq \bigcup_{j \in \text{cover}(S, x_{\bar{S}})} T_j$;
- (ii) $|\text{cover}(S, x_{\bar{S}})| \leq |S|$;
- (iii) for all $j \in \text{cover}(S, x_{\bar{S}})$, T_j is alive under the partial assignment $x_{\bar{S}}$ (not yet fixed to 0 or 1);
- (iv) for all $j \in \text{cover}(S, x_{\bar{S}})$, T_j contains at least one variable of S .

Proof. Clear by inspecting [Algorithm 1](#). □

Definition 17 ($\mathcal{S}_{d,u}$). Let $\mathcal{S}_{d,u}$ be the family of sets S of d variables such that the most frequent value of $\text{cover}(S, x_{\bar{S}})$ is u (breaking ties arbitrarily). In other words,

$$\mathcal{S}_{d,u} = \{S \subseteq [n] : |S| = d \wedge u = \arg \max_u \Pr_{x_{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u]\}.$$

When $S \in \mathcal{S}_{d,u}$, we get to assume that $\text{cover}(S, x_{\bar{S}}) = u$ for only a small extra factor. In doing this, we replace [Lemmas 7](#) and [10](#) by the following two lemmas.

Lemma 18. If $S \in \mathcal{S}_{d,u}$, then $|\widehat{f}(S)| \leq (wd + 1) \times \Pr_{x_{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u]$.

Proof. As we showed before, the maximum value of $u(S, x_{\bar{S}})$ is wd , so it can take at most $wd + 1$ values.⁴ Therefore, since u maximizes $\Pr_{x_{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u]$, we have

$$\Pr_{x_{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = d] \leq (wd + 1) \Pr_{x_{\bar{S}}}[\text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u].$$

We then conclude by [Lemma 7](#). □

Lemma 19. $\sum_{S \in \mathcal{S}_{d,u}} |\widehat{f}(S)| \leq (wd + 1) 2^{-(n-d)} \times \#\{(S, x_{\bar{S}}) : |S| = d \wedge \text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u\}$

Proof. Sum up [Lemma 18](#) then transform the probability into an average, similar to [Lemma 10](#). □

We can now run the same encoding argument, but with the guarantee that $|T_{j_1} \cup \dots \cup T_{j_l}| = u$, to show the following more specialized 1-norm bound.

Lemma 20. $\sum_{S \in \mathcal{S}_{d,u}} |\widehat{f}(S)| \leq (wd + 1) \binom{u}{d} 2^{2d} = \left(\frac{u}{d}\right)^{O(d)}$.

Proof. Same as the proof of [Lemma 11](#), but now with a bound of u on the final size of c , which allows us to pick σ from the smaller set $\binom{[u]}{d}$. □

⁴In fact at most $wd - d + 1$ values, since $u(S, x_{\bar{S}})$ is always at least d .

4 In how many ways can a set of variables be covered by terms?

At this point, a fair question would be: what was the point of proving these refined bounds for the Fourier 1-norm based on the typical cover size u if the worst case ($d = w$, $u = wd$) gives $w^{\Theta(w)}$ anyway? To see how this is useful, let us look at the contribution of family $\mathcal{S}_{d,u}$ to the Fourier weight:

$$\sum_{\mathcal{S}_{d,u}} \widehat{f}(S)^2 \leq \left(\sum_{\mathcal{S}_{d,u}} |\widehat{f}(S)| \right) \times \max_{\mathcal{S}_{d,u}} |\widehat{f}(S)| \leq (wd + 1) \binom{u}{d} 2^{2d} \max_{\mathcal{S}_{d,u}} |\widehat{f}(S)|. \quad (1)$$

If we can bound $\max_{\mathcal{S}_{d,u}} |\widehat{f}(S)|$ by something that decreases faster than $(wd + 1) \binom{u}{d} 2^{2d}$ increases, then we can bound $\mathcal{S}_{d,u}$'s contribution to the Fourier weight.

Now, how can we use the fact that $S \in \mathcal{S}_{d,u}$ to bound $|\widehat{f}(S)|$? Well, we know from [Lemma 18](#) that

$$|\widehat{f}(S)| \leq (wd + 1) \times \Pr_{x_{\bar{S}}} [\text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u]. \quad (2)$$

It turns out that we can bound the above probability by bounding the number of ways that one can cover S by a union of terms whose total size is u .

Definition 21. Given $S \in \mathcal{S}_{d,u}$, let

$$\text{numCovers}(S) := \# \left\{ \mathcal{T} \subseteq [m] : S \subseteq \bigcup_{j \in \mathcal{T}} T_j \wedge |\mathcal{T}| \leq |S| \wedge \left| \bigcup_{j \in \mathcal{T}} T_j \right| = u \right\}.$$

This roughly represents the “number of ways S can be covered by terms”, and as we will see, this is an upper bound on the number of possible values of $\text{cover}(S, x_{\bar{S}})$.

Lemma 22. Let $S \in \mathcal{S}_{d,u}$. Then

$$\begin{aligned} \Pr_{x_{\bar{S}}} [\text{DT}(f_{S|x_{\bar{S}}}) = d \wedge u(S, x_{\bar{S}}) = u] &\leq 2^{-(u-d)} \# \{ \text{cover}(S, x_{\bar{S}}) : \text{DT}(S, x_{\bar{S}}) = d \wedge u(S, x_{\bar{S}}) = u \} \\ &\leq 2^{-(u-d)} \text{numCovers}(S). \end{aligned}$$

Proof. Let us first show the first inequality. Let $\{j_1, \dots, j_i\}$ be the value of $\text{cover}(S, x_{\bar{S}})$ for some $x_{\bar{S}}$ such that $\text{DT}(f_{S|x_{\bar{S}}}) = d$ and $u(S, x_{\bar{S}}) = u$. Then by [Fact 16\(iii\)](#), terms T_{j_1} must all be alive under partial assignment $x_{\bar{S}}$. This constraint fixes the values of all variables in $(T_{j_1} \cup \dots \cup T_{j_i}) \setminus S$, of which there are $u - d$, so this value of $\text{cover}(S, x_{\bar{S}})$ can only contribute $2^{-(u-d)}$ to the probability. This shows the first inequality. The second inequality is a consequence of [Fact 16\(i\)](#) and (ii). \square

Putting together (1), (2) and [Lemma 22](#), we obtain the following.

Lemma 23. $\sum_{\mathcal{S}_{d,u}} \widehat{f}(S)^2 \leq (wd + 1)^2 \binom{u}{d} 2^{3d} 2^{-u} \times \max_{|S|=d} \text{numCovers}(S)$.

Among the factors in front of the max, 2^{-u} is the one that will dominate. So if we can prove that $\max_{|S|=d} \text{numCovers}(S) \ll 2^u$, that is, that there are significantly less than 2^u ways to cover a set of d variables by a union of terms of total size u , then we can show that $\mathcal{S}_{d,u}$'s contribution to the Fourier weight is negligible.

5 Proof of our main theorems

Let us summarize the approach. We approximate f in three ways:

1. First, **Fact 6** tells us that f is $(\varepsilon/3)$ -concentrated on degree at most $Cw \log(3/\varepsilon)$ (for some constant $C > 1$). In particular, this shows that f is $(\varepsilon/3)$ -concentrated on the union of the families $\mathcal{S}_{d,u}$ for $d \leq Cw \log(3/\varepsilon)$.
2. Second, using **Lemma 23** we will show that among those families $\mathcal{S}_{d,u}$, f is $(\varepsilon/3)$ -concentrated on the families with $u \leq u^*$ for some u^* (for **Theorem 26**, we obtain $u^* = O(w \log k \log 1/\varepsilon)$, and for **Theorem 30**, we obtain $u^* = O(w \log 1/\varepsilon)$). In other words, we will prove

$$\sum_{u=\lfloor u^* \rfloor + 1}^{\infty} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 \leq \varepsilon/3.$$

3. Finally, using **Lemma 20**, we will show that the Fourier 1-norm for $u \leq u^*$ is at most some quantity M . In other words, we will prove

$$\sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} |\widehat{f}(S)| \leq M.$$

By **Fact 3**, this implies that the sum of the corresponding monomials is $(\varepsilon/3)$ -concentrated on $3M^2/\varepsilon$ coefficients.

Those three approximations together will show that the original function f is $(\varepsilon/3 + \varepsilon/3 + \varepsilon/3 = \varepsilon)$ -concentrated on $3M^2/\varepsilon$ coefficients.

To make our job in step 2 slightly easier in advance of proving **Theorem 26** and **Theorem 30**, let us sum up, specialize and simplify **Lemma 23**.

Corollary 24. *For large enough w , if $u \geq 100Cw \log(3/\varepsilon)$, then*

$$\sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 \leq 2^{-u/2} \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S).$$

Proof. By **Lemma 23** and ugly arithmetics, for large enough w ,

$$\begin{aligned} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 &\leq \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} (wd + 1)^2 \binom{u}{d} 2^{3d} 2^{-u} \times \max_{|S|=d} \text{numCovers}(S) \\ &\leq \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} (Cw^2 \log(3/\varepsilon) + 1)^2 \binom{u}{Cw \log(3/\varepsilon)} 2^{3Cw \log(3/\varepsilon)} 2^{-u} \\ &\quad \times \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S) \\ &\leq u^{O(1)} 2^{(\log(\frac{eu}{Cw \log(3/\varepsilon)}) + 3)Cw \log(3/\varepsilon)} 2^{-u} \times \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S) \\ &\leq 2^{-u/2} \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S). \quad \square \end{aligned}$$

5.1 General improvement to Mansour's theorem

How large can $\text{numCovers}(S)$ get for $|S| = d$ if f has read k ? In other words, how many ways are there to cover a set S by terms of a read- k DNF? By [Fact 16\(iv\)](#), each term in the cover must contain a variable of S , and each variable is present in at most k terms, so there are at most kd terms to choose from. In addition, by [Fact 16\(ii\)](#), the cover can contain at most d terms, so

$$\text{numCovers}(S) \leq \sum_{i=0}^d \binom{kd}{i} \leq \binom{kd+d}{d} \leq (e(k+1))^d = O(k)^d.$$

Now, looking at [Corollary 24](#), we see that we need to choose u^* big enough that this is much smaller than $2^{u^*/2}$. Thus it suffices to pick u^* to be about $\Theta(d \log k) = \Theta(Cw \log k \log 1/\varepsilon)$. The following lemma makes this precise

Lemma 25. For $u^* = 100Cw \log(k+2) \log(3/\varepsilon)$,

$$\sum_{u=\lfloor u^* \rfloor+1}^{\infty} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 \leq \varepsilon/3.$$

Proof. By [Corollary 24](#),

$$\begin{aligned} \sum_{u=\lfloor u^* \rfloor+1}^{\infty} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 &\leq \sum_{u=\lfloor u^* \rfloor+1}^{\infty} 2^{-u/2} \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S) \\ &\leq \sum_{u=\lfloor u^* \rfloor+1}^{\infty} 2^{-u/2} (e(k+1))^{Cw \log(3/\varepsilon)} \\ &\leq \sum_{u=\lfloor u^* \rfloor+1}^{\infty} 2^{-u/4} 2^{-25Cw \log(k+2) \log(3/\varepsilon)} 2^{\log(e(k+1))Cw \log(3/\varepsilon)} \\ &\leq \sum_{u=\lfloor u^* \rfloor+1}^{\infty} 2^{-u/4} \\ &\leq \frac{1}{1-2^{-1/4}} \times 2^{-25Cw \log(k+2) \log(3/\varepsilon)} \\ &\leq \varepsilon/3. \quad \square \end{aligned}$$

Now, all we have to do is to plug this value of u^* into [Lemma 20](#) to get the following theorem.

Theorem 26 (“width” version of [Theorem 1](#)). *Let f be a width- w , read- k DNF. Then f is ε -concentrated on $\log(k+2)^{O(w \log 1/\varepsilon)}$ coefficients.*

Proof. By [Fact 6](#), f is $\varepsilon/3$ -concentrated up to degree $Cw \log(3/\varepsilon)$, and by [Lemma 25](#), the coefficients in $\mathcal{S}_{d,u}$ for $d \leq Cw \log(3/\varepsilon)$ and $u > u^* = 100Cw \log(k+2) \log(3/\varepsilon)$ also only amount to Fourier

weight at most $\varepsilon/3$. In addition, by [Lemma 20](#) the remaining coefficients have total 1-norm at most

$$\begin{aligned}
\sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} |\widehat{f}(S)| &\leq \sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} (wd+1) \binom{u}{d} 2^{2d} \\
&\leq (u^*+1) \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} (wd+1) \binom{u^*}{d} 2^{2d} \\
&\leq (u^*+1)(Cw \log(3/\varepsilon)+1)(Cw^2 \log(3/\varepsilon)+1) \\
&\quad \times \left(\frac{100Cw \log(k+2) \log(3/\varepsilon)}{Cw \log(3/\varepsilon)} \right) 2^{2Cw \log(3/\varepsilon)} \\
&\leq (u^*+1)(w \log 1/\varepsilon)^{O(1)} (400e \log(k+2))^{Cw \log(3/\varepsilon)} \\
&= \log(k+2)^{O(w \log 1/\varepsilon)}.
\end{aligned}$$

Therefore, using [Fact 3](#) with error $\varepsilon/3$, f is ε -concentrated on $3(\log(k+2))^{O(w \log 1/\varepsilon)}/\varepsilon = \log(k+2)^{O(w \log 1/\varepsilon)}$ coefficients. \square

5.2 A proof of Mansour's conjecture for small enough read

In the previous subsection, we bounded the number of covers of S by $O(k)^d$, which was only small enough when $O(k)^d < 2^{u/4} \Leftrightarrow u = O(d \log k)$. If we want to prove Mansour's conjecture, we need to do better: we need to bound the number of covers by $2^{u/4}$ for any $u = \omega(d)$.

The way to achieve this is to bound the number of terms that can be involved in the cover. In the previous subsection, we simply observed that the cover is made of at most $|S| = d$ terms among the kd terms that contain variables of S . But when the read is small, we can do better.

To build some intuition, suppose that every term involves *exactly* w variables, rather than at most w . Since every variable can only occur in at most k terms, this would mean (by double counting) that any union of l terms has total size at least lw/k . Therefore, if we want the union to have size u , there can only be ku/w terms in it. Thus there would be at most

$$\binom{kd}{ku/w} \leq \binom{ku}{ku/w} \leq (ew)^{ku/w} = 2^{ku \log(ew)/w}$$

ways to cover S with total size u . As long as $k \leq \frac{w}{4 \log(ew)}$, this is at most $2^{u/4}$, and we easily get the following theorem.

Theorem 27. *Let f be a DNF whose terms are all conjunctions of exactly w variables, and suppose f has read $k \leq \frac{w}{4 \log(ew)}$. Then f is ε -concentrated on $2^{O(w \log 1/\varepsilon)}$ coefficients.*

Proof. By [Fact 6](#), f is $\varepsilon/3$ -concentrated up to degree $Cw \log(3/\varepsilon)$. Let $u^* = 100Cw \log(3/\varepsilon)$. Then,

by [Corollary 24](#),

$$\begin{aligned}
\sum_{u=\lfloor u^* \rfloor + 1}^{\infty} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 &\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S) \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} \binom{kCw \log(3/\varepsilon)}{ku/w} \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} \binom{ku}{ku/w} \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} 2^{ku \log(ew)/w} \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/4} \\
&\leq \frac{1}{1 - 2^{-1/4}} \times 2^{-25Cw \log 1/\varepsilon} \\
&\leq \varepsilon/3.
\end{aligned}$$

In addition, by [Lemma 20](#), the remaining coefficients have total 1-norm at most

$$\begin{aligned}
\sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} |\widehat{f}(S)| &\leq \sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} (wd + 1) \binom{u}{d} 2^{2d} \\
&\leq (u^* + 1)(Cw \log(3/\varepsilon) + 1)(Cw^2 \log(3/\varepsilon) + 1) 2^{u^*} 2^{2Cw \log(3/\varepsilon)} \\
&= 2^{O(w \log 1/\varepsilon)}.
\end{aligned}$$

Therefore, using [Fact 3](#) with error $\varepsilon/3$, f is ε -concentrated of $3(2^{O(w \log 1/\varepsilon)})^2/\varepsilon = 2^{O(w \log 1/\varepsilon)}$ coefficients. \square

However, this reasoning breaks down if some terms of f are allowed to have width smaller than w . For example, if f contained the one-variable term x_i for each $i \in S$, then the union could after all contain as many as d terms, rather than ku/w .

We get out of this issue (though not without some loss) by using the following lemma from [\[ST19\]](#), which tells us in essence that f cannot contain too many short terms without being very biased.

Fact 28 (Lemma 1.1 in [\[ST19\]](#), rephrased). *Let $f = T_1 \vee \dots \vee T_s$ be a read- k DNF. Then*

$$\sum_{j=1}^s 2^{-|T_j|} \leq k \ln \left(\frac{1}{1 - \Pr_x[f(x)]} \right).$$

For the purposes of ε -approximation, we can thus assume that

$$\sum_{j=1}^s 2^{-|T_j|} \leq k \ln 1/\varepsilon$$

(otherwise, we can simply approximate f by the constant 1 function).

We can now show that the union will be made of few terms by proving the following combinatorial lemma.

Lemma 29. *Let A_1, \dots, A_l be a family of finite sets such that*

- (i) $|A_1| + \dots + |A_l| \leq v$;
- (ii) $2^{-|A_1|} + \dots + 2^{-|A_l|} \leq F$,

with $v > F$. Then $l \leq \frac{4v}{\log(v/F)}$.

Proof. Intuitively, the two constraints are in direct tension: if we want to keep $|A_r|$ small, this makes $2^{-|A_r|}$ big, and vice versa. So we will show that each set A_r uses up a $\frac{\log(v/F)}{2v}$ fraction of the “budget” for either sum (i) or sum (ii). Concretely, for any A_r , either $|A_r| \geq \log(v/F)/2 \geq \left(\frac{\log(v/F)}{2v}\right)v$, or

$$2^{-|A_r|} \geq \sqrt{\frac{F}{v}} \geq \frac{F}{v} \times \frac{\log(v/F)}{2} = \left(\frac{\log(v/F)}{2v}\right)F,$$

where the second inequality comes from the fact that $\sqrt{x} \geq \frac{\log x}{2}$ for $x > 0$, applied to $x = v/F$. Both of those cases can only happen $\frac{2v}{\log(v/F)}$ times without violating either (i) or (ii), which means there can only be $l \leq \frac{4v}{\log(v/F)}$ sets in the family. \square

Since f is read- k , if a union of terms has size u , then the sum of the size of its terms is at most ku . In addition, by [Fact 28](#), the terms T_{j_1}, \dots, T_{j_l} forming the union must obey

$$2^{-|T_{j_1}|} + \dots + 2^{-|T_{j_l}|} \leq \sum_{j=1}^s 2^{-|T_j|} \leq k \ln 1/\varepsilon.$$

Therefore, we can apply [Lemma 29](#) with $v = ku$ and $F = k \ln 1/\varepsilon$ to show that there are at most

$$\left(\frac{kd}{\frac{4ku}{\log(u/\ln(1/\varepsilon))}}\right) \leq \left(\frac{ku}{\frac{4ku}{\log(u/\ln(1/\varepsilon))}}\right) \leq \left(\frac{e \log(u/\ln(1/\varepsilon))}{4}\right)^{\frac{4ku}{\log(u/\ln(1/\varepsilon))}} \leq 2^{ku \frac{4 \log \log(u/\ln(1/\varepsilon))}{\log(u/\ln(1/\varepsilon))}}$$

ways to cover S by a union of terms of total size u .

As long as $k \leq \frac{\log(u/\ln(1/\varepsilon))}{16 \log \log(u/\ln(1/\varepsilon))}$ for the smallest value of u we have to consider, this is at most $2^{u/4}$. We will set $u^* := 100Cw \log(3/\varepsilon)$, so the smallest value we have to consider is

$$\lfloor u^* \rfloor + 1 \geq 100Cw \log(3/\varepsilon) \geq w \ln(1/\varepsilon).$$

Therefore, $k \leq \frac{\log w}{16 \log \log w}$ suffices, and we get the following theorem.

Theorem 30 (“width” version of [Theorem 2](#)). *Let f be a width- w DNF that has read $k \leq \frac{\log w}{16 \log \log w}$. Then f is ε -concentrated on $2^{O(w \log 1/\varepsilon)}$ coefficients.*

Proof. The proof is very similar to the proof of [Theorem 27](#). By [Fact 6](#), f is $\varepsilon/3$ -concentrated up to degree $Cw \log(3/\varepsilon)$. Let $u^* := 100Cw \log(3/\varepsilon)$. Then, by [Corollary 24](#),

$$\begin{aligned}
\sum_{u=\lfloor u^* \rfloor + 1}^{\infty} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} \widehat{f}(S)^2 &\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} \max_{|S| \leq Cw \log(3/\varepsilon)} \text{numCovers}(S) \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} \left(\frac{kCw \log(3/\varepsilon)}{\frac{4ku}{\log(u/\ln(1/\varepsilon))}} \right) \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} \left(\frac{ku}{\frac{4ku}{\log(u/\ln(1/\varepsilon))}} \right) \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} 2^{ku \frac{4 \log \log(u/\ln(1/\varepsilon))}{\log(u/\ln(1/\varepsilon))}} \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/2} 2^{ku \frac{4 \log \log(u^*/\ln(1/\varepsilon))}{\log(u^*/\ln(1/\varepsilon))}} \\
&\leq \sum_{u=\lfloor u^* \rfloor + 1}^{\infty} 2^{-u/4} \\
&\leq \frac{1}{1 - 2^{-1/4}} \times 2^{-25Cw \log(3/\varepsilon)} \\
&\leq \varepsilon/3.
\end{aligned}$$

In addition, by [Lemma 20](#), the remaining coefficients have total 1-norm at most

$$\begin{aligned}
\sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} \sum_{S \in \mathcal{S}_{d,u}} |\widehat{f}(S)| &\leq \sum_{u=0}^{\lfloor u^* \rfloor} \sum_{d=0}^{\lfloor Cw \log(3/\varepsilon) \rfloor} (wd + 1) \binom{u}{d} 2^{2d} \\
&\leq (u^* + 1)(Cw \log(3/\varepsilon) + 1)(Cw^2 \log(3/\varepsilon) + 1) 2^{u^*} 2^{2Cw \log(3/\varepsilon)} \\
&= 2^{O(w \log 1/\varepsilon)}.
\end{aligned}$$

Therefore, using [Fact 3](#) with error $\varepsilon/3$, f is ε -concentrated of $3(2^{O(w \log 1/\varepsilon)})^2/\varepsilon = 2^{O(w \log 1/\varepsilon)}$ coefficients. \square

6 Conclusion

In this section, we present some open problems, and a direction that the results in this paper suggest.

6.1 Open problems

Besides the obvious open problem which is to prove Mansour's conjecture, we see two ways one could extend the results in our paper.

The first would be to improve the dependence on k in [Theorem 1](#). After all, if three exponential improvements were possible starting from [\[KLW10\]](#), how about a fourth? In fact, any significant improvement over the current dependence on k would strictly improve Mansour’s theorem even for general DNFs. Indeed, given that $k \leq s$, improving from $s^{O(\log \log k)}$ to, say, $s^{O(\log \log \log k)}$ would improve Mansour’s theorem from $s^{O(\log \log s)}$ to $s^{O(\log \log \log s)}$.

The second (and perhaps easier) option would be to prove Mansour’s conjecture for a bigger range of reads, improving on [Theorem 2](#). Indeed, in [Theorem 27](#), we showed that Mansour’s conjecture holds for k up to $\Omega(w/\log w)$ instead of $\Omega(\log w/\log \log w)$ if all terms have exactly w variables, instead of just at most w . To us, it intuitively feels like width exactly w is the “hardest case”, and it is hard to see how having shorter terms should not help a DNF have a much more spread-out Fourier spectrum, but we have not been able to make this intuition formal. In addition, our argument in [Theorem 2](#) does not feel tight: the way we use [\[ST19\]](#)’s lemma ([Fact 28](#)) feels “wasteful”, since we apply it to only the very few terms that are involved in covering some set S , rather than to the entire DNF. Because of this, we conjecture that [Theorem 2](#) can be improved with similar techniques to handle reads up to $\tilde{\Omega}(w) = \tilde{\Omega}(\log s)$ rather than the current $\tilde{\Omega}(\log \log s)$.

6.2 A structure vs pseudorandomness approach to Mansour’s conjecture?

A recent trend in solving hard combinatorics problems has been the “structure vs pseudorandomness” paradigm, which consists in decomposing an object into a “structured” part and a “pseudorandom” part, where “pseudorandom” can mean stand for property that a randomly drawn object would typically have. In particular, this paradigm has recently been used by Alweiss, Lovett, Wu and Zhang [\[ALWZ20\]](#) to improve bounds on the sunflower lemma from $w^{O(w)}$ to $(\log w)^{O(w)}$.

We think that a similar argument can be applied to Mansour’s conjecture. In fact, our techniques (and in particular, [Lemma 20](#)) suggest a natural candidate for what it means to a DNF to be “pseudorandom”: f is pseudorandom if for any set of variables S , there are few ways to cover S minimally using terms of f .⁵ Indeed, random DNFs where $\text{poly}(n)$ terms of size $\Theta(\log n)$ are drawn at random have this property (which gives an alternate proof of [\[KLW10\]](#)’s results about random DNFs).

However, some work remains to be done in order to deal with the “structured” case. Perhaps the most difficult case that remains unsolved is the random DNF where n terms of size $\log n$ are drawn at random *from the first $\log^2 n$ variables only*. Indeed, in this case, the overlaps between terms are so strong that [\[KLW10\]](#)’s techniques become applicable, and each set $S \subseteq [\log^2 n]$ of variables has many minimal covers by terms. In fact, we personally know researchers who have devoted significant amounts of time attempting to either prove Mansour’s conjecture for this DNF or use it as a counterexample. Therefore, we feel that this DNF is a natural next challenge to attack, and we feel optimistic that if someone manages to prove Mansour’s conjecture for it, they would be very close to proving the general case.

Acknowledgments

We thank the anonymous reviewers, whose comments have helped improve this paper. Li-Yang is supported by NSF CAREER Award 1942123.

⁵perhaps weighting covers with a factor 2^{-u} where u is the size of the cover’s union

References

- [ALWZ20] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 624–630, 2020. 6.2
- [GKK08] Parikshit Gopalan, Adam Kalai, and Adam Klivans. Agnostically learning decision trees. In *Proceedings of the 40th ACM Symposium on Theory of Computing (STOC)*, pages 527–536, 2008. 1
- [Hås87] Johan Håstad. *Computational limitations of small-depth circuits*. MIT press, 1987. 1.2, 2.3, 3
- [KKL⁺20] Esty Kelman, Guy Kindler, Noam Lifshitz, Dor Minzer, and Muli Safra. Towards a proof of the fourier-entropy conjecture? In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258, 2020. 1.1
- [KLW10] Adam R. Klivans, Homin K. Lee, and Andrew Wan. Mansour’s conjecture is true for random DNF formulas. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 368–380, 2010. 1, 6.1, 6.2
- [LZ19] Shachar Lovett and Jiapeng Zhang. Dnf sparsification beyond sunflowers. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 454–460, 2019. 1.2
- [Man92] Yishay Mansour. An $O(n^{\log \log n})$ learning algorithm for DNF under the uniform distribution. In David Haussler, editor, *Proceedings of the 5th Annual ACM Conference on Computational Learning Theory (COLT)*, pages 53–61, 1992. 1, 1.2, 2, 2.3
- [O’D09] Ryan O’Donnell. 15-855: Intensive Intro to Complexity Theory. Lecture 14: The Switching Lemma. <https://www.cs.cmu.edu/~odonnell/complexity/lecture14.pdf>, 2009. 2.3
- [O’D12] Ryan O’Donnell. Analysis of Boolean Functions at CMU. <https://www.youtube.com/playlist?list=PLm3J0oaFux3YypJNaF6sRAf2zC1QzMuTA>, 2012. 2
- [O’D14] Ryan O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. 2, 3, 4
- [Raz95] Alexander A Razborov. Bounded arithmetic and lower bounds in boolean complexity. In *Feasible Mathematics II*, pages 344–386. Springer, 1995. 1.2, 3
- [ST19] Rocco A. Servedio and Li-Yang Tan. Pseudorandomness for read- k DNF formulas. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 621–638, 2019. 1, 1.2, 5.2, 28, 6.1