# The Computational Complexity Column

by

**Vikraman Arvind**

Institute of Mathematical Sciences, CIT Campus, Taramani
Chennai 600113, India
http://www.imsc.res.in/~arvind

In the last few years there has been exciting progress on lower bounds for small depth circuits. A well-known result in the area is the depth-hierarchy theorem of Håstad based on random restrictions and the Håstad Switching Lemma. Rossman et al [14] have shown an average-case depth-hierarchy theorem using a new Switching Lemma based on random projections. This technique has lead to several new results in lower bounds for small depth circuits.

In this timely expository article, sharing many insights, Srikanth Srinivasan explains two switching lemmas, based on random projections, along with applications.

# On Some Recent Projection Switching Lemmas for Small Depth Circuits

Srikanth Srinivasan *

**Abstract**

*Switching Lemmas* are an important technique for analyzing and proving lower bounds for constant-depth Boolean circuits. Typically, in applying a switching lemma, we restrict the circuit under consideration by setting a few of the input variables to constants at random while leaving the other variables unset. A Switching lemma guarantees, roughly, that any small Boolean circuit simplifies considerably under such a restriction.

Recently, researchers have analyzed what happens to constant-depth circuits under *random projections*, where in addition to the above, we also identify some variables with each other. This analysis has yielded strong *Projection Switching Lemmas*, which have been used to prove some new results in Boolean circuit complexity, including the resolution of some long standing open problems in the area. We review some of these projection switching lemmas and their applications.

## 1 Introduction

Boolean Circuit complexity is a classical field of study in Computational Complexity Theory. A Boolean circuit is a combinatorial model of computation for computing a Boolean function $f : \{0,1\}^n \to \{0,1\}$. Starting with the input variables, the circuit computes $f$ by successively applying some "simple", predefined operations (such as ANDs, ORs etc.) until the function $f$ has been computed.

Formally, a circuit over a basis of Boolean functions $\mathcal{B}$ (e.g. say the family of AND and OR functions with any number of input variables) is a directed acyclic graph, whose leaves (i.e. sources) are labelled by the input variables — say $x_1, \ldots, x_n$ — and such that each internal node, called a *gate*, is labelled by a function from $\mathcal{B}$. On a given input, the value computed by each gate in the circuit can be defined inductively, with the leaves taking the values of the corresponding

This e-mail address is being protected from spambots. You need JavaScript enabled to view it
*Department of Mathematics, IIT Bombay. Email: srikanth@math.iitb.ac.in

input variables, and an internal node labelled with a function $g$ taking $g(b_1, \ldots, b_k)$ where $b_1, \ldots, b_k$ are the values computed by the gates that feed into $g$. In this way, each gate computes a Boolean function of the Boolean variables $x_1, \ldots, x_n$. The function computed by a designated gate called the *output gate* is defined to be the function computed by the circuit.

As with any computational model, there are natural notions of efficiency associated with Boolean circuits. By the *size* of a circuit, we refer to the number of gates in the circuit. The *depth* of a Boolean circuit is the longest path from an input variable to the output gate.

The principal problem in Boolean circuit complexity is the Lower Bound problem, which is the problem of finding explicit functions that cannot be computed by small (say polynomial-sized) circuits. While it is easy to see that a randomly chosen Boolean function does not have circuits of subexponential size, finding explicit examples (say, for a sequence of functions that can be computed by a uniform NP algorithm) of such functions can be a very challenging problem.

Starting from the early 1980s, there has been a large body of work in the area (see [4] or e.g. [11] for a more recent account), with particular emphasis on *constant-depth circuits* with various kinds of gates performing simple Boolean operations such as AND, OR, Threshold and modular computations.

In this article, we consider only functions made up of AND, OR and NOT gates. One of the early successes in circuit complexity was the proof of lower bounds for constant-depth circuits over this basis. The results of Ajtai [1] and Furst, Saxe and Sipser [6] showed that constant-depth circuits over this basis of polynomial size cannot compute the *Parity* function on $n$ variables; here, the Parity function is the function that computes the bitwise XOR of all its input variables. The results of [1, 6] were subsequently extended by Yao [17] and culminated in a near-optimal exponential lower bound due to Håstad [7].

The basic ideas behind these proofs are similar. The AND and OR functions can easily be forced to output 0 or 1 respectively by setting any single input variable to 0 or 1 respectively. However, the Parity function does not have this property: in fact, to force the XOR of $n$ Boolean variables to a constant, *all* the variables must be set to constants. Motivated by this observation, the idea behind the lower bound is to show that any small constant-depth circuit $C$ can be forced to a constant 0 or a constant 1 by a *restriction* that sets a strict subset of its input variables to constants, which implies that $C$ does not compute the Parity function.

In general, it is not clear how to set the input variables in order to do this, since there may be competing constraints. For example, if there is an AND gate which takes an input the variable $x_1$ we might want to set $x_1$ to 0 while an OR gate which takes the same variable as input indicates that setting $x_1$ to 1 might be a good idea. However, it turns out that setting a fraction of the input variables *independently and uniformly at random* satisfies all these constraints with high probability.

The key lemma that allows us to prove this is the *Switching Lemma* [7]. We review this fundamental result in Section 3, but for now we note that it implies that a *random restriction* of the above form simplifies any small constant-depth circuit by reducing its depth by 1. Applying the lemma repeatedly allows us to reduce the circuit to a constant by setting a strict subset of the input variables. This yields a lower bound for the Parity function as noted above.

While the above technique is good for proving lower bounds for the Parity function, we run into roadblocks when we try to prove analogous results for other functions. For example, consider the case when we have a function $f : \{0,1\}^n \to \{0,1\}$ for which we want to prove a depth-$d$ circuit lower bound, but the function itself can be computed by constant-depth circuits of small size, though of a depth larger than $d$. By the Switching Lemma, this function is *not* immune to the above random restrictions to the same extent as the Parity function (in fact, upon applying a random restriction, it turns into a constant with high probability) and hence, a different strategy is required to prove such a lower bound.

It turns out that this can be done, as shown by works of Sipser [15], Yao [17] and Håstad [7]. Håstad proved a strong *Depth-hierarchy theorem* which says that for every constant $d$, there is a function computed by a circuit $C$ of polynomial size and depth $d$ that cannot be computed by circuits of depth $d-1$ and subexponential size. The idea behind the proof is to design a special family of random restrictions where variables are not set uniformly at random but in a careful and correlated fashion so that the function computed by $C$ remains non-constant (and in fact, retains quite a bit of its structure), while at the same time the depth of any subexponential-sized depth-$(d-1)$ circuit reduces by 1. This requires a new Switching Lemma tailored to the modified family of restrictions.

Recently, in a breakthrough work, Rossman, Servedio and Tan [14] proved an *average-case* version of the Depth-hierarchy theorem (see Section 5 for more details), thus solving a long-standing open problem in the area [7]. In this work, Rossman et al. study a variant of the random restriction paradigm that they call *random projections* and their effects on constant-depth circuits. Random projections extend restrictions in the following sense: while restrictions either leave variables unset or set variables to constants 0 or 1, random projections also *identify* various subsets of variables (i.e. make them equal to each other). This allows for further simplification of the circuit being analyzed and offsets the effect of the correlations between the settings to different variables. Subsequent to the work of Rossman et al. [14], random projections were used by Chen, Oliveira, Servedio and Tan [5] to prove stronger lower bounds for constant-depth circuits testing small-distance connectivity in graphs, and also by Pitassi, Rossman, Servedio and Tan [13] for proving proof complexity lower bounds. The work of [14] was strengthened to higher depths by Håstad [9].

An important technical tool in each of these results is a *Projection Switching*

*Lemma* which shows that small constant-depth circuits simplify under random projections. We review two such Projection Switching lemmas due to Chen et al. [5] and Rossman et al. [14] and give an application to circuit lower bounds due to Chen et al. [5].

**Organization.** After some Preliminaries in Section 2, we review the proof of the Håstad Switching lemma. In Section 4, we prove a Projection Switching Lemma and use it to prove a strong constant-depth circuit lower bound for a family of functions that have polynomial-sized but larger depth circuits [5]. In Section 5, we state the average case depth hierarchy theorem of [14] and prove a weaker version of their Projection Switching Lemma.

# 2 Preliminaries

Throughout, we deal with constant-depth circuits made up of AND, OR and NOT gates. Unless otherwise mentioned, $n$ will denote the number of input variables and $d$ the depth of the circuit. We do not count the NOT gates in analyzing the depth of the circuit.

We will always assume that the NOT gates have only variable inputs, i.e. the circuit can be seen as made up of AND and OR gates only with $2n$ inputs corresponding to the positive literals $x_1, \ldots, x_n$ and negative literals $\neg x_1, \ldots, \neg x_n$. It is easy to check that any circuit can be transformed into this form without increasing the depth and increasing the size only by a constant factor.

We recall some well-known special subclasses of constant-depth circuits that will also be interesting.

- A *formula* is a circuit where the underlying Directed Acyclic graph is a rooted, directed tree.
- A CNF formula is a depth-2 formula where the output is an AND gate and its inputs are OR gates with literals feeding into them; the functions computed by the OR gates are *clauses*. A $k$-CNF formula is a CNF where each clause has at most $k$ literals.
- We define DNF formulas and $k$-DNF formula in the same way, except that the output gate is an OR and inputs to it are AND gates that compute *terms*.
- A Decision tree is a simple query model for computing Boolean functions. The model is defined to be a rooted full binary tree where each internal node is labelled by a Boolean variable, which is the variable queried at that node, and each leaf is labelled by 0 or 1. The computation path follows a

root to leaf path in the following way: at a node querying a variable $x_i$, the computation proceeds to the left child if $x_i$ takes the value 0 and to the right child otherwise. When the computation reaches a leaf, the value labelling that leaf is the value output by the function. The *height* of a decision tree is the longest root to leaf path in the tree.

Decision trees can be efficiently simulated by both DNFs and CNFs. More precisely, a decision tree of height at most $k$ can be computed by a $k$-CNF with at most $2^k$ clauses and also by a $k$-DNF with at most $2^k$ terms.

We use $D(f)$ to denote the minimal depth of a decision tree computing the function $f$.

The primary tools that we use to analyze constant-depth circuits are *restrictions* and *projections*.

A *restriction* on a set $X$ of variables is a string $\rho \in \{0,1,*\}^X$ (or equivalently a function $\rho : X \to \{0,1,*\}$). The meaning attached to a restriction is that variables $x \in X$ such that $\rho(x) \in \{0,1\}$ are set to the constant $\rho(x)$ whereas a variable $x$ such that $\rho(x) = *$ is left as is. Given a Boolean function $f : \{0,1\}^X \to \{0,1\}$, a restriction $\rho$ naturally defines a restriction of $f$ to a Boolean function of the variables $X' \subseteq X$. We use the notation $f \restriction \rho$ to denote this function.

Given restrictions $\rho_1, \ldots, \rho_s \in \{0,1,*\}^X$, we use the notation $\rho_1 \cdots \rho_s$ to denote the composed restriction $\rho$ where a variable $x$ is set to $*$ if it is set to $*$ by all the $\rho_i$s and otherwise set to $\rho_i(x)$ where $i = \min\{j \mid \rho_j(x) \neq *\}$.

Given another set of variables $Y$, we define a *projection* to be a restriction $\rho$ along with a function $\pi : X \to Y$. Here, we restrict the variables according to $\rho$ as above, but further, each variable $x \in \rho^{-1}(*)$ is replaced by $\pi(x) \in Y$. This process transforms $f : \{0,1\}^X \to \{0,1\}$ to a Boolean function over the variable set $Y$; we denote this function by $\mathrm{Proj}(f \restriction \rho)$. (When we use this notation, $Y$ and $\pi$ will be clear from context.)

A *random* restriction or projection is simply a probability distribution over the space of restrictions or projections respectively.

Let $\mathcal{R}$ be a random restriction over the variables in $X$. For each $\rho \in \{0,1,*\}^X$, we define the *weight of* $\rho$, denoted $\mathrm{wt}(\rho)$, to be the probability that $\rho$ is obtained on sampling a restriction according to the distribution $\mathcal{R}$. Given a set $\mathcal{B} \subseteq \{0,1,*\}^X$ of restrictions, we define its weight $\mathrm{wt}(\mathcal{B})$ to be the sums of the weights of the restrictions in $\mathcal{B}$. When this notation is used, distribution $\mathcal{R}$ will be clear from the context.

# 3 Håstad's Switching Lemma

In this section, we review the proof of the classical Håstad Switching lemma [7] since the statements and proofs of the projection switching lemmas build on this. We will follow the Razborov-style encoding proof of this statement as expounded by Beame [2] and its modification by Thapen [16].

Let $X = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. For a parameter $p \in [0, 1]$, let $\mathcal{R}_1(p)$ be the distribution of the restriction $\rho$ output by the following sampling algorithm: for each $i \in [n]$, independently set $\rho(x_i)$ to be $*$ with probability $p$ and 0 or 1 with probability $(1 - p)/2$ each.

Let us compute the weight of a restriction $\rho \in \{0, 1, *\}$ under the above random restriction. It can be checked that if a $\rho \in \{0, 1, *\}^X$ has exactly $a$ many $*$s, then we have

$$\mathrm{wt}(\rho) = p^a \left(\frac{1-p}{2}\right)^{n-a}. \tag{1}$$

**Lemma 1** (Håstad Switching Lemma)**.** *Assume $p \leq 1/10$. Let $F$ be a $k$-DNF over the variables in $X$. Then for $\rho$ sampled from $\mathcal{R}_1(p)$, we have*

$$\Pr_{\rho}[D(F \restriction \rho) \geq \ell] \leq (10pk)^{\ell}.$$

**Remark 2.** *While we have stated the above lemma for $k$-DNFs, it is easy to see that it holds for $k$-CNFs also. To see this, note that if $F$ is a $k$-CNF, then its negation $\neg F$ can be represented as a $k$-DNF. By the above lemma, with high probability, $\neg F \restriction \rho$ has a decision tree of depth at most $\ell$. However, since the class of functions with decision trees of depth at most $\ell$ is closed under complement, the same statement holds for $F$ as well.*

*Proof.* The proof of the lemma actually analyzes the following simple decision tree strategy for computing $F \restriction \rho$. Fix some arbitrary ordering of the terms of $F$ and inside each term, fix any ordering of the literals appearing in the term.

$\underline{\mathcal{T}_1(F, \rho)}$:

1. Choose the first term $T$ (according to the fixed ordering) that has not yet been fixed to 0 by $\rho$, i.e. such that $\rho$ has not yet falsified any literal appearing in $T$. If there is no such term, output 0.

2. Let $X' \subseteq X$ be the set of unset variables (i.e. variables from $\rho^{-1}(*)$) appearing in $T$. Query all the variables in $X'$.

3. If the term $T$ is set to 1 by the assignment to the variables in $X'$, output 1.

4. Otherwise, let $\tau \in \{0,1,*\}^X$ be the restriction that sets the variables in $X'$ according to the answers to the queries. Replace $\rho$ by $\rho\tau$ and go back to Step 1.

It is easy to check that for any $F$ and $\rho$, $\mathcal{T}_1(F,\rho)$ indeed computes the function $F \restriction \rho$. The crux of the argument is to show that with high probability, $\mathcal{T}_1(F,\rho)$ is a small depth decision tree. We would like to bound $\mathrm{wt}(\mathcal{B})$ where $\mathcal{B}$ is the set of "bad" restrictions $\rho$ such that $\mathrm{depth}(\mathcal{T}_1(F,\rho)) \geq \ell$.

To do this, we use the encoding style proof of Razborov [2, 16]. We will define an encoding function $\mathcal{E} : \mathcal{B} \to \{0,1\}^X \times \mathcal{A}$ for a suitable finite "auxiliary" set $\mathcal{A}$ with the following properties.

- Decodability: The map $\mathcal{E}$ is 1-1. Or equivalently, given $\mathcal{E}(\rho) = (\mathcal{E}_1(\rho), \mathcal{E}_2(\rho)) \in \{0,1,*\}^X \times \mathcal{A}$ for any $\rho \in \mathcal{B}$, we can recover the restriction $\rho$.
- Weight increase: For each $\rho \in \mathcal{B}$, we have $\mathrm{wt}(\mathcal{E}_1(\rho)) \geq \mathrm{wt}(\rho) \cdot \Gamma$ for some $\Gamma \geq 1$.

If we have an encoding $\mathcal{E}$ as above, we can bound $\mathrm{wt}(\mathcal{B})$ as follows.

$$\begin{aligned}\mathrm{wt}(\mathcal{B}) &= \sum_{\rho\in\mathcal{B}} \mathrm{wt}(\rho) \leq \sum_{\rho\in\mathcal{B}} \frac{\mathrm{wt}(\mathcal{E}_1(\rho))}{\Gamma} = \frac{1}{\Gamma}\sum_{\rho\in\mathcal{B}} \mathrm{wt}(\mathcal{E}_1(\rho)) \\ &\leq \sum_{\rho'\in\{0,1,*\}^X} \frac{\mathrm{wt}(\rho')}{\Gamma}\cdot|\mathcal{A}| = \frac{|\mathcal{A}|}{\Gamma}\cdot\sum_{\rho'\in\{0,1,*\}^X}\mathrm{wt}(\rho') = \frac{|\mathcal{A}|}{\Gamma} \qquad (2)\end{aligned}$$

where for the first inequality we have used the Weight increase property of $\mathcal{E}_1$ and for the second inequality we have used the fact that $\mathcal{E}$ is 1-1 and hence for each $\rho' \in \{0,1,*\}^X$, the number of $\rho$ such that $\mathcal{E}_1(\rho) = \rho'$ is bounded by $|\mathcal{A}|$.

We are now ready to define the Encoding function $\mathcal{E}$ and show that it has the desired properties. We will take the set $\mathcal{A}$ to be $[k]^\ell \times \{0,1\}^\ell \times \{0,1\}^\ell$.

Fix any $\rho \in \mathcal{B}$, i.e. such that $\mathrm{depth}(\mathcal{T}_1(F,\rho)) \geq \ell$. We identify each path in $\mathcal{T}_1(F,\rho)$ by the tuple of Boolean answers obtained for the queries along the path. Let $\pi' \in \{0,1\}^t$ for $t \geq \ell$ be the lexicographically first path of length at least $\ell$ in $\mathcal{T}_1(F,\rho)$ and let $\pi \in \{0,1\}^\ell$ be the answers to the first $\ell$ queries along the path.

During the course of first $\ell$ queries, assume that the terms considered by $\mathcal{T}_1(F,\rho)$ are $T_1,\ldots,T_s$ (in order) and let $X_1,\ldots,X_s$ be the variables queried in these terms. Note that the sets $X_1,\ldots,X_s$ are pairwise disjoint.

Let $\tau_1,\ldots,\tau_s \in \{0,1,*\}^X$ be the restrictions that set the variables in $X_1,\ldots,X_s$ according to $\pi$ (i.e. according the answers to the queries made by $\mathcal{T}_1(F,\rho)$). We define the restrictions $\sigma_1,\ldots,\sigma_s \in \{0,1,*\}^X$ where $\sigma_i$ sets the variables in $X_i$ in the unique way so that the term $T_i$ is not forced to 0 by $\sigma_i$ (note that $\sigma_i$ might force $T_j$ to 0, which will not be relevant to us).

We set $\mathcal{E}_1(\rho) = \rho\sigma_1 \cdots \sigma_s$. At this point, we can calculate the amount of weight increase. Since $\mathcal{E}_1(\rho)$ has exactly $\ell$ fewer $*$s than $\rho$, by (1), we have

$$\frac{\mathrm{wt}(\mathcal{E}_1(\rho))}{\mathrm{wt}(\rho)} = \left(\frac{1-p}{2p}\right)^{\ell}. \tag{3}$$

We now need to specify the auxiliary data $\mathcal{E}_2(\rho)$, which is designed in a way that allows us to decode $\rho$ from $\mathcal{E}(\rho)$, i.e. to find the additional variables that are set by $\mathcal{E}_1(\rho)$ and "unset" them. To do this, we make the following important observations:

- Say $s > 1$. The term $T_1$ is the first term that is set to 1 in $F \restriction \mathcal{E}_1(\rho)$.
- More generally, for $i < s$, if we define the restriction $\rho^{(i)} = \rho\tau_1 \cdots \tau_{i-1}\sigma_i \cdots \sigma_s$, then $T_i$ is the first term that is set to 1 in $F \restriction \rho^{(i)}$.
- The term $T_s$ is the first term not set to 0 by $\rho^{(s)}$. [1]

By the above observation, given the restriction $\mathcal{E}_1(\rho) = \rho^{(1)}$, we can recover the term $T_1$. In this term, the set of variables $X_1$ can then be specified by specifying for each variable $x \in X_1$, a $j(x) \in [k]$ that gives the relative position of $x$ among the variables in $T_1$. This allows us to undo the settings to the variables in $X_1$.

Motivated by this, we set $\mathcal{E}_2(\rho) = (\mathbf{j}, \mathbf{b}, \pi)$ where

- $\mathbf{j} = (j(x_{i_1}), \ldots, j(x_{i_\ell})) \in [k]^\ell$ where $x_{i_1}, \ldots, x_{i_\ell}$ are the variables queried along the path $\pi$ (in that order) and $j(x_{i_r}) \in [k]$ is the index of the variable $x_{i_r}$ in the corresponding term ($T_i$ if $x_{i_r} \in X_i$).
- $\mathbf{b} = (b(x_{i_1}), \ldots, b(x_{i_\ell})) \in \{0,1\}^\ell$ that tells us for each $r \in [\ell]$ if $x_{i_r}$ is the last variable queried in the corresponding term (term $T_i$ if $x_{i_r} \in X_i$).
- $\pi \in \{0,1\}^s$ that allows us to recover $\tau_1, \ldots, \tau_s$.

To decode $\rho$ from $(\mathcal{E}_1(\rho), \mathcal{E}_2(\rho))$, we proceed as above. We start with $\rho^{(1)} = \mathcal{E}_1(\rho)$ and obtain term $T_1$. Having done so, we can find $X_1$ using the auxiliary data and also the restriction $\tau_1$ (the bit vector $\mathbf{b}$ tells us exactly how many variables are in $X_1$ and hence how many entries of $\mathbf{j}$ and $\pi$ are relevant for $X_1$). Now, we can change the settings to the variables in $X_1$ in accordance with $\tau_1$ to construct the restriction $\rho^{(2)}$. We continue in the same fashion until all the sets $X_1, \ldots, X_s$ are found. Setting all the variables in these sets to $*$, we can recover $\rho$. This concludes the description and analysis of the decoding procedure.

[1] The reason for the slight difference in the case of $T_s$ is because we clipped the long path $\pi'$ to have length exactly $s$.

Since we have an encoding $\mathcal{E}$ satisfying the decoding property, we can use (2) and (3) to bound the weight of $\mathcal{B}$ as follows.

$$\mathrm{wt}(\mathcal{B}) \leq \frac{|\mathcal{A}|}{\Gamma} = \frac{(4k)^\ell}{(1-p/2p)^\ell} \leq \left(\frac{8pk}{1-p}\right)^\ell \leq (10pk)^\ell$$

where for the final inequality we have used $p \leq 1/10$. This concludes the proof of the Håstad Switching Lemma. □

# 4 A blockwise Projection Switching Lemma

In this section, we describe a recent Projection Switching Lemma of Chen et al. [5] that was used to prove lower bounds for small-depth circuits testing small-distance connectivity in undirected graphs. We will also show an application of this lemma from [5] to prove depth-$d$ superpolynomial-size circuit lower bounds for a function that has a polynomial-sized circuit of depth $2d$.

While the statement of the Projection Switching lemma in [5] is quite specific to the setting of small-distance connectivity, the proof is more general. Here, we give a slightly more general statement that follows using the same techniques.

Assume that we have a set $X$ of $n$ variables that is partitioned into $m$ sets $X_1, \ldots, X_m$ of size $n/m$ each. We refer to the $X_1, \ldots, X_m$ as *blocks*. Let $Y = \{y_1, \ldots, y_m\}$ be a fresh set of variables. Let $p \in [0, 1]$ be a parameter.

We denote by $\mathcal{R}_2(p)$ the distribution over $\{0, 1, *\}^X$ output by the following sampling process.

1. For each $i \in [m]$, we choose a uniformly random assignment $a_i \in S$ where $S$ is some fixed multiset of Boolean assignments from $\{0, 1\}^{n/m}$.

2. For each $i \in [m]$, with probability $p$, we set *all* the variables in $X_i$ to $*$, and with probability $1 - p$, we set all the variables in $X_i$ according to the assignment $a_i$. We say that $X_i$ is a $*$-*block of* $\rho$ if all the variables in $X_i$ are set to $*$.

Note that the above restrictions are quite different from the restrictions $\mathcal{R}_1(p)$ defined in Section 3 since the assignments to distinct variables in the same block are highly correlated. This kind of correlation is necessary to ensure that the hard function for which we are trying to prove lower bounds does not simplify drastically after the restriction.

It is easy to come up with scenarios where restrictions of the above form do not yield good Switching lemmas. Nevertheless, as we show below, there are interesting situations where it is possible to use prove a switching lemma for random projections that utilize these restrictions.

Let $\mathcal{S}_2$ be the family of restrictions that can appear when we sample a restriction $\rho \sim \mathcal{R}_2(p)$; equivalently, $\mathcal{S}_2$ is the set of restrictions $\rho \in \{0,1,*\}^X$ such that the restriction of $\rho$ to each $X_i$ is either all $*$s or an assignment from $S$. Note that if $\rho \in \mathcal{S}_2$ has exactly $a$ many $*$-blocks, then

$$\mathrm{wt}(\rho) = \frac{p^a(1-p)^{m-a}}{|S|^{m-a}}. \tag{4}$$

To be able to prove a Projection Switching Lemma, we will need the distribution $\mathcal{R}_2(p)$ to satisfy some additional properties.

Let $k \in \mathbb{N}$ and $\gamma \geq 1$ be parameters. We say that the distribution $\mathcal{R}_2(p)$ as defined above is $(k,\gamma)$-*feasible* if the following conditions are satisfied.

- Fix any $X_i$ and any subset $X_i' \subseteq X_i$ of size at most $k$. Then for each $b \in \{0,1\}$, there is an assignment $\alpha \in S$ that sets all the variables in $X_i'$ to $b$.
- $p|S| \leq 1/\gamma$.

**Projections.** Finally, we will project the variables by setting, for each $*$-block $X_i$ of $\rho$, all the variables in $X_i$ to the fresh variable $y_i \in Y$. We will use crucially the simplifications in this step to prove the switching lemma.

With the above notation in place, we state the Projection Switching lemma. As before we state the Switching lemma only for $k$-DNFs, but it also holds for $k$-CNFs as in Remark 2.

**Lemma 3.** *Fix a parameter $p \leq 1/2$. Let $\mathcal{R}_2(p)$ as defined above be $(k,\gamma)$-feasible. Let $F$ be a $k$-DNF. Then we have for $\rho \sim \mathcal{R}_2(p)$*

$$\Pr_{\rho}[D(\mathrm{Proj}(F \restriction \rho)) \geq \ell] \leq \left(\frac{10k}{\gamma}\right)^{\ell}.$$

Before we prove Lemma 3, let us see a couple of examples that motivate why we might want to consider the above projections.

**Example 1.** *1. Consider the setting where $k = 2$ and $|X_i| = t \geq 2$. Let $S = \{0^t, 1^t\}$. Define $F$ to be the following $k$-DNF*

$$F = \bigvee_{i=1}^{m} x_i' \wedge \neg x_i''$$

*where $x_i', x_i'' \in X_i$ are distinct variables.*

*For a random restriction $\rho \sim \mathcal{R}_2(p)$ as defined above, each term of $F$ is set to $0$ with probability $1-p$ and left unchanged with probability $p$ (note*

*that there is no assignment in $S$ that can set any term to 1). Thus, with good probability, $F \restriction \rho$ is a k-DNF with $\ell \approx pm$ many disjoint terms. In this case, it is easy to check that $D(F \restriction \rho) \geq \ell$ (which is large if $p$ and $m$ are chosen suitably). This tells us that we cannot hope to upper bound the probability that $D(F \restriction \rho)$ is large.*

*On the other hand, the function* $\mathrm{Proj}(F \restriction \rho)$ *is the identically zero function since each term collapses to* 0 *upon applying the projection. Hence, we have* $D(\mathrm{Proj}(F \restriction \rho)) = 0$ *with probability* 1.

*This illustrates why projections are crucial to proving Lemma 3.*

2. *Consider the case when $X_1, \ldots, X_m$ all have size $k$ and let $S = \{0,1\}^k \setminus \{1^k\}$. Let $F$ be the k-DNF*

$$F = \bigvee_{i=1}^{m} \bigwedge_{x \in X_i} x.$$

*Applying a random restriction $\rho \sim \mathcal{R}_2(p)$ leaves $F$ with $\ell \approx pm$ of its terms with good probability (this is because, as in the previous example, the assignments from $S$ cannot set any term of $F$ to* 1*). Upon applying a projection, we see that even the simpler* $\mathrm{Proj}(F \restriction \rho)$ *is an OR of size $\ell$, and hence* $D(\mathrm{Proj}(F \restriction \rho)) \geq \ell$*. Thus, we cannot hope to prove a Projection Switching lemma in this setting. The problem here is that the set $S$ does not contain the all* 1*s assignment and hence the restriction is not $(k, \gamma)$-feasible.*

We now turn to the proof of Lemma 3 which, modulo minor modifications, is exactly as in [5].

*Proof.* The proof of Lemma 3 follows the basic outline of the proof of Lemma 1 above. We will define a decision tree strategy for computing $\mathrm{Proj}(F \restriction \rho)$ and upper bound that the probability that this strategy fails to produce a decision tree of small depth. To do this, we will define an encoding function on the set of bad restrictions (those for which the decision tree has large depth) that satisfies both the decoding and weight increase properties from the proof of Lemma 1.

We now define a simple decision tree for computing $\mathrm{Proj}(F \restriction \rho)$. Assume that all the terms of $F$ have been ordered and so have the variables inside each term.

$\underline{\mathcal{T}_2(F, \rho)}$:

1. Consider the first term $T$ in $F$ that is not yet set to 0 and such that $T$ can be set to 1 by further setting some $*$-blocks in $\rho$ *using assignments from* $S$.

   Consider a term $T'$ has not been set to 0 by the current restriction $\rho$ but at the same time cannot be set to 1 using assignments from $S$. We claim that this must be because there is a $*$-block $X_i$ and variables $x', x'' \in X_i$ such that the

literals $x'$ and $\neg x''$ both appear in the term $T$. This is because if all the (at most $k$) literals in $T'$ from the same $*$-block $X_i$ have the same sign, then the $(k, \gamma)$-feasibility of the space $\mathcal{R}_2(p)$ guarantees that there is an assignment in $S$ that satisfies all these literals. Pasting together these different assignments for each block $X_i$ with variables in $T'$, we can construct a satisfying assignment for $T'$ using assignments in $S$ only. Thus, any currently non-zero term $T'$ that cannot be set to 1 using assignments from $S$ must contain two variables $x', x''$ from the same $*$-block with different signs. Upon applying the projection, such a term $T'$ will be set to 0 and it is hence not relevant to computing $\text{Proj}(F \restriction \rho)$.

This justifies the following: if there is no term $T$ of the above form, then output 0.

2. For each block $X_i$ such that some unset $x \in X_i$ appears in $T$ (note that $X_i$ is necessarily a $*$-block of $\rho$ since otherwise all the variables of $X_i$ are set to constants), query the variable $y_i$.

3. Let $\tau \in \{0, 1, *\}^X$ be the restriction that sets, for each $y_i$ queried in the previous step, all the variables $x \in X_i$ to the value obtained on querying $y_i$. Let $\rho' = \rho\tau$. If this satisfies the term $T$, then output 1. Otherwise, replace $\rho$ with $\rho'$ and go back to Step 1.

It is easy to see that $\mathcal{T}_2(F, \rho)$ indeed computes the function $\text{Proj}(F, \rho)$. Let $\mathcal{B}$ be the set of restrictions $\rho \in \mathcal{S}_2$ such that $\text{depth}(\mathcal{T}_2(F, \rho)) \geq \ell$. We need to bound $\text{wt}(\mathcal{B})$.

As in Lemma 1, we define an encoding function $\mathcal{E} : \mathcal{B} \to \mathcal{S}_2 \times \mathcal{A}$ where $\mathcal{A} = [k]^\ell \times \{0, 1\}^\ell \times \{0, 1\}^\ell$. Our encoding function will be decodable as before: for any $\rho \in \mathcal{B}$, we will be able to recover $\rho$ from $\mathcal{E}(\rho) = (\mathcal{E}_1(\rho), \mathcal{E}_2(\rho))$. Further, $\mathcal{E}(\rho)$ will satisfy a weight increase property. Putting these two facts together, we bound $\text{wt}(\mathcal{B})$ exactly as in the proof of Lemma 1.

Fix a $\rho \in \mathcal{B}$. Let $\pi' \in \{0, 1\}^t$ ($t \geq \ell$) be the answers received along the lexicographically first path of length at least $\ell$ in $\mathcal{T}_2(\text{Proj}(F \restriction \rho))$. We denote by $\pi$ the initial segment of $\pi'$ of length exactly $\ell$. Let $T_1, \ldots, T_s$ be the terms encountered by the decision tree along this computational path and let $Y_1, \ldots, Y_s$ be the sets of variables (from $Y$) queried while scanning terms $T_1, \ldots, T_s$ respectively. [2] Recall that after a variable $y_j \in Y_i$ is queried, all the variables in the block $X_j$ are set to constants and hence the variable $y_j$ cannot be queried again. Thus, the sets $Y_1, \ldots, Y_s$ are pairwise disjoint.

[2] Note that the set $Y_s$ need not contain all the variables queried during the scan of term $T_s$ along the path $\pi'$, but rather only those that appear along the clipped path $\pi$.

For each $i \in [s]$, let $\pi_i \in \{0,1\}^{|Y_i|}$ be the values of the variables in $Y_i$ as seen along the path $\pi$. Let $\tau_i$ be the restriction applied to the variables in $\bigcup_{y_j \in Y_i} X_j$ in Step 3 of $\mathcal{T}_2(F,\rho)$ after scanning term $T_i$.

The encoding $\mathcal{E}$ is defined as follows.

- $\mathcal{E}_1(\rho)$: For each $i \in [s-1]$, before the term $T_i$ is scanned by the decision tree $\mathcal{T}_2(F,\rho)$, the restriction $\rho\tau_1 \cdots \tau_{i-1}$ has already been applied to the variables. Let $\sigma_i$ denote any restriction that sets, for each $y_j \in Y_i$, all the variables in the block $X_j$ using an assignment $\alpha_j \in S$ so that the restriction $\rho\tau_1 \cdots \tau_{i-1}\sigma_i$ sets the term $T_i$ to 1 (there is such a $\sigma_i$ because this is exactly how the term $T_i$ is chosen in Step 1 of the description of $\mathcal{T}_2(F,\rho)$).

  For the case when $i = s$, we choose $\sigma_s$ in exactly the same way except that we cannot ensure that $\sigma_s$ sets $T_s$ to 1 since there may be unset variables in $T_s$ even after all the $*$-blocks corresponding to the variables in $Y_s$ are assigned. Instead, we can choose $\sigma_s$ so that $\rho\tau_1 \cdots \tau_{s-1}\sigma_s$ does not set the term $T_s$ to 0.

  We define $\mathcal{E}_1(\rho) = \rho\sigma_1 \cdots \sigma_s$.

- $\mathcal{E}_2(\rho)$: This is almost exactly the same as in Lemma 1. For each variable $y_r \in Y_i$, we fix a $j(y_r) \in [k]$ that indexes the first variable from the block $X_r$ that appears in the term $T_i$; we also fix a $b(y_r) \in \{0,1\}$ that tells us if $y_r$ was the last variable queried while scanning the term $T_i$ or not.

  Let $y_{i_1}, \ldots, y_{i_\ell}$ be the variables read by the $\mathcal{T}_2(F,\rho)$ along the path $\pi$ in that order. The auxiliary data $\mathcal{E}_2(\rho)$ is defined to be $(\mathbf{j}, \mathbf{b}, \pi)$ where $\mathbf{j} = (j(y_{i_1}), \ldots, j(y_{i_\ell}))$ and $\mathbf{b} = (b(y_{i_1}), \ldots, b(y_{i_\ell}))$.

We need to show that $\mathcal{E}(\rho)$ has both the decodability and the weight increase properties. Then, we will be done using (2).

We first describe how to recover $\rho$ given access to $\mathcal{E}(\rho)$. The decoding procedure inductively finds $Y_1, \ldots, Y_s$ and $\tau_1, \ldots, \tau_s$ as follows.

- Assume we already have $Y_1, \ldots, Y_{i-1}$ and $\tau_1, \ldots, \tau_{i-1}$. By setting the variables in the blocks corresponding to $y \in \bigcup_{j<i} Y_j$ according to the restriction $\tau_1 \cdots \tau_{i-1}$ and the remaining variables as per $\mathcal{E}_1(\rho)$, we obtain the restriction $\rho^{(i)} = \rho\tau_1 \cdots \tau_{i-1}\sigma_i \cdots \sigma_s$.

- We apply the restriction $\rho^{(i)}$ to $F$ and find the first term $T$ that is either set to 1 or that can be set to 1 by setting some $*$-blocks using assignments from $S$ (the latter condition is useful when $i = s$). This is the term $T_i$.

- Using the auxiliary data and $T_i$, we can recover both $\pi_i$ and $Y_i$ (exactly as in Lemma 1), and hence $\tau_i$.

By the end of this procedure, we have the sets $Y_1, \ldots, Y_s$. The restriction $\mathcal{E}_1(\rho)$ is obtained by setting the corresponding $*$-blocks in $\rho$ to assignments in $S$. Hence, to recover $\rho$, we simply change these blocks back to $*$-blocks. We have thus shown that the encoding $\mathcal{E}$ is decodable.

We now show that $\mathrm{wt}(\mathcal{E}_1(\rho))$ is considerably larger than $\mathrm{wt}(\rho)$. Since $\mathcal{E}_1(\rho)$ has $|\bigcup_{i\in[s]} Y_i| = \ell$ fewer $*$-blocks than $\rho$, we see from (4) that

$$\frac{\mathrm{wt}(\mathcal{E}_1(\rho))}{\mathrm{wt}(\rho)} = \left(\frac{1-p}{p|S|}\right)^{\ell} \geq (\gamma(1-p))^{\ell}$$

where the last inequality uses the fact that the random restriction $\mathcal{R}_2(p)$ is $(k,\gamma)$-feasible and hence $p|S| \leq 1/\gamma$. Thus, we have as in (2) that

$$\mathrm{wt}(\mathcal{B}) \leq |\mathcal{A}| \cdot \left(\frac{1}{\gamma(1-p)}\right)^{\ell} = (4k)^{\ell} \cdot \left(\frac{1}{\gamma(1-p)}\right)^{\ell} \leq \left(\frac{10k}{\gamma}\right)^{\ell}.$$

□

## 4.1 Using the Projection Switching Lemma for lower bounds

In this section, we give an application of Lemma 3 from [5]. We will show that there exist functions that have polynomial (and indeed linear) sized formulas of depth $2d$ but do not have subexponential-sized depth-$d$ circuits. While this is slightly weaker than the Depth-hierarchy theorem of Håstad [7] that was mentioned in the Introduction, the functions used here imply strong lower bounds for the problem of testing small-distance connectivity in a graph, strengthening an earlier result of Beame, Impagliazzo and Pitassi [3]. We refer the reader to [5] for further details regarding this.

We now define formally the functions for which we will prove lower bounds. Let $w, u$ be growing integer parameters with $u \leq w^{1/4}$. Define the function $\mathsf{SkewSipser}_d$ as follows inductively as follows. For $d = 1$, the function $\mathsf{SkewSipser}_1$ is defined on a set of $n_1 := u^2$ variables $\{x_{i,j} \mid i \in [u], j \in [u]\}$ as follows:

$$\mathsf{SkewSipser}_1(x_{1,1}, \ldots, x_{u,u}) = \bigwedge_{i\in[u]} \bigvee_{j\in[u]} x_{i,j}.$$

For $d \geq 2$, we define $\mathsf{SkewSipser}_d$ on $n_d := n_{d-1} \cdot wu$ variables by partitioning the input set $X$ of $n_d$ variables into $w \cdot u$ sets $X_{i,j}$ ($i \in [u], j \in [w]$) of size $n_{d-1}$ each. We then define

$$\mathsf{SkewSipser}_d(X) = \bigwedge_{i\in[u]} \bigvee_{j\in[w]} \mathsf{SkewSipser}_{d-1}(X_{i,j}).$$

Note that the function $\mathsf{SkewSipser}_d$ can be computed by a depth-$(2d)$ formula of size $O(n_d)$ where $n_d = w^{d-1} \cdot u^{d+1}$. However, we will see that any depth-$d$ circuit for $\mathsf{SkewSipser}_d$ must have large size.

**Theorem 4.** *Let $d$ be a constant. Let $C$ be any depth-$d$ circuit computing $\mathsf{SkewSipser}_d$. Then, $C$ must have size $w^{\Omega(u)}$.*

*Proof.* We will prove the theorem by applying Lemma 3 for a suitably chosen random projection so that the following happens.

- The circuit simplifies: Given any circuit $C$ of size $s \leq w^{\Omega(u)}$, the projection causes the depth of $C$ to drop by 1 with high probability. This will be a simple application of the projection switching lemma and the fact that decision trees of height $k$ can be written as both $k$-DNFs and $k$-CNFs.

- $\mathsf{SkewSipser}_d$ retains structure: At the same time, however, we also ensure that with high probability, the function $\mathsf{SkewSipser}_d$ does not simplify too much. In fact, we will ensure that even after the projection, the function $\mathsf{SkewSipser}_d$ "contains" a copy of $\mathsf{SkewSipser}_{d-1}$. This will allow us to induct.

Let $k = u - 1$.

We actually prove the following stronger statement. Let $C$ be any circuit computing $\mathsf{SkewSipser}_d$ of depth $d + 1$ where all the gates at depth $d$ (i.e. just above the variables) have fan-in at most $k$. Then $C$ contains at least $w^{\Omega(u)}$ gates of depth at most $d - 1$.

Note that the above generalizes the statement we want to prove, since we can always convert a depth-$d$ circuit with no bound on its fan-in to a depth-$(d + 1)$ circuit with bottom fan-in 1 by introducing dummy AND and OR gates of fan-in 1 just above the variables. This modification does not change the number of gates at depth at most $d - 1$.

The proof is by induction on the depth $d$. The base case is $d = 1$, which is trivial since the function $\mathsf{SkewSipser}_1$ cannot be expressed as $k$-CNF or a $k$-DNF (the proof of this is left to the reader).

Now, consider the case when $d \geq 2$. Define $m = n_d/u^2$ and $n = n_d$. Let $F_1, \ldots, F_m$ be the $m$ copies of $\mathsf{SkewSipser}_1$ at depth $2d - 2$ in the formula for $\mathsf{SkewSipser}_d$, and let $X_1, \ldots, X_m$ respectively be the sets of variables that appear in these formulas. The sets $X_1, \ldots, X_m$ all have the same size $u^2$ and partition the set $X$. We can further partition each $X_i$ into $X_{i,j}$ ($j \in [u]$) so that for each $i \in [s]$ we have

$$F_i = \bigwedge_{j \in [u]} \bigvee_{x \in X_{i,j}} x.$$

For $j \in [u]$ and $b \in \{0,1\}$, let $\alpha_{j,b} \in \{0,1\}^{u^2}$ be the assignment that sets all the variables in $X_{i,j}$ to $b$ and all the variables in $X_{i,j'}$ $(j \neq j')$ to $1-b$. Define $S = \{\alpha_{j,b} \mid j \in [u], b \in \{0,1\}\}$ to be the set of all such assignments. We have $|S| = 2u$. The set $S$ of assignments is chosen to satisfy the following properties:

- No assignment in $S$ sets any $F_i$ to 1.
- Every set of at most $k$ variables in $F_i$ can be simultaneously set to 0 or to 1 by some assignment in $S$.
- $|S|$ is not too large.

Let $C$ be any circuit of depth-$(d+1)$ and bottom fan-in at most $k$ computing $\mathsf{SkewSipser}_d$. Let $F'_1, \ldots, F'_t$ be the depth-2 subcircuits of $C$, which are either $k$-CNFs or $k$-DNFs.

We apply the projection switching lemma (Lemma 3) with the following family of random projections. We will use the family of restrictions $\mathcal{R}_2(p)$ with the blocks being $X_1, \ldots, X_m, S$ being the assignments to these blocks as defined above and $p = \frac{1}{w^{2/3}}$. After the restriction is applied, the $*$-blocks are projected down to a fresh set $Y = \{y_1, \ldots, y_m\}$ of variables. It can be checked that $\mathcal{R}_2(p)$ is $(k, \gamma)$-feasible for $k$ as above and $\gamma = w^{1/3}$ for $w$ large enough.

Let $\rho \sim \mathcal{R}_2(p)$. We analyze $\mathrm{Proj}(F \restriction \rho)$ for $F \in \{F'_1, \ldots, F'_t, F_1, \ldots, F_m\}$.

- Each $F \in \{F'_1, \ldots, F'_t\}$ is a $k$-CNF or a $k$-DNF. By Lemma 3, we see that

$$\Pr_{\rho}[D(\mathrm{Proj}(F \restriction \rho)) \geq u] \leq (10u/\gamma)^u \leq w^{-u/20}$$

  for large enough $w$. If the number of depth-2 subcircuits $t \geq w^{u/20}/10$, then we have our lower bound. Otherwise, by a union bound we see that with probability at least $9/10$, each $F'_i$ satisfies $D(\mathrm{Proj}(F \restriction \rho)) \leq u - 1 = k$. If this is the case, each $F''_i = \mathrm{Proj}(F'_i \restriction \rho)$ can be written as both a $k$-CNF or a $k$-DNF. Choosing the CNF or DNF representation for each $F''_i$ appropriately and merging with the gates that it feeds into, we can reduce the depth of the projected circuit $\mathrm{Proj}(C \restriction \rho)$ to $d$ while keeping the bottom fan-in bounded by $k$.

  Note that this last step might increase the number of gates in the circuit overall, but does not change the number of gates at depths $d-2$ or less. So the induction hypothesis for depth $d$ is now applicable.

- Now consider $F_i$ for some $i \in [t]$. Each $X_i$ is a $*$-block with probability $p$, in which case the entire formula $F_i$ collapses to a fresh variable $y_i$, and with probability $1-p$ all the variables in $F_i$ are set according to an assignment

in $S$, which causes the formula $F_i$ to collapse to 0 (note that no assignment in $S$ sets $F_i$ to 1).

Let $G$ be any OR-gate at depth $2d-3$ in $\mathsf{SkewSipser}_d$. The OR gate has $w$ many inputs, each of which collapses to a distinct variable from $Y$ with probability $p$ and to 0 with probability $1-p$. Thus the function $\mathrm{Proj}(G \restriction \rho)$ is an OR of expected fan-in $pw = w^{1/3}$. By a standard Chernoff bound, the probability that the $\mathrm{Proj}(G \restriction \rho)$ has fan-in less than $u$ (which in turn is smaller than $w^{1/3}/2$ for large enough $w$) is $\exp(-\Omega(w^{1/3}))$. Taking a union bound over all the OR-gates at depth $2d-3$ (there are $w^{O(d)}$ of them), we see that with probability at least $9/10$, each OR-gate has fan-in at least $u$ after the projection.

By the above reasoning, with probability at least $8/10$ over the choice of the random projection, we have both that the depth of $C' = \mathrm{Proj}(C \restriction \rho)$ can be efficiently reduced to $d$, and that $\mathsf{SkewSipser}'_d := \mathrm{Proj}(\mathsf{SkewSipser}_d \restriction \rho)$ becomes a depth $2d-2$ formula over the variables $Y$ where each OR gate at depth $2d-3$ has fan-in at least $u$.

The latter formula is almost the formula $\mathsf{SkewSipser}_{d-1}$ with the only difference being that the gates at depth $2d-3$ in $\mathsf{SkewSipser}_{d-1}$ have fan-in exactly $u$. However, this can be easily fixed by setting some of the variables in $Y$ to 0 to ensure that each gate at depth $2d-3$ in $\mathsf{SkewSipser}'_d$ has fan-in exactly $u$.

We can now apply the induction hypothesis to show that $C'$ has at least $w^{\Omega(u)}$ many gates at depth $d-2$ or less. This implies the same lower bound for $C$ and hence completes the induction. □

# 5 A Projection Switching lemma for average-case hardness

In this section, we present a variant of the Projection Switching lemma due to Rossman et al. [14], which was used to prove an *average-case Depth Hierarchy theorem* for small-depth circuits. We begin by describing the statement of the result of [14].

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and $\mathcal{C}$ a class of circuits over $n$ Boolean variables. We say that $f$ is *$\varepsilon$-correlated* with $\mathcal{C}$ if for any $C \in \mathcal{C}$ and a uniformly random input $x \in \{0,1\}^n$ we have

$$\Pr_x[f(x) = C(x)] \leq \frac{1}{2} + \varepsilon.$$

Note that any Boolean function $f$ agrees with either the constant 0 or the constant 1 function on at least half its inputs; so, the property of being $\varepsilon$-correlated (for

some small $\varepsilon$) with $C$ ensures that no circuit $C \in C$ can achieve much better agreement with $f$ than the trivial agreement achieved by one of the constant functions. For $\varepsilon = o(1)$ for example, this is a strong notion of average-case hardness for Boolean functions.

Using the classical Switching Lemma (Lemma 1), Håstad [7] showed that the Parity function is $o(1)$-correlated with the class of constant-depth circuits of subexponential size (more refined versions of this statement were proved recently by Impagliazzo, Matthews and Paturi [10] and Håstad [8]). However, the following question remained open for a long time: do there exist functions that have constant-depth $d$ circuits of polynomial-size and are $o(1)$-correlated with any depth-$(d-1)$ circuits of polynomial size?

This question was recently resolved positively by Rossman, Servedio and Tan [14] using the technique of random projections. They were, in fact, able to resolve the question for depths $d \leq \frac{\varepsilon\sqrt{\log n}}{\log\log n}$ for some absolute constant $\varepsilon > 0$. More recently, Håstad [9] strengthened this theorem to give such a result for all depths up to $\Omega(\frac{\log n}{\log\log n})$. We state this theorem below.

**Theorem 5.** *The following holds for some absolute constant $\delta > 0$. Let $n$ be a growing parameter and assume $2 \leq d \leq \frac{\delta \log n}{\log\log n}$. There is an explicit Boolean function $f_d$ that is computed by depth-$d$ formulas of linear size such that $f_d$ is $n^{-\Omega(1/d)}$-correlated with any circuit of depth at most $d-1$ and size at most $2^{n^{1/5(d-1)}}$.*

We do not attempt to prove this theorem here. However, we outline some ideas of the proof in order to motivate the projection switching lemma proved in the next section.

The functions $f_d$ in Theorem 5 are similar to the SkewSipser functions from Theorem 4, with the difference being that the fan-ins of all the gates are chosen carefully to ensure that on a uniformly random input, the output of the function $f_d$ is a near-uniformly random bit.

At a very high level, the proof of Theorem 5 proceeds along similar lines to Theorem 4. As in the proof of Theorem 4, we cannot apply the standard Switching Lemma (Lemma 1) since those random restrictions also collapse the function $f_d$ to constant with high probability. Instead, we would like to set some of the input variables in a correlated way to preserve the structure of $f_d$. At the same time, however, we would like to ensure that the input bits are set independently at random (since we are measuring correlation under the uniform distribution). These two requirements, the need for correlation on the one hand and independence on the other, are seemingly at odds with each other. However, it turns out that there is a way of reconciling them with the use of random projections. The basic idea is illustrated by the following observation of O'Donnell and Wimmer [12].

**The O'Donnell-Wimmer trick.** Let $p \in [0,1]$ be a parameter and suppose we want to generate an input from the product distribution $\mu_p$ over $\{0,1\}^w$ where each bit is set to 1 independently with probability $1 - p$.[3] It can be checked that the following sampling process generates a random $\alpha \in \{0,1\}^w$ with distribution $\mu_p$.

- Sample a random element $\alpha' \in \{*,1\}^w \setminus \{1^w\}$ by setting each entry of $\alpha'$ to be independently 1 with probability $1-p$ *subject to the constraint* that there is at least one $*$ in $\alpha'$. I.e., we sample a sequence $\alpha^{(1)}, \alpha^{(2)}, \ldots,$ from $\{*,1\}^p$ where each entry of $\alpha^{(i)}$ is independently set to 1 with probability $1-p$ and $*$ otherwise. We then pick the first $\alpha^{(j)}$ that is not $1^w$ and call it $\alpha'$.
- Choose $z$ to be 1 with probability $p' = (1-p)^w$ and 0 with probability $1-p'$. Let $\alpha \in \{0,1\}^w$ be the string obtained from $\alpha'$ by replacing all the $*$-locations by $z$.

(Note that the first step of the above process is essentially a random projection to a single variable.)

Interestingly, while each step of the above process sets the variables in a correlated fashion, the composition of the two steps results in an uncorrelated distribution across the $w$ coordinates. This allows to view the process of sampling an input from, say, the uniform distribution in a series of steps where the input variables are set in a correlated way.

The proofs of [14, 9] are based on designing a sequence of carefully chosen correlated random projection steps, similar to but more involved than the O'Donnell-Wimmer trick above, such that each step preserves the structure of the hard function $f_d$ and moreover, the projections compose to yield the uniform distribution over all the inputs.

To prove the lower bound, we also need to show that each random projection step helps simplify a sub-exponential sized circuit of smaller depth. For this, we need a projection switching lemma for this family of random projections. We describe such a projection switching lemma in the next section.

## 5.1 The Projection Switching lemma of [14]

We describe a simple version of a random projection due to Rossman et al. [14] and prove a projection switching lemma for this random projection. While the random projections used to prove the average-case depth hierarchy theorems in [14] are much more sophisticated, most of the additional technicality is introduced to preserve the structure of the hard functions; the proof of the Switching lemma is not very different from what is presented here.

[3] While we are interested in correlation over the uniform distribution, analogous questions for biased product distributions naturally appear at intermediate stages of the proof.

Let $p, q, \lambda \in [0,1]$ and $w \in \mathbb{N}$ be parameters. Define

$$p' = (1-p)^w \quad \text{and} \quad \gamma = \min\left\{\frac{1-\lambda-q}{q}, \frac{\lambda(1-p')}{p'q}\right\}.$$

Let $X$ be a set of $n$ variables partitioned into $m$ parts $X_1, \ldots, X_m$, each of size $w$. We will call the sets $X_1, \ldots, X_m$ *blocks*. Let $Y = \{y_1, \ldots, y_m\}$ be a set of fresh variables.

We first define a *local random restriction* that only acts on a single block. This restriction is a probability distribution $\mathcal{R}'_3(p,q,\lambda)$ over $\{0,1,*\}^w$ defined as follows:

- Choose a restriction $\xi'$ from $\{*,1\}^w \setminus \{1^w\}$ such that each entry is set independently to 1 with probability $p$ and $*$ otherwise, subject to the constraint that there is at least one $*$ (exactly as in the O'Donnell-Wimmer trick above).
- Choose a $z \in \{0,1,*\}$ such that $z = 1$ with probability $\lambda$, $*$ with probability $q$ and 0 with probability $1-\lambda-q$.
- Output $\xi \in \{0,1,*\}^w$ obtained by replacing all the $*$s in $\xi'$ with $z$.

In informal terms, the above process can be visualized as first applying the O'Donnell-Wimmer trick to project the $w$ random bits to a single random bit and applying a random restriction to this bit.

Note that the process of sampling a local restriction never produces a restriction containing both 0s and $*$s. For $z \in \{0,*\}$, we say that $\xi$ is of type $z$, denoted $\text{type}(\xi) = z$, if $\xi \in \{1,z\}^* \setminus \{1^w\}$. If $\xi = 1^w$, we define $\text{type}(\xi) = 1$.

We define the weight $\text{wt}'(\xi)$ of a local random restriction to be the probability that it is output under the distribution $\mathcal{R}'_3(p,q,\lambda)$. Note that

$$\text{wt}'(\xi) = \begin{cases} \frac{p^{w-a}(1-p)^a q}{1-p'} & \text{if } \text{type}(\xi) = * \text{ and } \xi \text{ has } a \text{ 1s}, \\ \frac{p^{w-a}(1-p)^a(1-\lambda-q)}{1-p'} & \text{if } \text{type}(\xi) = 0 \text{ and } \xi \text{ has } a \text{ 1s}, \\ \lambda & \text{if } \text{type}(\xi) = 1. \end{cases} \tag{5}$$

To sample a random restriction from $\mathcal{R}_3(p,q,\lambda)$ over the set of variables $X$, we sample a $\xi_i \in \{0,1,*\}^{X_i}$ according to the distribution $\mathcal{R}'_3(p,q,\lambda)$ and output $\rho = \xi_1 \cdots \xi_m$ (where we think of each local restriction $\xi_i$ as leaving variables outside $X_i$ unset). Clearly, we have $\text{wt}(\rho) = \prod_{i \in [m]} \text{wt}'(\xi_i)$.

For $\rho$ sampled from $\mathcal{R}_3(p,q,\lambda)$ as above and $i \in [m]$, we say that $X_i$ is a *$z$-block of $\rho$* if $\text{type}(\xi_i) = z$.

Finally, for every $X_i$ that is a $*$-block of $\rho$, we identify all the unset variables in $X_i$ with the variable $y_i$. This yields the random projection that we will analyze.

We will prove the following Projection Switching lemma due to Rossman et al. [14].

**Lemma 6.** *Assume $p \leq 1/2$. Say $F$ is a k-DNF and $\rho \sim \mathcal{R}_3(p, q, \lambda)$. Then,*

$$\Pr_{\rho}[D(\mathrm{Proj}(F \restriction \rho)) \geq \ell] \leq \left(\frac{4ke^{2pk}}{\gamma}\right)^{\ell}.$$

*Proof.* We start by defining the decision tree for $\mathrm{Proj}(F \restriction \rho)$ whose depth we will analyze.

$\underline{\mathcal{T}_3(F, \rho)}$:

1. Consider the first term $T$ in $F$ that is not yet set to 0 by $\rho$. If there is no such term, output 0.

2. For each block $X_i$ of $\rho$ such that some unset $x \in X_i$ appears in $T$ (note that $X_i$ is necessarily a $*$-block of $\rho$ since otherwise all the variables of $X_i$ are set to constants), query the variable $y_i$.

3. Let $\tau \in \{0, 1, *\}^X$ be the restriction that sets, for each $y_i$ queried in the previous step, all the variables $x \in X_i \cap \rho^{-1}(*)$ to the value obtained on querying $y_i$. Let $\rho' = \rho\tau$. If this restriction satisfies the term $T$ then output 1. Otherwise, replace $\rho$ with $\rho'$ and go back to Step 1.

Let $\mathcal{B}$ be the set of restrictions $\rho \in \{0, 1, *\}^X$ such that $\mathrm{depth}(\mathcal{T}_3(F, \rho)) \geq \ell$. We need to bound $\mathrm{wt}(\mathcal{B})$.

As in Lemmas 1 and 3, we define an encoding function $\mathcal{E} : \mathcal{B} \to \mathcal{S}_2 \times \mathcal{A}$ that is both decodable and satisfies a suitable weight increase property. We will use $\mathcal{A} = [k]^{\ell} \times \{0, 1\}^{\ell} \times \{0, 1\}^{\ell} \times \{0, 1\}^{\ell k}$.

Fix a $\rho \in \mathcal{B}$. Let $\pi' \in \{0, 1\}^t$ ($t \geq \ell$) be the answers received along the lexicographically first path of length at least $\ell$ in $\mathcal{T}_3(\mathrm{Proj}(F \restriction \rho))$. We denote by $\pi$ the initial segment of $\pi'$ of length exactly $\ell$. Let $T_1, \ldots, T_s$ be the terms encountered by the decision tree along the path $\pi$ and let $Y_1, \ldots, Y_s$ be the sets of variables queried while scanning terms $T_1, \ldots, T_s$ respectively. Note that the sets $Y_1, \ldots, Y_s$ are pairwise disjoint.

For each $i \in [s]$, let $\pi_i \in \{0, 1\}^{|Y_i|}$ be the values obtained for the variables in $Y_i$ along the path $\pi$. Define $X'_i = \bigcup_{y_j \in Y_i} X_j \cap \rho^{-1}(*)$ and let $\tau_i$ be the restriction applied to the variables in $X'_i$ in Step 3 of $\mathcal{T}_3(F, \rho)$ after scanning the term $T_i$.

The encoding $\mathcal{E}$ is defined as follows.

- $\mathcal{E}_1(\rho)$: For each $i \in [s-1]$, before the term $T_i$ is scanned by the decision tree $\mathcal{T}_2(F, \rho)$, the restriction $\rho\tau_1 \cdots \tau_{i-1}$ has already been applied to the variables. Let $\sigma_i$ denote the restriction that sets all the variables in the set $X'_i$ as follows:

  - Each $x \in X'_i$ that appears positively in $T_i$ is set to 1,

– All other $x \in X'_i$ are set to 0 (i.e. variables that either appear negatively in $T_i$ or don't appear at all in $T_i$ are set to 0).[4]

We define $\mathcal{E}_1(\rho) = \rho\sigma_1 \cdots \sigma_s$.

- $\mathcal{E}_2(\rho)$: For each variable $y_r \in Y_i$, we fix a $j(y_r) \in [k]$ that indexes the first variable from the block $X_r$ that appears in the term $T_i$, and we also fix a $b(y_r) \in \{0,1\}$ that tells us if $y_r$ was the last variable read while scanning the term $T_i$ or not.

  Finally, for each $i \in [s]$, we let $\eta_i \in \{0,1\}^k$ be the bit string such that $\eta_i(j) = 1$ if and only if the $j$th variable in the term $T_i$ is left unset by $\rho$ but set to 1 by $\mathcal{E}_1(\rho)$. We define $\eta = \eta_1 \cdots \eta_s \cdot 0^{k(\ell-s)} \in \{0,1\}^{\ell k}$.

  Let $y_{i_1}, \ldots, y_{i_\ell}$ (in that order) be the variables read by $\mathcal{T}_2(F, \rho)$ along the path $\pi$. The auxiliary data is defined to be

$$\mathcal{E}_2(\rho) = (\mathbf{j}, \mathbf{b}, \pi, \eta)$$

  where $\mathbf{j} = (j(y_{i_1}), \ldots, j(y_{i_\ell}))$ and $\mathbf{b} = (b(y_{i_1}), \ldots, b(y_{i_\ell}))$.

We now see how to recover $\rho$ given access to $\mathcal{E}(\rho)$. The decoding procedure will recover for each $i \in [s]$ the following.

- the sets $Y_1, \ldots, Y_s$,
- for each $y_j \in \bigcup_{i \in [s]} Y_i$, the set $X_j \cap \rho^{-1}(*)$,
- the restrictions $\tau_1, \ldots, \tau_s$.

This is done as follows.

- Assume we already have $Y_1, \ldots, Y_{i-1}$, $\tau_1, \ldots, \tau_{i-1}$, and for each $y_j \in Y_1 \cup \cdots Y_{i-1}$, we have the set $X_j \cap \rho^{-1}(*)$. By setting the variables in $\bigcup_{j<i} X_j \cap \rho^{-1}(*)$ according to the restriction $\tau_1 \cdots \tau_{i-1}$ and other variables according to $\mathcal{E}_1(\rho)$, we obtain the restriction $\rho^{(i)} = \rho\tau_1 \cdots \tau_{i-1}\sigma_i \cdots \sigma_s$.
- We apply the restriction $\rho^{(i)}$ to $F$ and find the first term $T$ that is not set to 0. This is the term $T_i$.
- From $T_i$ and the auxiliary data in $\mathbf{j}, \mathbf{b}$, and $\pi$, we obtain the set $Y_i$ and $\pi_i$. This is exactly as in Lemma 1.

  For each $y_j \in Y_i$, we can recover $X_j \cap \rho^{-1}(*)$ as follows. Since some variable of $X_j$ was unset in $F \restriction \rho$, $X_j$ must be a $*$-block of $\rho$. Hence, it follows $\rho(x)$

[4] It is crucial here that all the variables in $X'_i$ (including the ones that don't appear in $T_i$) are set to constants to ensure that the weight increase property holds)

is either $*$ or 1 for each $x \in X_j$. Thus, we see that any variable from $X_j$ that is set to 0 in $\rho^{(i)}$ must lie in $\rho^{-1}(*)$. It only remains to find the variables of $X_j \cap \rho^{-1}(*)$ that are set to 1 by $\rho^{(i)}$. These variables can be obtained from $\eta_i$. Hence, we have $X_j \cap \rho^{-1}(*)$.

Finally, having found $X_j \cap \rho^{-1}(*)$ for each $y_j \in Y_i$, we can set $\tau_i$ to be the restriction that sets all the variables in $\bigcup_{y_j \in Y_i}(X_j \cap \rho^{-1}(*))$ according to the values seen along $\pi_i$.

By the end of this procedure, we have the sets $Y_1, \ldots, Y_s$ and further for each $y_j \in \bigcup_i Y_i$ the set $X_j \cap \rho^{-1}(*)$ of variables from the block $X_j$ that were unset in $\rho$ but have been set to constants in $\mathcal{E}_1(\rho)$. To recover $\rho$ from $\mathcal{E}_1(\rho)$, we simply unset (i.e. change to $*$) all these variables. This concludes the proof of the fact that $\mathcal{E}$ is decodable.

We now bound $\mathrm{wt}(\mathcal{B})$. The argument here will be a little more indirect than in Lemmas 1 and 3. For any string $\alpha$ over the alphabet $\{0, 1, *\}$, we use $|\alpha|$ to denote the number of 1s in $\alpha$. Observe that for any $\rho \in \mathcal{B}$, we have $|\mathcal{E}_1(\rho)| = |\rho| + |\eta|$ where $\eta$ is the last component of $\mathcal{E}_2(\rho)$ as defined above. As $\eta \in \{0,1\}^{\ell k}$, we have

$$|\rho| \leq |\mathcal{E}_1(\rho)| \leq |\rho| + \ell k.$$

For $c \in \{0, \ldots, \ell k\}$, we define $\mathcal{B}_c = \{\rho \mid |\mathcal{E}_1(\rho)| - |\rho| = c\}$. The encoding function $\mathcal{E}$ restricts to a function $\mathcal{E}^c : \mathcal{B}_c \to \{0,1,*\}^X \times \mathcal{A}_c$ where

$$\mathcal{A}_c = [k]^\ell \times \{0,1\}^\ell \times \{0,1\}^\ell \times \{\eta \in \{0,1\}^{\ell k} \mid |\eta| = c\}.$$

We claim the following weight increase property of $\mathcal{E}^c$ for each $c \in \{0, \ldots, \ell k\}$.

**Claim 7.** *For each $\rho \in \mathcal{B}_c$, we have*

$$\frac{\mathrm{wt}(\mathcal{E}_1(\rho))}{\mathrm{wt}(\rho)} \geq \gamma^\ell \cdot \left(\frac{1-p}{p}\right)^c.$$

Assuming the above claim for now, we finish bounding $\mathrm{wt}(\mathcal{B})$ as follows. By (2) applied to each $\mathcal{E}^c$, we get that for $c \in \{0, \ldots, \ell k\}$

$$\mathrm{wt}(\mathcal{B}_c) \leq |\mathcal{A}_c| \cdot \frac{1}{\gamma^\ell}\left(\frac{p}{1-p}\right)^c = \left(\frac{4k}{\gamma}\right)^\ell \cdot \binom{\ell k}{c} \cdot \left(\frac{p}{1-p}\right)^c.$$

Since $\mathcal{B}$ is the disjoint union of $\mathcal{B}_0, \ldots, \mathcal{B}_{\ell k}$ we get

$$\begin{aligned}
\mathrm{wt}(\mathcal{B}) &= \sum_{c=0}^{\ell k} \mathrm{wt}(\mathcal{B}_c) \\
&\leq \left(\frac{4k}{\gamma}\right)^{\ell} \cdot \sum_{c=0}^{\ell k} \binom{\ell k}{c} \cdot \left(\frac{p}{1-p}\right)^{c} \\
&= \left(\frac{4k}{\gamma}\right)^{\ell} \cdot \left(1 + \frac{p}{1-p}\right)^{\ell k} \\
&\leq \left(\frac{4k}{\gamma}\right)^{\ell} \cdot e^{p\ell k/1-p} \leq \left(\frac{4k}{\gamma}\right)^{\ell} \cdot e^{2p\ell k} \\
&= \left(\frac{4ke^{2pk}}{\gamma}\right)^{\ell}.
\end{aligned}$$

This yields the statement of the lemma modulo the proof of Claim 7, which appears below.

*Proof of Claim 7.* Fix any restriction $\rho \in \mathcal{B}_c$. We can write $\rho = \xi_1 \cdots \xi_m$ for local restrictions $\xi_i$ on block $X_i$ and similarly, we have $\mathcal{E}_1(\rho) = \xi'_1 \cdots \xi'_m$. Let $J \subseteq [m]$ index the first $\ell$ variables in $Y$ that are read along the lexicographically first path of length at least $\ell$ in $\mathcal{T}_2(F, \rho)$.

By the definition of $\mathcal{E}_1(\rho)$, $\xi_j \neq \xi'_j$ if and only if $j \in J$. Thus we have

$$\frac{\mathrm{wt}(\mathcal{E}_1(\rho))}{\mathrm{wt}(\rho)} = \prod_{j \in [m]} \frac{\mathrm{wt}'(\xi'_j)}{\mathrm{wt}'(\xi_j)} = \prod_{j \in J} \frac{\mathrm{wt}'(\xi'_j)}{\mathrm{wt}'(\xi_j)}.$$

(Recall that $\mathrm{wt}'(\xi)$ denotes the weight of a local restriction $\xi$.)

It thus suffices to prove that for each $j \in J$, we have

$$\frac{\mathrm{wt}'(\xi'_j)}{\mathrm{wt}'(\xi_j)} \geq \gamma \cdot \left(\frac{1-p}{p}\right)^{c_j} \tag{6}$$

where $c_j = |\xi'_j| - |\xi_j|$.

This can be argued from (5) and the definition of $\gamma$, as follows.

We know that $\mathrm{type}(\xi_j) = *$ whereas $\mathrm{type}(\xi'_j)$ is either 0 or 1 since all the variables in $X_j$ are set to constants by $\mathcal{E}_1(\rho)$. In the case that $\mathrm{type}(\xi'_j) = 1$, we have $|\xi'_j| = w$, which implies that $|\xi_j| = w - c_j$. Substituting in (5), we see that

$$\frac{\mathrm{wt}'(\xi'_j)}{\mathrm{wt}'(\xi_j)} = \frac{\lambda(1-p')}{p'q} \cdot \left(\frac{1-p}{p}\right)^{c_j} \geq \gamma \cdot \left(\frac{1-p}{p}\right)^{c_j}.$$

In the case that $\text{type}(\xi'_j) = *$, a similar computation gives

$$\frac{\text{wt}'(\xi'_j)}{\text{wt}'(\xi_j)} = \frac{1-\lambda-q}{q} \cdot \left(\frac{1-p}{p}\right)^{c_j} \geq \gamma \cdot \left(\frac{1-p}{p}\right)^{c_j}.$$

This proves (6) and completes the proof of the claim. □

□

# 6 Acknowledgements

I would like to thank V Arvind for inviting me to write this article. I would also like to thank Li-Yang Tan for sending me his slides on the results of [14] and Igor Oliveira for helpful discussions on Switching lemmas and Projection Switching lemmas.

# References

[1] M. Ajtai. $\Sigma^1_1$-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

[2] Paul Beame. A switching lemma primer. Technical report, Department of Computer Science and Engineering, University of Washington, April 1994.

[3] Paul Beame, Russell Impagliazzo, and Toniann Pitassi. Improved depth lower bounds for small distance connectivity. *Computational Complexity*, 7(4):325–345, 1998.

[4] Ravi B. Boppana and Michael Sipser. The complexity of finite functions. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, pages 757–804. 1990.

[5] Xi Chen, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 612–625, 2016.

[6] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

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

[8] Johan Håstad. On the correlation of parity and small-depth circuits. *SIAM J. Comput.*, 43(5):1699–1708, 2014.

[9] Johan Håstad. An average-case depth hierarchy theorem for higher depth. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 79–88, 2016.

[10] Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for $AC^0$. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 961–972, 2012.

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

[12] Ryan O'Donnell and Karl Wimmer. Approximation by DNF: examples and counterexamples. In *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, pages 195–206, 2007.

[13] Toniann Pitassi, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Poly-logarithmic frege depth lower bounds via an expander switching lemma. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 644–657, 2016.

[14] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1030–1048, 2015.

[15] Michael Sipser. Borel sets and circuit complexity. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 61–69, 1983.

[16] Neil Thapen. Notes on switching lemmas. *Unpublished manuscript*, 2009.

[17] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10, 1985.