Electronic Colloquium on Computational Complexity, Report No. 24 (2021)

A Majority Lemma for Randomised Query Complexity

Mika Göös
EPFL

Gilbert Maystre
EPFL

February 14, 2021

Abstract. We show that computing the majority of n copies of a boolean function g has randomised query complexity $\mathrm{R}(\textsc{Maj} \circ g^n) = \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g))$. In fact, we show that to obtain a similar result for any composed function $f \circ g^n$, it suffices to prove a sufficiently strong form of the result only in the special case $g = \textsc{GapOr}$.

1 Introduction

In boolean function complexity theory, a typical *direct sum problem* asks: For a given boolean function $g\colon \{0,1\}^m \to \{0,1\}$, how much harder is it to compute g on n separate inputs, that is, computing $g^n(x^1,\dots,x^n) \coloneqq (g(x^1),\dots,g(x^n))$, compared to computing g on a single input? For randomised query complexity, a complete answer was recently obtained by Blais and Brody [BB19] (improving on [JKS10, BK18]). They showed that the most obvious way to compute g^n is optimal: Evaluate each copy of g separately with a "reduced" error probability $\ll 1/n$ so that, by a union bound, the n-bit output will be correct with high probability. More precisely, their result states

$$\forall g: \quad \mathrm{R}(g^n) \;=\; \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g)). \qquad \text{(Direct sum)}$$

Here we used standard notation: $\mathrm{R}(g) \coloneqq \mathrm{R}_{1/3}(g)$ where $\mathrm{R}_\epsilon(g)$ denotes the ϵ-error query complexity of g, that is, the least number of queries a randomised algorithm (decision tree) must make to the input bits $x_i \in \{0,1\}$ of an unknown input $x \in \{0,1\}^m$ in order to output $g(x)$ with probability at least $1-\epsilon$ (where the probability is over the internal randomness of the algorithm). Similarly, $\overline{\mathrm{R}}_\epsilon(g)$ denotes the ϵ-error *expected* query complexity of g where we measure the expected (rather than worst-case) number of queries made by the algorithm. See Section 2 for precise definitions.

How far can we push the direct sum result? What if, instead of all the n output bits of g^n, we only wanted to compute their parity? In other words, what is the randomised query complexity of the composed function $\textsc{Xor} \circ g^n$? Do we still have to compute each g with reduced error? Brody et al. [BKLS20] provided an affirmative answer:

$$\forall g: \quad \mathrm{R}(\textsc{Xor} \circ g^n) \;=\; \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g)). \qquad \text{(}\textsc{Xor}\text{ Lemma)}$$

More generally, we can ask the following question.

Problem 1. *For which n-bit outer functions f (assume $\mathrm{R}(f) = \Theta(n)$ for simplicity) and inner functions g does the composed function $f \circ g^n$ necessitate error reduction?*

There is no conjectured characterisation for when error reduction is necessary. To showcase the subtlety of this question, we mention that $f = \textsc{Or}$, despite having a highly "sensitive" input $x = 0^n$, never necessitates error reduction. By now, there are many proofs [FRPU94, KK94, New09, GS10, BGKW20] showing that $\mathrm{R}(\textsc{Or} \circ g^n) = O(n \cdot \mathrm{R}(g))$ for every g.

Our goal in this paper is to make further progress on Problem 1.

ISSN 1433-8092

1.1 Our results

Our main result is to prove tight bounds for composing with the n-bit majority function Maj. This in particular confirms a conjecture made in [BB19, BGKW20].

Theorem 1 (Maj lemma)**.** $\mathrm{R}(\textsc{Maj} \circ g^n) = \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g))$ *for every partial function* g*.*

Previously, Ben-David et al. [BGKW20] proved Theorem 1 in the special case $g = \textsc{GapOr}$. Here $\textsc{GapOr} = \textsc{GapOr}_m$ is the m-bit partial function defined by $\textsc{GapOr}(x) = \textsc{Or}(x)$ on inputs of Hamming weight $|x| \in \{0, m/2\}$ and is undefined otherwise. This is a particularly clean example of a function whose query complexity behaves as (assuming $m \geq \log(1/\epsilon)$)

$$\overline{\mathrm{R}}_\epsilon(\textsc{GapOr}) \;=\; \Theta(\log(1/\epsilon)).$$

We prove Theorem 1 by a direct *reduction* to this previous result! Our more general result says, informally, that error reduction is necessary for any composed function $f \circ g^n$ if it is necessary in the special case $g = \textsc{GapOr}$. Our key conceptual insight is to formulate a sense in which every g can be "simulated" by GapOr. There is, however, a slight technical caveat. For the reduction to work, we need to assume that the lower bound for $f \circ \textsc{GapOr}^n$ holds not only against randomised decision trees but also against a more powerful model called ϵ*-approximate nonnegative degree* $\deg^+_\epsilon$ (aka conical junta degree, partition bound), which we will recall in Section 2.

Theorem 2 (Reduction to GapOr)**.** *Suppose that a function* f *satisfies* $\deg^+_\epsilon(f \circ h^n) \geq \Omega(n \log n)$ *for some constant* $\epsilon > 0$ *and for both* $h \in \{\textsc{GapOr}_{\log n}, \neg\textsc{GapOr}_{\log n}\}$*. Then*

$$\forall g: \quad \mathrm{R}(f \circ g^n) \;=\; \Omega(n \cdot \overline{\mathrm{R}}_{1/n}(g)).$$

Theorem 1 follows immediately by combining Theorem 2 with [BGKW20, Theorem 4], which proved the required nonnegative degree lower bound for $\textsc{Maj} \circ \textsc{GapOr}^n$ (we only note that their proof works equally well for $\neg$GapOr in place of GapOr). In fact, the nonnegative degree lower bound holds more generally for any $(2n+1)$-bit outer function that agrees with Maj on inputs of weight n and $n+1$. For example, Xor is such a function, and hence the Xor lemma of Brody et al. [BKLS20] can be recovered using Theorem 2. However, the original proof in [BKLS20] is much simpler than ours, and moreover, the result of [BKLS20] actually characterises $\overline{\mathrm{R}}_\epsilon(\textsc{Xor} \circ g^n)$ for all $\epsilon > 0$ while we focus on the bounded-error case $\epsilon = 1/3$.

Our goal for the rest of the paper is to prove Theorem 2.

Optimality? We note that our choice of GapOr in Theorem 2 is optimal at least in the sense that it cannot be replaced with the more symmetric alternative GapMaj, which is defined by $\textsc{GapMaj}_m(x) = \textsc{Maj}_m(x)$ on inputs of weight $|x| \in \{m/3, 2m/3\}$ and undefined otherwise. There are known examples of *partial* f (but no known *total* ones) for which GapOr does not need error reduction while GapMaj does [BGKW20, Section 4]. We suspect however that other aspects of Theorem 2 can be improved; see Section 1.4 for open problems.

1.2 Techniques: Leaf Lemma

Our main technical contribution, which might be of independent interest, is what we call Leaf Lemma. It states that every boolean function g admits a balanced input distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$, where μ^i is a distribution supported on $g^{-1}(i)$, and a "hard side" $b \in \{0,1\}$ satisfying the following: If we run a decision tree of shallow depth $\ll \overline{\mathrm{R}}_\epsilon(g)$ on a random input $x \sim \mu$ then we will typically reach a leaf ℓ making *one-sided error*, that is, if the leaf ℓ is reached by $x \sim \mu^b$ with probability p, then ℓ is also reached by $x \sim \mu^{1-b}$ with probability at least $\epsilon \cdot p$. Interestingly, this property is inherently one-sided and the choice of the hard side b depends on the function g. For example, GapOr and ¬GapOr have distinct hard sides. See our proof overview in Section 3 for more details.

1.3 Other related work

Complexity of composition. A major theme in boolean function complexity theory is to understand the complexity of the composition $f \circ g^n$ in terms of the complexities of its two constituent functions. It has been long known that many well-studied complexity measures behave *multiplicatively* under composition. For example, deterministic query complexity satisfies $\mathrm{D}(f \circ g^n) = \mathrm{D}(f)\,\mathrm{D}(g)$ [Sav02], quantum query complexity satisfies $\mathrm{Q}(f \circ g^n) = \Theta(\mathrm{Q}(f)\,\mathrm{Q}(g))$ [Rei11, LMR+11], and yet more examples (degree, certificate complexity, sensitivity) are discussed in [Tal13]. An interesting exception to this rule is randomised query complexity, where we can have two types of counter-examples.

- *Super-multiplicative:* There are functions f and g such that $\mathrm{R}(f \circ g^n) \geq \omega(\mathrm{R}(f)\,\mathrm{R}(g))$. For example, this happens whenever f necessitates error reduction for $g = $ GapOr.
- *Sub-multiplicative:* Recent work [GLSS19, BB20a] has found surprising examples of *partial* f and g such that $\mathrm{R}(f \circ g^n) \leq o(\mathrm{R}(f)\,\mathrm{R}(g))$.

It is still open to quantify the extent to which multiplicativity can fail. For example, it has not been ruled out that $\mathrm{R}(f \circ g^n) \geq \mathrm{R}(f)\,\mathrm{R}(g)/\mathrm{poly}(\log n)$ for all partial functions. It is also possible that a strict multiplicative lower bound holds for all *total* functions. This latter question is known as the *randomised composition conjecture* (for total functions) and it has been studied in a long line of work [BK18, AGJ+17, GLSS19, BDG+20, BB20a, BB20b].

Noisy decision trees. Necessity of error reduction is closely related to the model of "noisy decision trees" [FRPU94, EP98, DR08, GS10]. In this model, the goal is to compute a boolean function f given *noisy query access* to its input bits. A single query to an input variable x_i returns its correct value with probability $2/3$ (say) and the opposite value $1 - x_i$ with probability $1/3$. This model is effectively equivalent to computing $f \circ \textsc{GapMaj}^n$ in the standard query model. With this interpretation, one of the results of [FRPU94] states that $\mathrm{R}(\textsc{Maj} \circ \textsc{GapMaj}^n) = \Theta(n \log n)$. We note that this is weaker (in two respects) than the result $\deg^+_\epsilon(\textsc{Maj} \circ \textsc{GapOr}^n) = \Theta(n \log n)$ from [BGKW20], which we used to derive our main result (although see Problem 2 below).

1.4 Open problems

How optimal is Theorem 2? We suspect that our assumption about nonnegative degree is an artifact of our proof and can be relaxed as follows.

Problem 2. *Show that the hypothesis in Theorem 2 can be weakened to* $\mathrm{R}(f \circ h^n) \geq \Omega(n \log n)$.

Whether we need to assume hardness for both GapOr and its negation, we do not know.

Problem 3. *Are there examples of f with* $\mathrm{R}(f \circ \textsc{GapOr}^n) \geq \omega(\mathrm{R}(f \circ \neg\textsc{GapOr}^n))$*?*

Theorem 2 could be useful in showing tight composition results for yet more outer functions. For example, consider the well-studied partial function $\textsc{SqrtGapMaj}_n$ (often called simply *the* gap majority function) defined as $\textsc{Maj}_n$ but restricted to inputs of Hamming weight $|x| \notin n/2 \pm \sqrt{n}$.

Problem 4. *Show* $\mathrm{R}(\textsc{SqrtGapMaj} \circ g^n) = \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g))$ *for every* g.

2 Query complexity basics

We study *partial* boolean functions $f\colon \{0,1\}^n \to \{0,1,*\}$. The *domain* of the function is $\mathrm{dom}(f) \coloneqq f^{-1}(\{0,1\})$ and the inputs $f^{-1}(*)$ are *undefined*. We say f is *total* if $\mathrm{dom}(f) = \{0,1\}^n$. For partial functions f and g, their *composition* $f \circ g^n$ is defined by $(f \circ g^n)(x^1, \dots, x^n) \coloneqq f(g(x^1), \dots, g(x^n))$ if $x^i \in \mathrm{dom}(g)$ for all $i \in [n]$; otherwise $(f \circ g^n)(x^1, \dots, x^n) \coloneqq *$. Standard references for boolean function complexity are [BdW02, Juk12].

Decision trees. A *(deterministic) decision tree* t is an algorithm for computing a boolean function on an unknown input $x \in \{0,1\}^n$. The algorithm repeatedly queries the input variables $x_i \in \{0,1\}$ in some order (which can depend on outcomes of queries made so far) until eventually producing an output $t(x)$. Such an algorithm can be represented as a binary tree, with internal nodes labelled with variables x_i, outgoing edges of the internal nodes labelled with query outcomes ($x_i = 0$ and $x_i = 1$), and leaves labelled with output values. Each input x determines a unique root-to-leaf path, obtained by following the query outcomes consistent with x. The most important cost measure of t is its *depth*, denoted $\mathrm{depth}(t)$, which is the longest root-to-leaf path in the tree and equals $\max_x q(t,x)$ where $q(t,x)$ denotes the number of queries made by t on input x.

A *randomised decision tree* T is a distribution over deterministic decision trees $t \sim T$. We say T computes $f\colon \{0,1\}^n \to \{0,1,*\}$ with error ϵ if for every $x \in \mathrm{dom}(f)$ we have $\mathbb{P}_{t\sim T}[t(x) = f(x)] \geq 1-\epsilon$. There are two cost measures for T: the *(worst-case) depth* is the maximum depth of any decision tree in the support of T; the *expected depth* is $\max_x \mathbb{E}_{t\sim T}[q(t,x)]$. The *$\epsilon$-error query complexity of f*, denoted $\mathrm{R}_\epsilon(f)$, is the least depth of a randomised decision tree that computes f with error ϵ. The *ϵ-error expected query complexity*, denoted $\overline{\mathrm{R}}_\epsilon(f)$, is defined analogously.

Error reduction. It is well known that the error probability of an algorithm (computing a boolean-valued function) can be reduced from any constant $1/2 - \delta$, where $\delta > 0$, to any other constant $\epsilon > 0$ by repeating the algorithm constantly many times (in fact, $O(\log(1/\epsilon)/\delta^2)$ many) and outputting the majority answer. Hence we often set $\epsilon \coloneqq 1/3$ and omit ϵ from notation. In this bounded-error regime, we have $\overline{\mathrm{R}}(f) \leq \mathrm{R}(f) \leq O(\overline{\mathrm{R}}(f))$ where the second inequality follows by truncating executions that query many more bits than the expectation. For vanishing $\epsilon = o(1)$ (as $n \to \infty$), it is possible that $\overline{\mathrm{R}}_\epsilon(f) \leq o(\mathrm{R}_\epsilon(f))$. For example, consider the partial $2n$-bit function f where the task is to distinguish inputs of the form $x0^n$ from inputs of the form $0^n x$ with the promise that $|x| = n/2$. We have $\overline{\mathrm{R}}_{1/n}(f) = O(1)$ while $\mathrm{R}_{1/n}(f) = \Theta(\log n)$. In this small-error regime, the following fine-grained error reduction calculation will be useful.

Claim 3. *$\overline{\mathrm{R}}_{\epsilon^k}(f) \leq 4k \cdot \overline{\mathrm{R}}_\epsilon(f)$ for every $k \geq 1$ and $\epsilon \leq 1/16$.*

Proof. Suppose T computes f with error ϵ and consider the algorithm T' that runs T $4k-1$ times and outputs the majority answer. Then T' errs iff at least $2k$ of the runs err. This happens with probability at most $\sum_{i=2k}^{4k-1} \binom{4k-1}{i} \epsilon^i (1-\epsilon)^{4k-1-i} \leq 2^{4k}\epsilon^{2k} \leq \epsilon^k$. □

Leaf indicators. Let t be a decision tree with n-bit inputs. We denote by $\mathcal{L}(t)$ the set of its leaves and by $\ell_x^t \in \mathcal{L}(t)$ the unique leaf reached on input x. We often identify a leaf $\ell \in \mathcal{L}(t)$ with its associated *leaf indicator* function $\ell\colon \{0,1\}^n \to \{0,1\}$ defined by $\ell(x) \coloneqq 1$ iff input x reaches leaf ℓ. Thus each ℓ is simply a conjunction of at most $\mathrm{depth}(t)$ literals (x_i or $\bar{x}_i$) determined by the unique root-to-ℓ path in t. If t outputs boolean values, we let $\mathcal{A}(t) \subseteq \mathcal{L}(t)$ denote the set of *accepting* leaves, that is, those that output 1. Since the leaf indicators have pairwise disjoint supports, we can write the function computed by t as

$$t(x) \;=\; \textstyle\sum_{\ell\in\mathcal{A}(t)} \ell(x). \tag{1}$$

Nonnegative degree. Let $p\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$ be a nonnegative function. We say p is a *nonnegative d-junta* if it depends on at most d of its variables. For example, if t is a depth-d decision tree, then each $\ell \in \mathcal{L}(t)$ is a nonnegative d-junta. More generally, we say that p is a *conical junta* of degree d if it can be written as a conical combination of nonnegative d-juntas, that is, $p(x) = \sum_i a_i q_i(x)$ where $a_i \geq 0$ are nonnegative scalars and the q_i are nonnegative d-juntas. For example, the function computed by t is a degree-d conical junta, as given by the expression (1). The *nonnegative degree* of p, denoted $\deg^+(p)$, is the least d such that p is a degree-d conical junta.

Let $f\colon \{0,1\}^n \to \{0,1,*\}$ be a partial function. We say that p ϵ-approximates f if $p(x) \in f(x) \pm \epsilon$ for every $x \in \mathrm{dom}(f)$. The *ϵ-approximate nonnegative degree* of f, denoted $\deg^+_\epsilon(f)$, is the least degree of a conical junta that ϵ-approximates f. For example, if T is a depth-d randomised ϵ-error decision tree for f, then there exists a degree-d conical junta p_T that ϵ-approximates f, namely,

$$p_T(x) \;\coloneqq\; \mathbb{E}_{t\sim T}[t(x)] \;\in\; f(x) \pm \epsilon.$$

This shows that $\deg^+_\epsilon(f) \leq \mathrm{R}_\epsilon(f)$. The gap betweeen $\deg^+_{1/3}(f)$ and $\mathrm{R}(f)$ can be huge for partial functions. For example, consider the n-bit UniqueOr defined by $\textsc{UniqueOr}(x) = \textsc{Or}(x)$ for inputs of weight $|x| \in \{0,1\}$ and undefined othwerwise. Then $\deg^+(\textsc{UniqueOr}) = 1$ (computed by $\sum_i x_i$) while $\mathrm{R}(\textsc{UniqueOr}) = \Theta(n)$. For total functions, the gap is at most polynomial [BdW02].

Nonnegative degree has been studied under many names: (one-sided) partition bound [JK10], WAPP query complexity [GLM+16, BGKW20], and query complexity "in expectation" [KLdW15].

3 Proof overview

Here we outline the proof of Theorem 2. We phrase the proof in the contrapositive: Supposing that T is a randomised decision tree computing $f \circ g^n$ of shallow depth $\ll n \cdot \overline{\mathrm{R}}_{1/n}(g)$ we construct an approximate conical junta for $f \circ \textsc{GapOr}^n$ (or $f \circ \neg\textsc{GapOr}^n$) of degree $\ll n \log n$.

Our overview is in two parts.

(§3.1) We first formulate our main technical lemma called Leaf Lemma and its generalisation Multileaf Lemma. They describe what typical leaves of T look like: they are *noisy*, meaning that they make noticeable errors in predicting the outputs of many copies of g. The proofs of these lemmas will occupy the remaining sections of this paper.

(§3.2) Then we use Multileaf Lemma to prove Theorem 2. A notable component of this part of the proof is showing how the acceptance probabilities of noisy leaves can be "simulated" by low-degree conical juntas in the domain of $f \circ \textsc{GapOr}^n$.

3.1 Statement of Leaf Lemma

Example. We build up to the statement of Leaf Lemma by first considering the prototypical example $g = \textsc{GapOr}_m$. Define two distributions μ^0 and μ^1 so that μ^i is uniform over $\textsc{GapOr}_m^{-1}(i)$. Namely, μ^0 places probability 1 on the input 0^m and μ^1 is uniform over x of weight $|x| = m/2$. Suppose t is a deterministic decision tree of shallow depth $d \ll m$ trying to compute $\textsc{GapOr}_m$. For a leaf $\ell \in \mathcal{L}(t)$ and any input distribution μ we write for short

$$\ell(\mu) \;\coloneqq\; \mathbb{E}_{x\sim\mu}[\ell(x)] \;=\; \mathbb{P}_{x\sim\mu}[\ell(x) = 1].$$

What do the typical leaves look like when we run t on a random input $x \sim \mu^i$ for $i \in \{0,1\}$?

- *Easy side $i = 1$.* The tree will query a 1-bit after about 2 queries in expectation. Such leaves ℓ are safe to output 1 as they know $\textsc{GapOr}(x) = 1$ for certain: $\ell(\mu^0) = 0$ and $\ell(\mu^1) > 0$.

- *Hard side* $i = 0$. Here every query returns 0 and we reach a leaf ℓ reading d many 0s. Although the leaf ℓ can be quite confident that the input x was sampled from μ^0 rather than μ^1, some uncertainty remains: $\ell(\mu^0) = 1$ and $\ell(\mu^1) \geq \epsilon$ for $\epsilon := 2^{-\Omega(d)}$.

In both cases, we have $\ell(\mu^1) \geq \epsilon \cdot \ell(\mu^0)$ and we say that ℓ is *(one-sidedly) noisy.* We now formalise how every g gives rise to such noisy leaves.

General case. Fix a partial function $g\colon \{0,1\}^m \to \{0,1,*\}$. Let $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ be a balanced distribution where μ^i is supported on $g^{-1}(i)$. For a leaf ℓ over m bits, a "hard side" $b \in \{0,1\}$, and an error parameter $\epsilon \geq 0$, we define

$$\ell \text{ is } (\epsilon, \mu, b)\text{-}\textit{noisy} \quad \overset{\text{def}}{\iff} \quad \ell(\mu^{1-b}) \;\geq\; \epsilon \cdot \ell(\mu^b).$$

Our Leaf Lemma says that every partial function g admits a hard distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ such that if we run a shallow decision tree t on a random input $x \sim \mu$, the leaf reached ℓ^t_x will typically be noisy. For simplicity of notation, for small quantities $a, b \in [0,1]$, we write $a \ll b$ (resp. $a \lll b$) to mean $a \leq cb$ (resp. $a^c \leq b$) for a sufficiently small constant $c > 0$.

Leaf Lemma. *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over $\mathrm{dom}(g)$ and a hard side $b \in \{0,1\}$ such that for every deterministic tree t and $i \in \{0,1\}$:*

$$\frac{\mathbb{E}_{x\sim\mu^i}[q(t,x)]}{\overline{\mathrm{R}}_\epsilon(g)} \lll \delta \qquad \Longrightarrow \qquad \mathbb{P}_{x\sim\mu^i}[\, \ell^t_x \text{ is } (\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Leaf Lemma is our main technical contribution. The proof appears in Section 4. To whet the reader's appetite, we highlight two interesting challenges that make the lemma non-trivial.

(C1) *Which side is hard?* We need to somehow tease out a hard side for an arbitrary g and this can even depend on the choice of μ. For example, consider $g(b,x) := b \oplus \textsc{GapOr}(x)$ where $b \in \{0,1\}$. Rather than μ assigning b at random, the distribution can fix b to either 0 or 1, which reduces g to either $\textsc{GapOr}$ or $\neg\textsc{GapOr}$ (two functions with distinct hard sides).

(C2) *Behaviour of typical leaves.* The existence of μ is often proved using various minimax theorems (we use one due to Blais and Brody [BB19]). These theorems typically guarantee that any shallow decision tree incurs error at least ϵ on average relative to μ. This does not rule out the following bad scenario: the tree could make error $1/2$ on 2ϵ fraction of the leaves reached and no error on $1-2\epsilon$ fraction of the leaves—here the typical leaves are not noisy!

In order to use Leaf Lemma in the context of composed functions, we generalise it to the direct sum setting where the inputs come from $\mathrm{dom}(g^n) := \mathrm{dom}(g)^n$. Let ℓ be a leaf over nm bits and write $\ell(x) = \prod_{i\in[n]} \ell_i(x^i)$ where $x^i \in \{0,1\}^m$ and each ℓ_i is over m bits. We define

$$\ell \text{ is } (\delta,\epsilon, \mu, b)\text{-}\textit{noisy} \quad \overset{\text{def}}{\iff} \quad \ell_i \text{ is } (\epsilon,\mu,b)\text{-noisy for at least } (1-\delta)n \text{ many } i \in [n].$$

Our generalised lemma says that we will typically reach a noisy leaf if we run a shallow decision tree on a random input from the product distribution $\mu^y := \mu^{y_1} \times \cdots \times \mu^{y_n}$ where $y \in \{0,1\}^n$.

Multileaf Lemma. *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over $\mathrm{dom}(g)$ and a hard side $b \in \{0,1\}$ such that for every deterministic tree t taking inputs from $\mathrm{dom}(g^n)$ and having $\mathrm{depth}(t)/(n\overline{\mathrm{R}}_\epsilon(g)) \lll \delta$,*

$$\forall y \in \{0,1\}^n : \qquad \mathbb{P}_{x\sim\mu^y}[\, \ell^t_x \text{ is } (\delta,\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Given Leaf Lemma the proof of the generalisation is not difficult: we can use linearity of expectation to see that the expected number of queries t makes to most copies of g is low, and hence we can apply Leaf Lemma for those copies. The details appear in Section 5.

3.2 Proof of Theorem 2

We conclude this overview section with a proof of Theorem 2 using Multileaf Lemma. We start with a lemma that shows how the noisy leaves in the domain of g^n can be "simulated" by low-degree conical juntas in the domain of $\textsc{GapOr}^n$. For simplicity, we state the lemma assuming a hard side $b = 0$; an analogous lemma holds for $b = 1$ by replacing $\textsc{GapOr}$ with $\neg\textsc{GapOr}$.

Simulation Lemma. *Let ℓ be a $(\delta, \epsilon, \mu, 0)$-noisy leaf over the variables of g^n. There exists a conical junta $p_\ell\colon (\{0,1\}^{\log n})^n \to \mathbb{R}_{\geq 0}$ of degree at most $n \cdot [\delta \log n + \log(1/\epsilon)]$ such that*

$$\forall x \in \mathrm{dom}(\textsc{GapOr}^n_{\log n}): \qquad p_\ell(x) \;=\; \ell(\mu^{\textsc{GapOr}^n_{\log n}(x)}).$$

Proof. We start by defining three conical juntas in the domain of $\textsc{GapOr}_m$ for $m := \log n$. Let $\mathcal{S}^m_k$ be the distribution over multisets obtained by picking k random elements from $[m]$ with replacement.

$$\begin{aligned} q_1(x) &:= \tfrac{2}{m}\textstyle\sum_{i\in[m]} x_i && \text{of degree } 1,\\ q_2(x) &:= \textstyle\prod_{i\in[m]} \bar{x}_i && \text{of degree } m = \log n,\\ q_3(x) &:= \mathbb{E}_{S\sim\mathcal{S}^m_k} \textstyle\prod_{i\in S} \bar{x}_i && \text{of degree } k := \log(1/\epsilon). \end{aligned}$$

Note the following output values:

$$\begin{aligned} &\forall x \in (\textsc{GapOr}_m)^{-1}(0): && q_1(x) = 0, && q_2(x) = 1, && q_3(x) = 1,\\ &\forall x \in (\textsc{GapOr}_m)^{-1}(1): && q_1(x) = 1, && q_2(x) = 0, && q_3(x) = 2^{-k} = \epsilon. \end{aligned}$$

Let $y = (y^1, \ldots, y^n)$ be the input variables of g^n. We write $\ell(y) = \prod_i \ell_i(y^i)$ so that $\ell(\mu^{\textsc{GapOr}^n_m(x)}) = \prod_i \ell_i(\mu^{\textsc{GapOr}_m(x^i)})$. We simulate each factor in this product separately. For $i \in [n]$ consider the function $p_i\colon \{0,1\}^m \to \mathbb{R}_{\geq 0}$ defined by

$$p_i(x) \;:=\; \ell_i(\mu^{\textsc{GapOr}_m(x)}).$$

First note that p_i can always be written as a conical combination of q_1 and q_2 in degree $\log n$. Moreover, if ℓ_i is $(\epsilon, \mu, 0)$-noisy, meaning $\ell_i(\mu^1) \geq \epsilon \cdot \ell_i(\mu^0)$, then we can do better and write p_i as a conical combination of q_1 and q_3 in degree $\log(1/\epsilon)$. We now define $p_\ell := \prod_i p_i$. The claimed bound on the degree of p_ℓ follows because at most δ fraction of the ℓ_i are non-noisy. ☐

We are now ready to prove Theorem 2 using Multileaf Lemma and Simulation Lemma.

Proof of Theorem 2. Suppose for contradiction that T is a randomised decision tree for $f \circ g^n$ having error $1/3$ and depth $\gamma n \overline{\mathrm{R}}_{1/n}(g)$ where $\gamma = o(1)$ as $n \to \infty$. Our goal is to construct an $o(n \log n)$-degree $o(1)$-approximate conical junta for $f \circ \textsc{GapOr}^n_{\log n}$ (or $f \circ \neg\textsc{GapOr}^n_{\log n}$).

We make two simplifying assumptions wlog.

1. The randomised tree T has error $o(1)$. To ensure this, we may reduce T's error by running it $1/\sqrt{\gamma} = \omega(1)$ times. This will yield an $o(1)$-error tree of depth $\sqrt{\gamma} n \overline{\mathrm{R}}_{1/n}(g) = o(n\overline{\mathrm{R}}_{1/n}(g))$.
2. There is some $\epsilon := 1/n^{o(1)}$ such that T has depth $o(n\overline{\mathrm{R}}_\epsilon(g))$. To ensure this, we may apply Claim 3 to see that $\gamma n \overline{\mathrm{R}}_{1/n}(f) \leq \sqrt{\gamma} n \overline{\mathrm{R}}_\epsilon(f) \leq o(n\overline{\mathrm{R}}_\epsilon(f))$ where $\epsilon := 1/n^{4\sqrt{\gamma}}$.

We invoke Multileaf Lemma with the above $\epsilon \leq o(1)$ and $\delta := \max\{\gamma^c, \epsilon^c\} \leq o(1)$ for small enough constant $c > 0$. We get a hard distribution μ and a hard side b, say $b = 0$ (case $b = 1$ is similar, but using $\neg\textsc{GapOr}$), such that the following holds: For every t in the support of T if we run t on a random input $x \sim \mu^y$, where $y \in \{0,1\}^n$, then the leaf reached ℓ^t_x will be $(\delta, \epsilon, \mu, 0)$-noisy with probability $1 - o(1)$. This allows us to effectively ignore non-noisy leaves: denoting by $\mathcal{N}(t) \subseteq \mathcal{A}(t)$ the set of accepting leaves that are $(\delta, \epsilon, \mu, 0)$-noisy, we have

$$\begin{aligned}
\forall y \in \{0,1\}^n: \quad \mathbb{E}_{x\sim\mu^y}[t(x)] &= \mathbb{E}_{x\sim\mu^y}\left[\textstyle\sum_{\ell\in\mathcal{A}(t)} \ell(x)\right] && \text{(Using (1))} \\
&\in \mathbb{E}_{x\sim\mu^y}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} \ell(x)\right] \pm o(1). && (2)
\end{aligned}$$

We now define the approximating conical junta by

$$p(x) \;:=\; \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} p_\ell(x)\right],$$

where the p_ℓ are given by Simulation Lemma. Hence p has degree at most

$$n \cdot [\delta \log n + \log(1/\epsilon)] \;=\; n \cdot [o(1)\log n + \log n^{o(1)}] \;=\; o(n \log n).$$

We finish the proof of Theorem 2 by verifying that p indeed $o(1)$-approximates $f \circ \textsc{GapOr}^n_{\log n}$.

$$\begin{aligned}
\forall x: \quad p(x) &= \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} p_\ell(x)\right] \\
&= \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} \ell(\mu^y)\right] && (y := \textsc{GapOr}^n_{\log n}(x)) \\
&= \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} \mathbb{E}_{x'\sim\mu^y}[\ell(x')]\right] \\
&= \mathbb{E}_{t\sim T}\left[\mathbb{E}_{x'\sim\mu^y}[\textstyle\sum_{\ell\in\mathcal{N}(t)} \ell(x')]\right] \\
&\in \mathbb{E}_{t\sim T}\left[\mathbb{E}_{x'\sim\mu^y}[t(x')]\right] \pm o(1) && \text{(Using (2))} \\
&= \mathbb{E}_{x'\sim\mu^y}\left[\mathbb{E}_{t\sim T}[t(x')]\right] \pm o(1) \\
&\in \mathbb{E}_{x'\sim\mu^y}\left[(f \circ g^n)(x')\right] \pm o(1) && (T \text{ has error } o(1)) \\
&= f(y) \pm o(1) \\
&= (f \circ \textsc{GapOr}^n_{\log n})(x) \pm o(1). && \square
\end{aligned}$$

4 Proof of Leaf Lemma

We prove Leaf Lemma in three subsections.

(§4.1) We start by recalling a distributional characterisation due to Blais and Brody [BB19] of expected query complexity $\overline{\mathrm{R}}_\epsilon$ using decision trees that can "abort".

(§4.2) We then formulate a Hard Side Lemma, which encapsulates the core challenge in finding the hard side of a given function g and from which Leaf Lemma is easy to derive.

(§4.3) Finally, we prove the Hard Side Lemma.

4.1 Distributional characterisation of $\overline{\mathrm{R}}_\epsilon$ due to Blais–Brody

A *(deterministic) abort-tree* t is a decision tree that outputs either a boolean value (0 or 1) or the *abort symbol* $\perp$. When an abort-tree is trying to compute a boolean function g, we do not consider the output $\perp$ as an "error"; the tree simply gives up on the computation. Indeed, we say that $t(x)$ *errs* iff $t(x) = 1 - g(x)$, that is, $t(x) \neq \perp$ and $t(x) \neq g(x)$. As before, a *randomised abort-tree* is a probability distribution over deterministic abort-trees. For $\gamma \in (0,1)$ and $\epsilon \in [0, 1/2)$ we define $\mathrm{R}_{\gamma,\epsilon}(g)$ as the least (worst-case) depth of a randomised abort-tree T such that for all $x \in \mathrm{dom}(g)$:

$$\mathbb{P}_{t\sim T}[\,t(x) = \perp\,] \;\le\; \gamma \quad \text{and} \quad \mathbb{P}_{t\sim T}[\,t(x)\,\mathrm{errs}\,] \;\le\; \epsilon.$$

We formulate a distributional version of $\mathrm{R}_{\gamma,\epsilon}(g)$ as follows. For a distribution μ over $\mathrm{dom}(g)$, we define $\mathrm{D}^{\mu}_{\gamma,\epsilon}(g)$ as the least depth of a deterministic abort-tree t such that

$$\mathbb{P}_{x\sim\mu}[\,t(x)=\bot\,] \;\leq\; \gamma \quad\text{and}\quad \mathbb{P}_{x\sim\mu}[\,t(x)\,\mathrm{errs}\,] \;\leq\; \epsilon.$$

The following two lemmas from [BB19, §3.1] connect abort-trees and $\overline{\mathrm{R}}_{\epsilon}(g)$.

Lemma 4 (Abort vs. expected depth)**.** *For every* $\epsilon\in[0,1/2)$ *and* $\gamma\in(0,1)$*,*

$$\gamma\cdot\mathrm{R}_{\gamma,\,\epsilon}(g) \;\leq\; \overline{\mathrm{R}}_{\epsilon}(g) \;\leq\; \tfrac{1}{1-\gamma}\cdot\mathrm{R}_{\gamma,\,(1-\gamma)\epsilon}(g).$$

Lemma 5 (Minimax)**.** *For every* $\epsilon\in[0,1/2)$*,* $\gamma\in(0,1)$*, and* $\alpha,\beta\in(0,1)$ *with* $\alpha+\beta\leq 1$*,*

$$\max\nolimits_{\mu}\mathrm{D}^{\mu}_{\gamma/\alpha,\,\epsilon/\beta}(g) \;\leq\; \mathrm{R}_{\gamma,\epsilon}(g) \;\leq\; \max\nolimits_{\mu}\mathrm{D}^{\mu}_{\alpha\gamma,\,\beta\epsilon}(g).$$

4.2 Statement of Hard Side Lemma

When searching for the hard side of a partial function g under a distribution $\mu=\frac{1}{2}(\mu^0+\mu^1)$, it is convenient to study a more symmetric notion of noisiness than the one-sided variant defined earlier. For a leaf $\ell\in\mathcal{L}(t)$ of an abort-tree t, we define the *relative error* $\mathrm{re}(\ell,\mu)$ so that if ℓ is an aborting leaf, then $\mathrm{re}(\ell,\mu)\coloneqq 0$; otherwise

$$\mathrm{re}(\ell,\mu) \;\coloneqq\; \frac{\min\{\ell(\mu^0),\,\ell(\mu^1)\}}{\ell(\mu^0)+\ell(\mu^1)} \;\in\; [0,1/2].$$

This definition captures the best achievable error of a leaf in an abort-tree. Namely, let us say that t is *μ-smart* if every non-abort leaf $\ell\in\mathcal{L}(t)$ outputs a boolean value $i\in\{0,1\}$ that maximises $\ell(\mu^i)$. Then for every leaf ℓ in a μ-smart t we have $\mathbb{P}_{x\sim\mu}[\,t(x)\,\mathrm{errs}\mid \ell^t_x=\ell\,]=\mathrm{re}(\ell,\mu)$. An easy calculation gives the following claim, which we record for future use.

Claim 6. *For a μ-smart t we have* $\mathbb{P}_{x\sim\mu}[\,t(x)\,\mathrm{errs}\,]=\mathbb{E}_{x\sim\mu}[\mathrm{re}(\ell^t_x,\mu)]$. ☐

Another easy calculation shows that relative error implies noisiness.

Claim 7. *If* $\mathrm{re}(\ell,\mu)\geq\epsilon$*, then ℓ is (ϵ,μ,b)-noisy for both* $b\in\{0,1\}$*.* ☐

We are now ready to formulate Hard Side Lemma, which isolates the technical challenge (C1) (discussed in Section 3.1): Every partial function g admits a balanced distribution μ and a hard side b such that if we run a shallow abort-tree on the hard side μ^b of μ, then t must either abort with high probability or we reach a leaf of noticeable error (in expectation).

Hard Side Lemma. *For every partial function g and $0<\epsilon\lll\delta\ll 1$, there exists a distribution $\mu=\frac{1}{2}(\mu^0+\mu^1)$ over $\mathrm{dom}(g)$ and a hard side $b\in\{0,1\}$ such that for any deterministic abort-tree t with $\mathrm{depth}(t)/\overline{\mathrm{R}}_{\epsilon}(g)\lll\delta$ we have either*

$$\mathbb{P}_{x\sim\mu^b}[\,t(x)=\bot\,] \;>\; 1-\delta \qquad\text{or}\qquad \mathbb{E}_{x\sim\mu^b}[\,\mathrm{re}(\ell^t_x,\mu)\,] \;>\; \epsilon. \tag{3}$$

We defer the proof until Section 4.3. We first use the lemma to prove Leaf Lemma, and here is where we address challenge (C2): we exploit the high abort probability (namely, $1-\delta$) guaranteed by Hard Side Lemma to show that typical leaves are noisy.

Leaf Lemma (restated)**.** *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over* $\mathrm{dom}(g)$ *and a hard side $b \in \{0,1\}$ such that for every deterministic tree t and $i \in \{0,1\}$:*

$$\frac{\mathbb{E}_{x\sim\mu^i}[q(t,x)]}{\overline{\mathrm{R}}_\epsilon(g)} \lll \delta \quad\implies\quad \mathbb{P}_{x\sim\mu^i}[\,\ell_x^t \text{ is } (\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Proof. We observe first that regardless of μ, b, or even the expected depth of t, the lemma holds for the easy side $i = 1-b$. Indeed, if we let $\mathcal{B} \subseteq \mathcal{L}(t)$ denote the set of non-(ϵ,μ,b)-noisy leaves,

$$\mathbb{P}_{x\sim\mu^{1-b}}[\ell_x^t \in \mathcal{B}] \;=\; \textstyle\sum_{\ell\in\mathcal{B}} \ell(\mu^{1-b}) \;<\; \epsilon \sum_{\ell\in\mathcal{B}} \ell(\mu^b) \;\leq\; \epsilon \sum_{\ell\in\mathcal{L}(t)} \ell(\mu^b) \;=\; \epsilon \;\leq\; \delta.$$

Let us then focus on the interesting case $i = b$ where the careful choice of μ and b is essential. We invoke Hard Side Lemma with parameters ϵ and $\dot\delta \coloneqq \delta^2$ (assuming suitably $0 < \epsilon \lll \dot\delta \ll 1$) to obtain μ and b such that for every abort-tree $\dot t$ with $\mathrm{depth}(\dot t) \leq \dot\delta^c \overline{\mathrm{R}}_\epsilon(g)$ the property (3) holds (with dotted parameters). Let $x \sim \mu^b$ henceforth and write $\mathrm{re}(\ell) \coloneqq \mathrm{re}(\ell,\mu)$ for short. Suppose t satisfies $\mathbb{E}_x[q(t,x)] \leq \dot\delta^{c+2}\overline{\mathrm{R}}_\epsilon(g)$ (where we chose $2(c+2)$ as the exponent hidden by $\lll$). Recalling from Claim 7 that relative error implies noisiness, our goal is to show

$$\mathbb{P}_x[\,\mathrm{re}(\ell_x^t) \geq \epsilon\,] \;\geq\; 1-\delta. \tag{4}$$

We convert t into an abort-tree by letting t' be a modification of t that aborts whenever more than $\dot\delta^c\overline{\mathrm{R}}_\epsilon(g)$ queries are made. Using Markov's inequality and the low expected depth of t,

$$\mathbb{P}_x[t'(x) = \bot] \;=\; \mathbb{P}_x[q(t,x) > \dot\delta^c\overline{\mathrm{R}}_\epsilon(g)] \;\leq\; \mathbb{E}_x[q(t,x)]/\dot\delta^c\overline{\mathrm{R}}_\epsilon(g) \;\leq\; \dot\delta^2.$$

We also have $\mathbb{P}_x[\mathrm{re}(\ell_x^t) \geq \epsilon] \geq \mathbb{P}_x[\mathrm{re}(\ell_x^{t'}) \geq \epsilon]$ since we only made more executions abort. To prove (4), suppose for contradiction that $\mathbb{P}_x[\mathrm{re}(\ell_x^{t'}) \geq \epsilon] < 1-\delta$. Let $\dot t$ be a further modification of t' that aborts any leaf $\ell \in \mathcal{L}(t')$ with $\mathrm{re}(\ell) \geq \epsilon$. Note that

$$\mathbb{P}_x[\dot t(x) = \bot] \;\leq\; \mathbb{P}_x[t'(x) = \bot] + \mathbb{P}_x[\mathrm{re}(\ell_x^{t'}) \geq \epsilon] \;\leq\; \dot\delta^2 + 1 - \delta \;\leq\; 1-\dot\delta.$$

Hence we get from (dotted) property (3) that $\mathbb{E}_x[\mathrm{re}(\ell_x^{\dot t})] > \epsilon$. But this contradicts the fact that $\mathrm{re}(\ell) < \epsilon$ for all $\ell \in \mathcal{L}(\dot t)$ by construction. This verifies (4) and concludes the proof. □

4.3 Proof of Hard Side Lemma

Let ν be a distribution that witnesses $D \coloneqq \max_{\nu'} \mathrm{D}^{\nu'}_{1-\delta,\,\epsilon^{1/3}}(g)$ so that every abort-tree t with $\mathrm{depth}(t) < D$ fails to satisfy at least one of the following:

$$\mathbb{P}_{x\sim\nu}[\,t(x) = \bot\,] \;\leq\; 1-\delta, \tag{5}$$

$$\mathbb{P}_{x\sim\nu}[\,t(x)\,\mathrm{errs}\,] \;\leq\; \epsilon^{1/3}. \tag{6}$$

As a minor technicality, we re-balance ν. We can write $\nu = \lambda\mu^0 + (1-\lambda)\mu^1$ where $\lambda \in (0,1)$ and μ^i is a distribution supported on $g^{-1}(i)$. We define $\mu \coloneqq \frac{1}{2}(\mu^0 + \mu^1)$ as our balanced distribution.

Assume towards a contradiction that there does not exist a hard side for μ, that is, the claim of the lemma fails for both $b \in \{0,1\}$. This means there exists two abort-trees t_0 and t_1 of depth at most $\delta^3\overline{\mathrm{R}}_\epsilon(g)$ (where we chose 3 as the exponent hidden by $\lll$) such that for both $b \in \{0,1\}$:

$$\mathbb{P}_{x\sim\mu^b}[\,t_b(x) = \bot\,] \;\leq\; 1-\delta, \tag{7}$$

$$\mathbb{E}_{x\sim\mu^b}[\,\mathrm{re}(\ell_x^{t_b},\mu)\,] \;\leq\; \epsilon. \tag{8}$$

We will use t_0 and t_1 to construct a third tree t that computes g too well relative to ν contradicting our choice of ν. We may assume wlog that t_0 and t_1 are *μ-smart*, since the properties (7)–(8) do not depend on the boolean leaf-labels (only whether a leaf aborts or not). We now define t as follows: On input x we run both $t_0(x)$ and $t_1(x)$; if $t_0(x) \neq \bot$, we output $t_0(x)$; otherwise we output $t_1(x)$. We will show that t has $\mathrm{depth}(t) < D$ and satisfies (5)–(6), which will contradict our choice of ν.

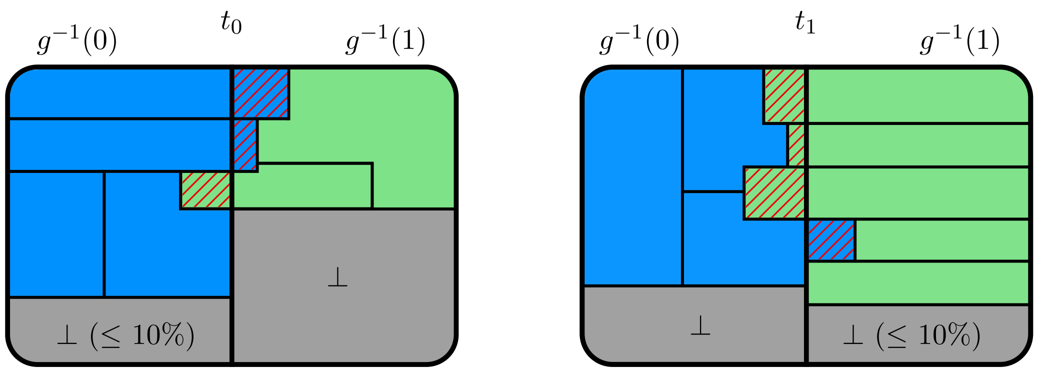

Figure 1: Two trees t_0 and t_1 in the proof of Hard Side Lemma. The leaves partition $\mathrm{dom}(g)$ into subcubes where grey leaves output $\perp$, green leaves output 1, and blue leaves output 0. Hatched regions are error. We are promised that, e.g., t_0 has bounded abort (10% in our figure) over μ^0, but not necessarily over μ^1.

Tree t is shallow. We have the following chain of inequalities

$$\begin{aligned}\mathrm{depth}(t) \;&\leq\; 2\delta^3\overline{\mathrm{R}}_\epsilon(g) \;\leq\; 32\delta^3\overline{\mathrm{R}}_{\epsilon^{1/4}}(g) \;<\; \delta^2\overline{\mathrm{R}}_{\epsilon^{1/4}}(g) \;\leq\; \mathrm{R}_{1-\delta^2,\,\delta^2\epsilon^{1/4}}(g)\\ &\leq\; \max_{\nu'} \mathrm{D}^{\nu'}_{(1-\delta^2)^2,\,\delta^4\epsilon^{1/4}}(g) \;\leq\; \max_{\nu'} \mathrm{D}^{\nu'}_{1-\delta,\,\epsilon^{1/3}}(g) \;=:\; D.\end{aligned}$$

The first inequality uses the definition of t. Second uses error reduction (Claim 3 with $k := 4$). Third uses $\delta \ll 1$. Fourth uses Lemma 4 (with $\gamma := 1-\delta^2$). Fifth uses the minimax lemma (Lemma 5 with $\alpha := 1-\delta^2$, $\beta := \delta^2$). The final inequality uses $\epsilon \lll \delta \ll 1$.

Tree t has bounded abort. We verify property (5) by

$$\begin{aligned}\mathbb{P}_{x\sim\nu}[t(x)=\perp] \;&=\; \mathbb{P}_{x\sim\nu}[t_0(x)=\perp \wedge t_1(x)=\perp]\\ &=\; \lambda\,\mathbb{P}_{x\sim\mu^0}[t_0(x)=\perp \wedge t_1(x)=\perp] + (1-\lambda)\,\mathbb{P}_{x\sim\mu^1}[t_0(x)=\perp \wedge t_1(x)=\perp]\\ &\leq\; \lambda\,\mathbb{P}_{x\sim\mu^0}[t_0(x)=\perp] + (1-\lambda)\,\mathbb{P}_{x\sim\mu^1}[t_1(x)=\perp]\\ &\leq\; 1-\delta. \qquad\qquad \text{(Using (7))}\end{aligned}$$

Tree t errs rarely. We start with a claim that says that if the expected relative error is low over one side μ^b of μ, then a μ-smart tree errs rarely over the whole distribution μ.

Claim 8. *Let t' be μ-smart and $b \in \{0,1\}$. If $\mathbb{E}_{x\sim\mu^b}[\mathrm{re}(\ell_x^{t'},\mu)] \leq \epsilon$ then $\mathbb{P}_{x\sim\mu}[t'(x)\,\mathrm{errs}] \leq \epsilon^{1/2}$.*

Proof. We prove the claim for $b=0$ as the other case is analogous. Since t' and μ are fixed, we drop them from notation writing $\mathrm{re}(\ell) := \mathrm{re}(\ell,\mu)$, $\ell_x := \ell_x^{t'}$, $\mathcal{L} := \mathcal{L}(t')$. We argue that relative error on one side of the distribution must spill over to the other side:

$$\mathbb{E}_{x\sim\mu^0}[\mathrm{re}(\ell_x)] \;=\; \textstyle\sum_{\ell\in\mathcal{L}} \ell(\mu^0)\,\mathrm{re}(\ell) \;\geq\; \sum_{\ell\in\mathcal{L}} \ell(\mu^1)\,\mathrm{re}(\ell)^2 \;=\; \mathbb{E}_{x\sim\mu^1}[\mathrm{re}(\ell_x)^2] \;\geq\; \mathbb{E}_{x\sim\mu^1}[\mathrm{re}(\ell_x)]^2.$$

Here the first inequality used $\ell(\mu^0) \geq \ell(\mu^1)\,\mathrm{re}(\ell)$ (from Claim 7) and the second inequality used Jensen's inequality. It follows that $\mathbb{E}_{x\sim\mu^1}[\mathrm{re}(\ell_x)] \leq \mathbb{E}_{x\sim\mu^0}[\mathrm{re}(\ell_x)]^{1/2} \leq \epsilon^{1/2}$ and therefore $\mathbb{E}_{x\sim\mu}[\mathrm{re}(\ell_x)] \leq \epsilon^{1/2}$. The claim then follows from Claim 6. □

We now verify property (6), which concludes the proof of Hard Side Lemma.

$$\begin{aligned}
\mathbb{P}_{x\sim\nu}[t(x)\,\text{errs}] &\le \mathbb{P}_{x\sim\nu}[t_0(x)\,\text{errs} \vee t_1(x)\,\text{errs}] \\
&\le \textstyle\sum_{b\in\{0,1\}} \mathbb{P}_{x\sim\nu}[t_b(x)\,\text{errs}] \\
&= \textstyle\sum_{b\in\{0,1\}} \lambda\,\mathbb{P}_{x\sim\mu^0}[t_b(x)\,\text{errs}] + (1-\lambda)\,\mathbb{P}_{x\sim\mu^1}[t_b(x)\,\text{errs}] \\
&\le \textstyle\sum_{b\in\{0,1\}} 2\,\mathbb{P}_{x\sim\mu}[t_b(x)\,\text{errs}] \\
&\le \textstyle\sum_{b\in\{0,1\}} 2\cdot\epsilon^{1/2} && \text{(Claim 8 and (8))} \\
&= 4\epsilon^{1/2} \\
&\le \epsilon^{1/3}. && (\epsilon \ll 1)
\end{aligned}$$

5 Proof of Multileaf Lemma

Multileaf Lemma (restated)**.** *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over $\operatorname{dom}(g)$ and a hard side $b \in \{0,1\}$ such that for every deterministic tree t taking inputs from $\operatorname{dom}(g^n)$ and having $\operatorname{depth}(t)/(n\overline{\mathrm{R}}_\epsilon(g)) \lll \delta$,*

$$\forall y \in \{0,1\}^n : \qquad \mathbb{P}_{x\sim\mu^y}[\,\ell_x^t \text{ is } (\delta,\epsilon,\mu,b)\text{-noisy}\,] \;\ge\; 1-\delta.$$

Proof. Apply Leaf Lemma with parameters ϵ and $\dot\delta := \delta^3$ (assuming suitably $0 < \epsilon \lll \dot\delta \ll 1$) to obtain $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ and $b \in \{0,1\}$ that satisfy the lemma for trees of depth at most $\dot\delta^c\overline{\mathrm{R}}_\epsilon(g)$. Fix $y \in \{0,1\}^n$ and a deterministic tree t over $\operatorname{dom}(g^n)$ with $\operatorname{depth}(t) \le \dot\delta^{c+4} n\overline{\mathrm{R}}_\epsilon(g)$ (where we chose $3(c+4)$ as the exponent hidden by $\lll$).

Here is the plan for our proof. An input $x \in \operatorname{dom}(g^n)$ can be seen as inducing several subtrees of t corresponding to distinct coordinates $i \in [n]$. Indeed, define $t^{x,i}$ as the tree over inputs from $\operatorname{dom}(g)$ that is obtained from t by substituting x as its input variables except retaining x^i as free variables. If we can show that $t^{x,i}$ has shallow depth in expectation over an input $z \sim \mu^{y_i}$ then we can hope to use Leaf Lemma and argue that the reached leaf $\ell_z \in \mathcal{L}(t^{x,i})$ (which is one of the n components of a leaf of t) is typically (ϵ,μ,b)-noisy.

Let us formalise this plan. Let $x \sim \mu^y$ henceforth. For $i \in [n]$ we define two events

$$\begin{aligned}
&\textit{i-th tree is shallow:} && S_i(x) && \overset{\text{def}}{\iff} && \mathbb{E}_{z\sim\mu^{y_i}}[q(t^{x,i},z)] \;\le\; \dot\delta^c\overline{\mathrm{R}}_\epsilon(g), \\
&\textit{i-th leaf is noisy:} && N_i(x) && \overset{\text{def}}{\iff} && \ell_{x^i} \in \mathcal{L}(t^{x,i}) \text{ is } (\epsilon,\mu,b)\text{-noisy}.
\end{aligned}$$

Note that Leaf Lemma states $\mathbb{P}_x[N_i \mid S_i] \ge 1-\dot\delta$. Thinking of S_i and N_i as indicator variables, we define $S := \frac{1}{n}\sum_i S_i$ and $N := \frac{1}{n}\sum_i N_i$. With this notation, Multileaf Lemma becomes equivalent to

$$\mathbb{P}_x[N \ge 1-\delta] \;\ge\; 1-\delta. \tag{9}$$

To show this, we compute as follows (using Claim 9 that is proved below)

$$\begin{aligned}
\mathbb{E}_x[N] &= \tfrac{1}{n}\textstyle\sum_i \mathbb{P}_x[N_i] \\
&\ge \tfrac{1}{n}\textstyle\sum_i (1-\dot\delta)\,\mathbb{P}[S_i] && \text{(Leaf Lemma)} \\
&= (1-\dot\delta)\,\mathbb{E}_x[S] \\
&\ge (1-\dot\delta)(1-\dot\delta) && \text{(Claim 9)} \\
&\ge 1-\delta^2. && (\dot\delta := \delta^3 \ll 1)
\end{aligned}$$

Hence (9) follows by applying Markov's inequality to the nonnegative random variable $1 - N \ge 0$. This completes the proof apart from the following claim. ☐

Claim 9. $\mathbb{E}_x[S] \geq 1 - \dot{\delta}$.

Proof. Let $q_i(t, x)$ denote the number of queries made by t to the i-th component of x. Define $x^{i \leftarrow z}$ as a copy of x but where z is inserted at the i-th component. Note that $q_i(t, x^{i \leftarrow z}) = q(t^{x,i}, z)$. Linearity of expectation gives

$$\textstyle\sum_{i \in [n]} \mathbb{E}_x[q_i(t, x)] \;\leq\; \text{depth}(t) \;\leq\; \dot{\delta}^{c+4} n \overline{\mathrm{R}}_\epsilon(g). \tag{10}$$

Define $\mathcal{I} \subseteq [n]$ as the set of coordinates i satisfying

$$\mathbb{E}_x[q_i(t, x)] \;\leq\; \dot{\delta}^{c+2} \overline{\mathrm{R}}_\epsilon(g). \tag{11}$$

We have that $|\mathcal{I}| \geq (1 - \dot{\delta}^2)n$ as otherwise more than $\dot{\delta}^2 n$ terms in the sum (10) are larger than $\dot{\delta}^{c+2}\overline{\mathrm{R}}_\epsilon(g)$ contradicting the upper bound on depth(t). Fix $i \in \mathcal{I}$. Sampling $x \sim \mu^y$ is equivalent to first taking $x \sim \mu^y$, then sampling independently $z \sim \mu^{y_i}$, and finally outputting $x^{i \leftarrow z}$. Hence

$$\mathbb{E}_x \, \mathbb{E}_{z \sim \mu^{y_i}}[q_i(t, x^{i \leftarrow z})] \;=\; \mathbb{E}_x[q_i(t, x)] \;\leq\; \dot{\delta}^{c+2} \overline{\mathrm{R}}_\epsilon(g).$$

We get from Markov's inequality and the above that

$$\mathbb{P}_x[\neg S_i] \;=\; \mathbb{P}_x\left[\mathbb{E}_{z \sim \mu^{y_i}}[q_i(t, x^{i \leftarrow z})] > \dot{\delta}^c \overline{\mathrm{R}}_\epsilon(g)\right] \;\leq\; \dot{\delta}^2. \tag{12}$$

In conclusion,

$$\mathbb{E}_x[S] \;\geq\; \tfrac{1}{n} \textstyle\sum_{i \in \mathcal{I}} \mathbb{P}_x[S_i] \;\geq\; \tfrac{1}{n}|\mathcal{I}| \cdot (1 - \dot{\delta}^2) \;\geq\; (1 - \dot{\delta}^2)^2 \;\geq\; 1 - \dot{\delta}.$$

□

Acknowledgements

We thank Thomas Watson for helpful comments on an earlier draft of this paper.

References

[AGJ+17] Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity. In *Proceedings of the 37th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 10:1–10:13. Schloss Dagstuhl, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.10.

[BB19] Eric Blais and Joshua Brody. Optimal separation and strong direct sum for randomized query complexity. In *Proceedings of the 34th Computational Complexity Conference (CCC)*, pages 29:1–29:17. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.CCC.2019.29.

[BB20a] Shalev Ben-David and Eric Blais. A new minimax theorem for randomized algorithms. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*, pages 403–411, 2020. doi:10.1109/FOCS46700.2020.00045.

[BB20b] Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*, pages 240–246, 2020. doi:10.1109/FOCS46700.2020.00031.

[BDG+20] Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. The power of many samples in query complexity. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168, pages 9:1–9:18. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.ICALP.2020.9.

[BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

[BGKW20] Shalev Ben-David, Mika Göös, Robin Kothari, and Thomas Watson. When is amplification necessary for composition in randomized query complexity? In *Proceedings of the 22nd International Conference on Randomization and Computation (RANDOM)*, volume 176, pages 28:1–28:16. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.28.

[BK18] Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. *Theory of Computing*, 14(1):1–27, 2018. doi:10.4086/toc.2018.v014a005.

[BKLS20] Joshua Brody, Jae Tak Kim, Peem Lerdputtipongporn, and Hariharan Srinivasulu. A strong XOR lemma for randomized query complexity. Technical report, arXiv, 2020. arXiv:2007.05580.

[DR08] Chinmoy Dutta and Jaikumar Radhakrishnan. Lower bounds for noisy wireless networks using sampling algorithms. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS)*, pages 394–402. IEEE, 2008. doi:10.1109/FOCS.2008.72.

[EP98] William Evans and Nicholas Pippenger. Average-case lower bounds for noisy boolean decision trees. *SIAM Journal on Computing*, 28(2):433–446, 1998. doi:10.1137/S0097539796310102.

[FRPU94] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994. doi:10.1137/S0097539791195877.

[GLM+16] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016. doi:10.1137/15M103145X.

[GLSS19] Dmitry Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity via max-conflict complexity. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 64:1–64:13. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.ICALP.2019.64.

[GS10] Navin Goyal and Michael Saks. Rounds vs. queries tradeoff in noisy computation. *Theory of Computing*, 6(1):113–134, 2010. doi:10.4086/toc.2010.v006a006.

[JK10] Rahul Jain and Hartmut Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Conference on Computational Complexity (CCC)*, pages 247–258. IEEE, 2010. doi:10.1109/CCC.2010.31.

[JKS10] Rahul Jain, Hartmut Klauck, and Miklos Santha. Optimal direct sum results for deterministic and randomized decision tree complexity. *Information Processing Letters*, 110(20):893–897, 2010. doi:10.1016/j.ipl.2010.07.020.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[KK94] Claire Kenyon and Valerie King. On boolean decision trees with faulty nodes. *Random Structures and Algorithms*, 5(3):453–464, 1994. doi:10.1002/rsa.3240050306.

[KLdW15] Jedrzej Kaniewski, Troy Lee, and Ronald de Wolf. Query complexity in expectation. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 761–772. Springer, 2015. doi:10.1007/978-3-662-47672-7_62.

[LMR+11] Troy Lee, Rajat Mittal, Ben Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS)*, pages 344–353. IEEE, 2011. doi:10.1109/FOCS.2011.75.

[New09] Ilan Newman. Computing in fault tolerant broadcast networks and noisy decision trees. *Random Structures and Algorithms*, 34(4):478–501, 2009. doi:10.1002/rsa.20240.

[Rei11] Ben Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)*, pages 560–569. SIAM, 2011.

[Sav02] Petr Savický. On determinism versus unambiquous nondeterminism for decision trees. Technical Report TR02-009, Electronic Colloquium on Computational Complexity (ECCC), 2002. URL: http://eccc.hpi-web.de/report/2002/009/.

[Tal13] Avishay Tal. Properties and applications of boolean function composition. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 441–454. ACM, 2013. doi:10.1145/2422436.2422485.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il