Electronic Colloquium on Computational Complexity, Report No. 112 (2022)

Randomised Composition and Small-Bias Minimax

Shalev Ben-David
University of Waterloo

Eric Blais
University of Waterloo

Mika Göös
EPFL

Gilbert Maystre
EPFL

August 12, 2022

Abstract

We prove two results about randomised query complexity $\mathsf{R}(f)$. First, we introduce a *linearised* complexity measure LR and show that it satisfies an *inner-optimal* composition theorem: $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{R}(f)\mathsf{LR}(g))$ for all partial f and g, and moreover, LR is the largest possible measure with this property. In particular, LR can be polynomially larger than previous measures that satisfy an inner composition theorem, such as the max-conflict complexity of Gavinsky, Lee, Santha, and Sanyal (ICALP 2019).

Our second result addresses a question of Yao (FOCS 1977). He asked if ϵ-error *expected* query complexity $\overline{\mathsf{R}}_\epsilon(f)$ admits a distributional characterisation relative to some hard input distribution. Vereshchagin (TCS 1998) answered this question affirmatively in the bounded-error case. We show that an analogous theorem *fails* in the small-bias case $\epsilon = 1/2 - o(1)$.

Contents

ISSN 1433-8092

1 Introduction

This paper is motivated by the following basic open problem in boolean function complexity theory.

Conjecture 1. $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{R}(f)\mathsf{R}(g))$ *for all total boolean functions* f, g.

Let us unpack what this conjecture is claiming. The randomised ϵ-error query complexity $\mathsf{R}_\epsilon(f)$ of a boolean function $f \colon \{0,1\}^n \to \{0,1\}$ is defined (see [BdW02] for the classic reference) as the least number of queries a randomised algorithm (decision tree) needs to make, on the worst-case input, to the bits x_i of $x \in \{0,1\}^n$ in order to compute $f(x)$ correctly with error at most ϵ. We write $\mathsf{R} \coloneqq \mathsf{R}_{1/3}$ for the bounded-error case. For functions f and g over n and m bits, their composition $f \circ g$ is defined over nm bits by

$$(f \circ g)(x) \;\coloneqq\; f(g(x^1), \dots, g(x^n)) \qquad \text{where} \qquad x = (x^1, \dots, x^n) \in (\{0,1\}^m)^n.$$

In particular, we have $\mathsf{R}(f \circ g) \leq O(\mathsf{R}(f)\mathsf{R}(g) \log \mathsf{R}(f))$ for all f, g. This holds since we can run an algorithm for f with query cost $\mathsf{R}(f)$ and whenever it queries an input bit, we can run, as a subroutine, an ϵ-error algorithm for g of cost $\mathsf{R}_\epsilon(g)$. Setting $\epsilon \ll 1/\mathsf{R}(f)$ makes sure that the errors made by the subroutines do not add up. Moreover, we have $\mathsf{R}_\epsilon(g) \leq O(\mathsf{R}(g) \log(1/\epsilon)) = O(\mathsf{R}(g) \log \mathsf{R}(f))$ by standard error reduction techniques. Conjecture 1 thus postulates that a converse inequality always holds (without the log factor).

The analogue of Conjecture 1 has been long resolved for many other well-studied complexity measures: deterministic query complexity satisfies a perfect multiplicative composition theorem, $\mathsf{D}(f \circ g) = \mathsf{D}(f)\mathsf{D}(g)$ [Sav02], quantum query complexity satisfies $\mathsf{Q}(f \circ g) = \Theta(\mathsf{Q}(f)\mathsf{Q}(g))$ [Rei11, LMR+11], and yet more examples (degree, certificate complexity, sensitivity, rank) are discussed in [Tal13, GSS16, DM21]. In the randomised case, however, the conjecture has proved more delicate, exhibiting a far richer, and more surprising, structure.

Partial counterexamples. Conjecture 1 is known to be false if we relax the requirement that f, g are total and instead consider *partial* functions (promise problems), which are undefined on some inputs x, $f(x) = *$. Indeed, works by Gavinsky, Lee, Santha, and Sanyal [GLSS19] and Ben-David and Blais [BB20b] have culminated in examples of partial functions f, g such that $\mathsf{R}(f \circ g) \leq o(\mathsf{R}(f)\mathsf{R}(g))$. Motivated by these counterexamples, we ask: *What is the best possible composition theorem one can prove for partial functions?*

1.1 A new composition theorem

Our first result is an *inner-optimal* composition theorem for partial functions. To state this result, we start by introducing a new *linearised* complexity measure defined for a partial function $f \colon \{0,1\}^n \to \{0,1,*\}$ by

$$\mathsf{LR}(f) \;\coloneqq\; \min_R \max_x \frac{\operatorname{cost}(R,x)}{\operatorname{bias}_f(R,x)},$$

- where R ranges over randomised decision trees;
- x ranges over the domain of f, namely, $\operatorname{Dom}(f) \coloneqq f^{-1}(\{0,1\})$;
- $\operatorname{cost}(R,x)$ denotes the *expected* number of queries R makes on input x; and
- $\operatorname{bias}_f(R,x)$ denotes the bias R has of guessing the value $f(x)$ correctly; formally, $\operatorname{bias}_f(R,x) \coloneqq \max\{1 - 2\operatorname{err}_f(R,x), 0\}$ where $\operatorname{err}_f(R,x) \coloneqq \Pr_R[R(x) \neq f(x)]$. We often omit the subscript f for brevity.

This definition might seem mysterious at first sight. To get better acquainted with it, let us first note that

$$\forall f\colon \qquad \Omega(\sqrt{\mathsf{R}(f)}) \;\leq\; \mathsf{LR}(f) \;\leq\; O(\mathsf{R}(f)). \tag{1}$$

Indeed, the second inequality follows by considering a bounded-error decision tree R, with $\operatorname{cost}(R,x) \leq \mathsf{R}(f)$ and $\operatorname{bias}(R,x) \geq 1/3$. For the first inequality, if we let R be a randomised tree that achieves the minimum in the definition of $\mathsf{LR}(f)$, we can amplify the bias of R, which is possibly tiny, as follows. On input x we run $R(x)$ repeatedly until we have made a total of $\mathsf{LR}(f)^2$ queries, and then output the majority answer over all runs. We expect this simulation to run $R(x)$ for $\mathsf{LR}(f)^2/\operatorname{cost}(R,x) \geq 1/\operatorname{bias}(R,x)^2$ many times, which, by standard Chernoff bounds, is enough to amplify the bias to a constant. This shows $\mathsf{R}(f) \leq O(\mathsf{LR}(f)^2)$.

Both extremes in (1) can be realised. First, consider the n-bit *parity* function $\textsc{Xor}_n$. It is not hard to see that any randomised tree that achieves bias δ for $\textsc{Xor}_n$ needs to query all the n bits with probability at least δ, resulting in expected query cost at least δn. This shows $\mathsf{LR}(\textsc{Xor}_n) = \mathsf{R}(\textsc{Xor}_n) = n$. Second, consider the partial n-bit *gap-majority* function (here $|x|$ denotes the Hamming weight)

$$\textsc{GapMaj}_n(x) \;\coloneqq\; \begin{cases} 1 & \text{if } |x| \geq n/2 + \sqrt{n}, \\ 0 & \text{if } |x| \leq n/2 - \sqrt{n}, \\ * & \text{otherwise.} \end{cases}$$

It is well known that $\mathsf{R}(\textsc{GapMaj}_n) = \Theta(n)$. By contrast, the algorithm R that queries and outputs a uniform random bit of x has $\operatorname{cost}(R, x) = 1$ and $\operatorname{bias}(R, x) \geq \Omega(1/\sqrt{n})$, which shows $\mathsf{LR}(\textsc{GapMaj}_n) \leq O(\sqrt{n})$.

Our first main result shows that a multiplicative composition theorem holds when the inner function is measured according to LR, and moreover, our choice of LR is optimal among all inner complexity measures. Ultimately, these theorems are what lends naturalness to our definition of LR.

Theorem 1. $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{R}(f)\mathsf{LR}(g))$ *for all partial boolean functions* f, g.

Theorem 2. *Theorem 1 is optimal: If* M *is any complexity measure such that* $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{R}(f)\mathsf{M}(g))$ *for all partial* f, g*, then* $\mathsf{LR}(g) \geq \Omega(\mathsf{M}(g))$ *for all partial* g.

Additionally, LR itself satisfies a composition theorem as well.

Theorem 3. $\mathsf{LR}(f \circ g) \geq \Omega(\mathsf{LR}(f)\mathsf{LR}(g))$ *for all partial boolean functions* f, g.

1.2 Comparison with previous work

The randomised composition conjecture for general boolean functions was first explicitly raised in [BK16]. Several complexity measures have since been shown to satisfy an inner composition theorem, including:

1. (block-)sensitivity s, bs [ABK16],
2. randomised sabotage complexity RS [BK16],
3. randomised complexity R_δ with small-bias error $\delta \coloneqq 1/2 - 1/n^4$ [AGJ+18],
4. max-conflict complexity $\overline{\chi}$ [GLSS19] (also studied in [Li21]).

By our optimality theorem, we have $\mathsf{LR}(f) \geq \Omega(\mathsf{M}(f))$ for all $\mathsf{M} \in \{\mathsf{s}, \mathsf{bs}, \mathsf{RS}, \mathsf{R}_\delta, \overline{\chi}\}$ and all f. In fact, we can show that the largest of the above measures, namely $\overline{\chi}$, can sometimes be polynomially smaller than LR.[1]

Lemma 4. *There exists a partial* f *such that* $\mathsf{LR}(f) \geq \Omega(\overline{\chi}(f)^{1.5})$.

Previous work has also investigated complexity measures M that admit an *outer* composition theorem, that is, $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{M}(f)\mathsf{R}(g))$ for all partial f, g. These measures include:

1. sensitivity s [GJPW18] (which was applied in [AKK16]),
2. fractional block sensitivity fbs [BDG+20],
3. noisy randomised complexity noisyR [BB20b] (also studied in [GTW21]).

In particular, noisyR is known to be *outer-optimal*: if we have $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{M}(f)\mathsf{R}(g))$ for all partial f, g, then $\mathsf{noisyR}(f) \geq \Omega(\mathsf{M}(f))$ for all partial f. Our result can be viewed as an inner analogue of this.

Finally, we mention that randomised composition has also been studied in the *super-multiplicative* regime, where we have examples of functions f, g with $\mathsf{R}(f \circ g) \geq \omega(\mathsf{R}(f)\mathsf{R}(g))$. Tight bounds exist when the outer function is identity [BB19] (building on [JKS10, BK16]), parity [BKLS20], or majority [BGKW20, GM21].

[1] Technically, it does not seem to be known in the literature whether R_δ is always at most $\overline{\chi}$; this doesn't matter much for our purposes, as LR is larger than both and it is easy to separate LR from R_δ (for example with the $\textsc{Or}$ function).

1.3 On small-bias minimax

Our second result addresses a question of Yao [Yao77]. Yao-style minimax theorems are routinely used to construct and analyse hard input distributions (including in our proof of the new composition theorem). For example, R_ϵ admits a distributional characterisation as

$$\mathsf{R}_\epsilon(f) \;=\; \max_{\mu} \min_{R\in\mathsf{R}(f,\epsilon,\mu)} \operatorname{depth}(R), \tag{2}$$

where μ ranges over distributions on $\operatorname{Dom}(f)$; the set $\mathsf{R}(f,\epsilon,\mu)$ consists of trees R with $\mathbb{E}_{x\sim\mu}[\operatorname{err}(R,x)] \leq \epsilon$; and $\operatorname{depth}(R)$ is the worst-case cost of R, that is, maximum number of queries over all inputs (and internal randomness if R is randomised). While the worst-case cost setting is perhaps what is most widely studied up to this day, Yao's original paper discussed, in fact, exclusively the expected cost setting. It is the expected cost setting that is currently undergoing a renaissance as it has proven important in the randomised composition literature surveyed above (Section 1.2).

Minimax for expected cost. We define the *ϵ-error expected query complexity* and the *ϵ-error distributional expected query complexity* by

$$\begin{aligned}\overline{\mathsf{R}}_\epsilon(f) \;&\coloneqq\; \min_{R\in\mathsf{R}(f,\epsilon)} \max_{x} \operatorname{cost}(R,x),\\ \overline{\mathsf{D}}_\epsilon(f) \;&\coloneqq\; \max_{\mu} \min_{R\in\mathsf{R}(f,\epsilon,\mu)} \operatorname{cost}(R,\mu),\end{aligned}$$

where $\mathsf{R}(f,\epsilon)$ is the set of randomised trees R such that $\operatorname{err}(R,x) \leq \epsilon$ for all inputs x; and $\operatorname{cost}(R,\mu) \coloneqq \mathbb{E}_{x\sim\mu}[\operatorname{cost}(R,x)]$ is the expected cost over μ (and internal randomness of R). We note that the set $\mathsf{R}(f,\epsilon,\mu)$ is sometimes restricted to contain only deterministic algorithms wlog (as can be done in (2)), but in the expected cost setting this may not necessarily be the case (see Open Problem 4); hence we allow $\mathsf{R}(f,\epsilon,\mu)$ to contain randomised trees.

Yao showed an exact distributional characterisation for zero-error algorithms, namely, $\overline{\mathsf{R}}_0(f) = \overline{\mathsf{D}}_0(f)$, and moreover, the optimal distributional algorithm is deterministic. He asked if a similar characterisation holds in the case $\epsilon > 0$. He observed that the "easy" direction of minimax, $\overline{\mathsf{D}}_\epsilon(f) \leq \overline{\mathsf{R}}_\epsilon(f)$, certainly holds (although Yao's version of this inequality had some loss in parameters as he was restricted to deterministic algorithms). Vereshchagin [Ver98] proved the "hard" direction with a modest loss in parameters; in summary,

$$\overline{\mathsf{D}}_\epsilon(f) \;\leq\; \overline{\mathsf{R}}_\epsilon(f) \;\leq\; 2\overline{\mathsf{D}}_{\epsilon/2}(f).$$

These bounds give a satisfying distributional characterisation in the bounded-error case. What happens in the small-bias case $\epsilon = 1/2 - o(1)$? Our second result shows that, surprisingly, the distributional characterisation fails in a particularly strong sense. We write $\dot{\delta} = (1-\delta)/2$ for short.

Theorem 5. *There is an n-bit partial function f and a bias $\delta(n) = o(1)$ such that $\overline{\mathsf{R}}_{\dot{\delta}}(f) \geq \overline{\mathsf{D}}_{\dot{\delta}}(f)^{1+\Omega(1)}$.*

This theorem says that there is no way to capture $\overline{\mathsf{R}}_\epsilon(f)$ relative to a *single* hard distribution. However, there does exist a distributional characterisation using a pair of distributions, as we explore next.

1.4 Discussion: How are our two results related?

Suppose we want to prove an inner composition theorem. All the previous proofs [BK16, AGJ+18, GLSS19] revolve around the following high-level idea. Let R be a randomised tree that on input x seeks to compute $f(g(x^1),\dots,g(x^n))$. The tree can invest different numbers of queries q_i to different components x^i, making $q = \sum_i q_i$ queries in total. If we had a complexity measure $\mathsf{M}(g)$ that allowed us to bound the bias the tree has for the i-th component $g(x^i)$ as *a linear function* of q_i—say, the bias for $g(x^i)$ is at most $q_i/\mathsf{M}(g)$—then, by *linearity of expectation*, the expected total sum of the biases for all components $g(x^1),\dots,g(x^n)$ is at most $q/\mathsf{M}(g)$. This would allow us to track the total progress R is making across all the inner functions.

What is the largest such "linearised" measure M? The most natural attempt at a definition (which the authors of this paper studied for a long time before finding the correct definition of LR) runs as follows. The measure should be such that with $q \coloneqq \overline{\mathsf{R}}_{\dot{\delta}}(f)$ queries one gets bias at most $\delta \le q/\mathsf{M}(f)$. Optimising for $\mathsf{M}(f)$ this suggest the following definition (a competitor for LR)

$$\mathsf{ULR}(f) \;\coloneqq\; \min_{\delta>0} \frac{\overline{\mathsf{R}}_{\dot{\delta}}(f)}{\delta} \;=\; \min_{R} \max_{x,y} \frac{\operatorname{cost}(R,x)}{\operatorname{bias}(R,y)}.$$

We call it *uniform*-LR, since the tree R that achieves the minimum has an upper bound on $\operatorname{cost}(R,x)$ that is uniformly the same for all x, and similarly there is a uniform lower bound on $\operatorname{bias}(R,x)$ for all x. By contrast, the definition of $\mathsf{LR}(f)$ is *non-uniform*: a tree R that achieves the minimum for $\mathsf{LR}(f)$ has only a bound on the cost/bias *ratio*, but the individual cost and bias functions can vary wildly as a function of x.

We clearly have $\mathsf{LR}(f) \le \mathsf{ULR}(f)$ by definition. How about the converse? It is enlightening to compare the distributional characterisations of these two measures, which can be derived using the recent minimax theorem for ratios of bilinear functions [BB20a]:

$$\mathsf{LR}(f) \;\coloneqq\; \min_{R} \max_{x} \frac{\operatorname{cost}(R,x)}{\operatorname{bias}(R,x)} \;=\; \max_{\mu} \min_{R} \frac{\operatorname{cost}(R,\mu)}{\operatorname{bias}(R,\mu)}, \tag{3}$$

$$\mathsf{ULR}(f) \;\coloneqq\; \min_{R} \max_{x,y} \frac{\operatorname{cost}(R,x)}{\operatorname{bias}(R,y)} \;=\; \max_{\mu,\nu} \min_{R} \frac{\operatorname{cost}(R,\mu)}{\operatorname{bias}(R,\nu)}. \tag{4}$$

Here, LR is captured using a single hard distribution μ such that both cost and bias are measured against it. By contrast, ULR needs a pair of distributions μ, ν, one to measure the cost, one to measure the bias. The upshot is that we are able to show that the two measures are polynomially separated.

Theorem 6. *There is an n-bit partial function f such that $\mathsf{ULR}(f) \ge \Omega(\mathsf{LR}(f)^{5/4}) \ge n^{\Omega(1)}$.*

Our optimality theorem thus implies that ULR *cannot* satisfy an inner composition theorem. This means that our attempt at finding a "linearised" measure at the start of this section missed a subtlety, namely, Yao's question: can we capture our measure relative to a single hard distribution? Our proof of the composition theorem will rely heavily on the fact that LR admits a single hard distribution. Our separation of LR and ULR is what allows us to prove the impossibility of capturing $\overline{\mathsf{R}}_{\epsilon}(f)$ relative to a single distribution. Indeed, Theorem 5 can be derived from Theorem 6 simply as follows.

Proof of Theorem 5. Let f be as in Theorem 6 and let R be a randomised tree witnessing $\mathsf{LR}(f)$. We may assume wlog that $\operatorname{cost}(R,x) \ge 1$ for all x. (If R places a lot of weight on a 0-cost tree, we may re-weight R without affecting the cost/bias ratio; see Lemma 9 for details.) Thus $\operatorname{bias}(R,x) \ge 1/n \eqqcolon \delta$ for all x. We show the following inequalities, which would prove Theorem 5.

$$\overline{\mathsf{R}}_{\dot{\delta}}(f) \;\ge\; \delta \cdot \mathsf{ULR}(f), \tag{5}$$

$$\overline{\mathsf{D}}_{\dot{\delta}}(f) \;\le\; \delta \cdot \mathsf{LR}(f), \tag{6}$$

Indeed, (5) holds since $\mathsf{ULR}(f) \le \overline{\mathsf{R}}_{\dot{\delta}}(f)/\delta$ by the definition of ULR. For (6) consider any input distribution μ. Define R' as the randomised tree that with probability $\lambda \coloneqq \delta/\operatorname{bias}(R,\mu)$ runs R, and with probability $1-\lambda$ makes no queries and outputs a random 0/1 answer. Then $\operatorname{bias}(R',\mu) = \lambda\operatorname{bias}(R,\mu) = \delta$ and $\operatorname{cost}(R',\mu) = \lambda\operatorname{cost}(R,\mu) = \delta\operatorname{cost}(R,\mu)/\operatorname{bias}(R,\mu) \le \delta\mathsf{LR}(f)$, as desired. □

1.5 Techniques

Composition theorem. Our first result, the inner-optimal composition theorem, is proved in Part I. As in other composition theorems for randomised algorithms, we start with a randomised algorithm R for the composition $f \circ g$ as well as hard distributions μ_0 and μ_1 for g (corresponding to distributions on $g^{-1}(0)$ and $g^{-1}(1)$), and we construct a randomised algorithm R' for f whose cost is significantly lower than that of R (we need the cost to decrease by a factor of $\mathsf{LR}(g)$). The algorithm R' will simulate R, but not every query that R makes to the large, mn-sized input to $f \circ g$ will turn into a query to the smaller, n-sized input to f that R' has access to. Instead, R' will attempt to delay making a true query as long as possible, and instead when R makes a query (i,j) (querying position j inside copy i of an input to g), R' will return an answer that is generated according to μ_0 and μ_1, so long as these two distributions approximately agree on the answer to that query.

So far, this is the same strategy employed by several other composition theorems, including in particular that of [GLSS19]. Our innovation comes from the precise way we choose when to query the bit i versus when to return an artificially-generated query answer to the query (i, j). Specifically, in Section 3, we prove the following simulation theorem for decision trees. Suppose we are given two distributions μ_0 and μ_1, we are asked to answer online queries to the bits of a string sampled from μ_b without knowing the value of b; moreover, suppose we have access to a big red button that, when pressed, provides the value of $b \in \{0, 1\}$. Then there is a strategy to answer these online queries with perfect soundness (i.e. with distribution identical to sampling a string from μ_b) with the following guarantee: if the decision tree that is making the online queries is D, then the probability we press the button is at most $\mathrm{TV}(\mathrm{tran}(D, \mu_0), \mathrm{tran}(D, \mu_1))$ (the total variation distance between the query outputs D receives when run on μ_0 and the query outputs D receives when run on μ_1).

This simulation theorem, though somewhat technical, ends up being stronger than the simulation guarantee used by Gavinsky, Lee, Santha, and Sanyal [GLSS19] to provide their composition result for max-conflict complexity. To get a composition theorem, we need to convert this total variation distance between transcripts into a more natural measure; this can be done via some minimax arguments, and the resulting measure is LR. We note that a similarly structured argument occurred in [BB20b], but the squared-Hellinger distance between the transcripts appeared instead of the total variation distance; in that result, the authors showed that this squared-Hellinger distance between transcripts characterized $\mathsf{R}(g)$, but they failed to construct a randomised algorithm R' for f, instead constructing only a "noisy" randomised algorithm. This gave them the result $\mathsf{R}(f \circ g) = \Omega(\mathsf{noisyR}(f)\mathsf{R}(g))$. In contrast, the total variation distance allows us to get $\mathsf{R}(f)$ on the outside, at the cost of getting only $\mathsf{LR}(g)$ on the inside.

The measure LR is arguably more natural than max-conflict complexity, but the real advantage is that our composition theorem turns out to be the best possible of its type: if $\mathsf{R}(f \circ g) = \Omega(\mathsf{R}(f)\mathsf{M}(g))$ for all partial functions f and g, then $\mathsf{LR}(g) = \Omega(\mathsf{M}(g))$. To show this, we give a characterization of $\mathsf{LR}(g)$ in terms of randomised query complexity: there is a family of partial functions f_m such that for all partial functions g, we have

$$\mathsf{LR}(g) = \Theta\left(\frac{\mathsf{R}(f_m \circ g)}{\mathsf{R}(f_m)}\right),$$

where m is the input size of g. Once we have this, it clearly follows that $\mathsf{R}(f \circ g) = \Omega(\mathsf{R}(f)\mathsf{M}(g))$ implies $\mathsf{LR}(g) = \Omega(\mathsf{M}(g))$. The function family f_m turns out to be the same as the one introduced in [BB20b] (based on a family of relations introduced in [GLSS19]); the randomised query complexity $\mathsf{R}(f_m)$ was already established in that paper, so all we need is an upper bound on $\mathsf{R}(f_m \circ g)$ which uses the existence of an LR-style algorithm for g. The linear dependence on the bias which is built into the definition of $\mathsf{LR}(g)$ turns out to be precisely what is needed to upper bound $\mathsf{R}(f_m \circ g)$ (see Section 5 for details).

Failure of small-bias minimax. Our second result, separation of LR and ULR, is proved in Part II. The function f that witnesses the separation $\mathsf{ULR}(f) \geq \Omega(\mathsf{LR}(f)^{5/4})$ is not hard to define. For simplicity, we denote its input length by $N := Bn$ and think of the input as being composed of $B = n^c$ blocks (for some large constant c) of n bits each. We define f as a composition of $\textsc{Maj}_B$ as an outer function, and $\textsc{Xor}_n$ as an inner function, where we are able to switch individual $\textsc{Xor}$-blocks to be easy (requiring $O(1)$ queries) or hard (requiring n queries). Moreover, we make the following promises about the input. Either

(1) all blocks are easy, and a random block has a value with bias $1/n$ towards the majority value; or
(2) $b := n^{-3/4}$ fraction of the blocks are hard, and a random block has bias $\Omega(b)$ towards the majority.

We claim that this function is easy for LR, namely, $\mathsf{LR}(f) = O(n)$. To see this, consider the algorithm R that chooses a block at random, computes it, and outputs its value. For inputs x of type (1) we have $\mathrm{cost}(R, x) = O(1)$ and $\mathrm{bias}(R, x) \geq 1/n$ so that cost/bias ratio is $O(n)$. For inputs x of type (2) we have $\mathrm{cost}(R, x) = bn + (1 - b)O(1) \leq O(bn)$ and $\mathrm{bias}(R, x) = b$ so that cost/bias ratio is $O(n)$ again.

The difficult part is to show that $\mathsf{ULR}(f) \geq \Omega(n^{5/4})$. For example, the above algorithm R has ULR-style measure $\max_{x,y} \mathrm{cost}(R, x)/\mathrm{bias}(R, y) = O(bn)/(1/n) = O(n^{5/4})$, and we would like to show that this is optimal. Intuitively, it is hard to get large bias for inputs of type (1) (although query cost is small here) and it is hard to get low query cost for inputs of type (2) (although bias is relatively high here). We first argue that an algorithm that wants to keep $\mathrm{cost}(R,x)$ small uniformly for all x (even those x with high $\mathrm{bias}(R,x)$) cannot afford to solve hard blocks very often. This is formalised by picking an appropriate pair of hard distributions for f according to the minimax formulation (4). What remains is the following task: Show that any algorithm that does not solve hard blocks, has large cost/bias ratio relative to a *single* hard distribution, that is, show an LR-style lower bound.

To this end, we develop a suite of techniques to prove lower bounds on the cost/bias trade-off achievable by decision trees in the small-bias expected cost setting, which has not really been studied in the literature before. Consequently, we end up having to re-establish some basic facts in the expected-cost setting that have been long known in the worst-case setting. For example, we show any algorithm for $\textsc{GapMaj}_n$ (with $\sqrt{n}$ gap promise) can achieve bias at most $O(\sqrt{\mathrm{cost}/n})$ (see Section 9). The proof here exploits the "$\textsc{And}$-trick" used by Sherstov [She12] to prove a lower bound on the (worst-case) randomised communication complexity of the gap-Hamming problem. These techniques also come in handy when we separate LR from $\overline{\chi}$ for the proof of Lemma 4.

1.6 Open questions

The foremost open question is to resolve Conjecture 1. We can equivalently formulate it as follows.

Open Problem 1 (Conjecture 1 rephrased)**.** *Does* $\mathsf{LR}(f) = \Theta(\mathsf{R}(f))$ *for all total functions* f*?*

One intriguing open problem regarding our new-found measure LR is to show that it is lower-bounded by quantum query complexity Q. Indeed, the bias of a quantum algorithm can be amplified linearly in the query cost, so it seems sensible to conjecture this is so. However, quantum query complexity has mostly been studied in the worst-case setting, and it is unclear how one should even define quantum query complexity in expectation (in such a way that it supports linear bias amplification).

Open Problem 2. *Does it hold that* $\mathsf{LR}(f) \geq \mathsf{Q}(f)$*?*

There is a second reason to care about this question, having to do with the *composition limit* of randomised algorithms. Define $\mathsf{R}^*(f) := \lim_{k\to\infty} \mathsf{R}(f^{\circ k})^{1/k}$; this is the limit of the k-th root of the randomised query complexity of the k-fold composition of f. Our results here imply that $\mathsf{R}^*(f) \geq \Omega(\mathsf{LR}(f))$ for all (possibly partial) functions f. Due to the composition theorem for quantum query complexity, it is also known that $\mathsf{R}^*(f) \geq \Omega(\mathsf{Q}(f))$. The above open problem asks whether one of these results dominates the other. More generally, it would be nice to characterize $\mathsf{R}^*(f)$ in terms of a simpler measure (for instance, one which is efficiently computable given the truth table of the function).

Our inner-optimal composition theorem for LR, together with the outer-optimal composition theorem for noisyR [BB20b] give a relatively satisfying picture of composition in the case of partial functions. However, we can still ask whether there remain other *incomparable* composition theorems.

Open Problem 3. *Are there multiplicative composition theorems, stating that* $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{M}_1(f)\mathsf{M}_2(g))$ *for all partial* f, g*, that can sometimes prove better lower bounds than* $\Omega(\max\{\mathsf{R}(f)\mathsf{LR}(g), \mathsf{noisyR}(f)\mathsf{R}(g)\})$*?*

Regarding the failure of the distributational characterization of $\overline{\mathsf{R}}_\epsilon$ in the low bias regime (Theorem 5), one may wonder whether the definition of $\overline{\mathsf{D}}_\epsilon$ should really involve randomized decision trees instead of deterministic ones. As hinted in Section 1.3, while considering deterministic trees is the natural choice in the bounded error regime, we feel it might not be in the regime where $\epsilon \approx 1/2$. Indeed, while a randomised decision tree can get cost arbitrarily close to zero for ϵ approaching $1/2$ (by taking an appropriate mixture with the zero-query tree), a deterministic one will get stuck at making one query and thus cost 1. Deciding whether the two versions are equivalent (up to constant factors and additive terms) is our last open question.

Open Problem 4. *Let* $\overline{\mathsf{D}}^\star_\epsilon(f) := \max_\mu \min_{D\in \mathsf{D}(f,\epsilon,\mu)} \mathrm{cost}(D,\mu)$ *where* $\mathsf{D}(f,\epsilon,\mu)$ *is the set of all deterministic decision trees solving* f *with error at most* ϵ *relative to inputs sampled from* μ*. For any partial* f *and* ϵ*, do we have* $\overline{\mathsf{D}}^\star_\epsilon(f) \leq O(\overline{\mathsf{D}}_\epsilon(f) + 1)$*?*

2 Preliminaries

2.1 Query complexity notation

Fix a natural number $n \in \mathbb{N}$. A total boolean function is a function $f\colon \{0,1\}^n \to \{0,1\}$. We will consider several generalizations of total boolean functions: first, there are *partial* boolean functions, which are defined on a domain which is a subset of $\{0,1\}^n$. We use $\mathrm{Dom}(f) \subseteq \{0,1\}^n$ to denote the domain of such a function. A further way to generalize boolean functions is to expand the input and output alphabets; that is, for finite sets Σ_I and Σ_O, we can consider functions $f\colon \mathrm{Dom}(f) \to \Sigma_O$ with $\mathrm{Dom}(f) \subseteq \Sigma_I^n$, which take in input strings over the alphabet Σ_I and output a symbol in Σ_O.

A still further way to generalize such functions is to consider *relations* instead of partial functions. A relation is a subset of $\Sigma_I^n \times \Sigma_O$, or alternatively, it is a function that maps Σ_I^n to a subset of Σ_O. Any partial function can be viewed as a (total) relation, where on an input x which is not in the domain of the partial function, the corresponding relation relates all output symbols to x (meaning that if x is the input, any output symbol is considered valid).

Given a boolean function f (or, more generally, a relation), we will denote its *deterministic query complexity* by $\mathsf{D}(f)$. This is the minimum height of a *decision tree* D which correctly computes $f(x)$ on any $x \in \mathrm{Dom}(f)$; in other words, it is the minimum number of worst-case adaptive queries required by a deterministic algorithm computing f. For a formal definition, see [BdW02].

In this work we will mostly be dealing with randomised algorithms rather than deterministic ones, so let us more carefully define those. A *randomised query algorithm* or *randomised decision tree* will be a probability distribution over deterministic decision trees. Such deterministic decision trees will have internal nodes labeled by $[n] \coloneqq \{1,2,\dots,n\}$ (representing the index of the input to query), arcs labeled by Σ_I (representing the symbol we might see after querying an index), and leaves labeled by Σ_O (representing output symbols to return at the end of the algorithm). We will assume that no internal node shares a label with an ancestor, meaning that a deterministic algorithm does not query the same index twice.

For such a randomised algorithm R and for an input $x \in \Sigma_I^n$, we denote by $R(x)$ the random variable we get by sampling a deterministic tree D from R, and returning $D(x)$ (the label of the leaf of D reached after starting from the root and taking the path determined by x). For a function f, we write $\mathrm{err}_f(R,x) \coloneqq \Pr_R[R(x) \neq f(x)]$ (or $\Pr_R[R(x) \notin f(x)]$ if f is a relation), and we write $\mathrm{bias}^{\pm}_f(R,x) \coloneqq 1 - 2\,\mathrm{err}_f(R,x)$, $\mathrm{bias}_f(R,x) \coloneqq \max\{\mathrm{bias}^{\pm}_f(R,x), 0\}$; we omit the subscript f when it is clear from context.

For a deterministic tree D, let $\mathrm{cost}(D,x)$ be the number of queries D makes on input x; this is the height of the leaf of D that is reached when D is run on x. For a randomised algorithm R, we then define $\mathrm{cost}(R,x) \coloneqq \mathbb{E}_{D\sim R}[\mathrm{cost}(D,x)]$ (this is the expected number of queries R makes when run on x).

We extend both of the above to distributions μ over Σ_I^n instead of just inputs x; that is, define

$$\mathrm{bias}^{\pm}_f(R,\mu) \coloneqq \mathbb{E}_{x\sim\mu}[\mathrm{bias}^{\pm}_f(R,x)] = \mathbb{E}_{x\sim\mu}\mathbb{E}_{D\sim R}[\mathrm{bias}^{\pm}_f(D,x)],$$

$$\mathrm{cost}(R,\mu) \coloneqq \mathbb{E}_{x\sim\mu}[\mathrm{cost}(R,x)] = \mathbb{E}_{x\sim\mu}\mathbb{E}_{D\sim R}[\mathrm{cost}(D,x)],$$

with $\mathrm{bias}_f(R,\mu) \coloneqq \max\{\mathrm{bias}^{\pm}_f(R,\mu), 0\}$. We also define $\mathrm{tran}(R,\mu)$ to be the random variable we get by sampling a decision tree D from R, a string x from μ, and returning the pair (D,ℓ), where ℓ is the leaf of D reached when D is run on x. Intuitively, $\mathrm{tran}(R,\mu)$ is the "transcript" when R is run on an input sampled from μ, and such a transcript records all information that an agent running R knows about the input x at the end of the algorithm. We will use $\mathrm{TV}(\mu,\nu) \coloneqq \frac{1}{2}\sum_{x\in\mathcal{X}} |\mu[x] - \nu[x]|$ to denote the total variation distance between distributions μ and ν over set $\mathcal{X}$. Most often, we will employ it with respect to the transcript of R on two different distributions as a way to quantify the extent to which R can tell these distributions apart.

We say that a randomised algorithm R computes f to error ϵ if $\mathrm{err}_f(R,x) \le \epsilon$ for all $x \in \mathrm{Dom}(f)$. We then let $\overline{\mathsf{R}}_\epsilon(f)$ the minimum possible value of $\max_x \mathrm{cost}(R,x)$ over randomised algorithms R satisfying $\mathrm{err}_f(R,x) \le \epsilon$ for all $x \in \mathrm{Dom}(f)$. We also use $\mathsf{R}_\epsilon(f)$ to denote the minimum number T such that there is a randomised algorithm R with $\mathrm{err}_f(R,x) \le \epsilon$ for all $x \in \mathrm{Dom}(f)$ such that all decision trees in the support of R have height at most T. The difference between $\mathsf{R}_\epsilon(f)$ and $\overline{\mathsf{R}}_\epsilon(f)$ is that the former measures the worst-case cost of an algorithm computing f to error ϵ (maximizing over both the input string and the internal randomness), while the latter measures the expected worst-case cost of the algorithm computing f to error ϵ (this still maximizes over the input strings x, but takes an expectation over the internal randomness of the algorithm R).

It is easy to see that $\overline{\mathsf{R}}_\epsilon(f) \le \mathsf{R}_\epsilon(f)$ for all f. The other direction also holds if we tolerate a constant-factor loss, as well as an additive constant loss in ϵ; to see this, note that if we cut off the $\overline{\mathsf{R}}_\epsilon(f)$ algorithm after it makes 10 times more queries than it is expected to, then the probability of reaching such a cutoff is at most $1/10$ by Markov's inequality, and hence the error probability of the algorithm increases by at most $1/10$; this converts an $\overline{\mathsf{R}}_\epsilon(f)$ algorithm into a $\mathsf{R}_\epsilon(f)$ algorithm.

Standard error reduction techniques imply that for a boolean function f, $\mathsf{R}_\epsilon(f)$ is related to $\mathsf{R}_{\epsilon'}(f)$ by a constant factor that depends only on ϵ and ϵ', so long as both are in $(0, 1/2)$. For this reason, the value of ϵ does not matter when ϵ is a constant in $(0, 1/2)$ (so long as we ignore constant factors and so long as the function is boolean), so we omit ϵ when $\epsilon = 1/3$. The same error reduction property holds for $\overline{\mathsf{R}}_\epsilon(f)$. Combined with the Markov inequality argument above, both $\mathsf{R}(f)$ and $\overline{\mathsf{R}}(f)$ are the same measure (up to constant factors) for a boolean function and for constant values of ϵ.

We warn that these equivalences break if f is not boolean (especially if f is a relation) or if the value of ϵ is not constant; in particular, when $\epsilon = 1/n$ or when $\epsilon = 1/2 - 1/n$, the values of $\mathsf{R}_\epsilon(f)$ and $\overline{\mathsf{R}}_\epsilon(f)$ may differ by more than a constant factor.

2.2 Linearised R

For a (possibly partial) boolean function f on n bits, we define

$$\mathsf{LR}(f) \coloneqq \min_R \max_x \frac{\operatorname{cost}(R,x)}{\operatorname{bias}(R,x)}.$$

Here R ranges over randomised decision trees and x ranges over the domain of f, and we treat $0/0$ as ∞.

We call this measure *linearised randomised query complexity*. The name comes from the linear dependence on the bias achieved by the algorithm. Note that if we wanted to amplify bias γ to constant bias, we would, in general, have to repeat the algorithm $\Theta(1/\gamma^2)$ times to do so. In some sense, then, the measure $\mathsf{R}(f)$ charges $1/\gamma^2$ for an algorithm that achieves bias γ instead of achieving constant bias. The measure $\mathsf{LR}(f)$, in contrast, charges only $1/\gamma$ for such an algorithm, so it can be up to quadratically smaller than $\mathsf{R}(f)$.

A minimax theorem for ratios such as [BB20a] (Theorem 2.18) can show that

$$\mathsf{LR}(f) = \max_\mu \min_D \frac{\operatorname{cost}(D,\mu)}{\operatorname{bias}(D,\mu)}, \tag{7}$$

where D ranges over deterministic decision trees and μ ranges over probability distributions over $\operatorname{Dom}(f)$.

It is not hard to see that the maximizing distribution μ above will place equal weight on 0 and 1 inputs. This is because otherwise, we could take D to be a decision tree that makes 0 queries, and then $\operatorname{cost}(D,\mu)$ would be 0 while $\operatorname{bias}(D,\mu)$ would be positive.

If μ is balanced over 0 and 1 inputs, we may express it as $\mu \coloneqq \mu_0/2 + \mu_1/2$ and it is not hard to show that for the best possible choice of leaf labels for an unlabeled decision tree D, we have

$$\operatorname{bias}(D,\mu)^{\pm} = \operatorname{bias}(D,\mu) = \operatorname{TV}(\operatorname{tran}(D,\mu_0), \operatorname{tran}(D,\mu_1)). \tag{8}$$

This follows, for example, from [BB20a] (Lemma 3.9); to see this intuitively, recall that $\operatorname{tran}(D,\mu)$ is the random variable for the leaf of D reached when D is run on μ, and note that the best choice of leaf label if D reaches a leaf ℓ is 0 if the probability of D reaching ℓ is higher when run on μ_0 than on μ_1, and it is 1 otherwise. Therefore, the bias for the best choice of leaf labels is the sum, over leaves ℓ of D, of $2\max\{\Pr_{\mu_0}[\ell], \Pr_{\mu_1}[\ell]\} - 1$, which is easily seen to be the total variation distance between the two distributions over leaves.

Given (8), we can also write

$$\mathsf{LR}(f) = \max_{\mu_0,\mu_1} \min_D \frac{\operatorname{cost}(D, \frac{\mu_0+\mu_1}{2})}{\operatorname{TV}(\operatorname{tran}(D,\mu_0), \operatorname{tran}(D,\mu_1))},$$

where μ_0 ranges over probability distributions with support $f^{-1}(0)$ and μ_1 ranges over probability distributions with support $f^{-1}(1)$. Observe that neither the top nor the bottom depend on the leaf labels of D, so we can now assume D is an unlabeled decision tree if we wish. Note also that $\mathrm{cost}(D,\mu)$ is linear in the second argument, so we can write

$$\mathsf{LR}(f) = \max_{\mu_0,\mu_1} \min_D \frac{\mathrm{cost}(D,\mu_0)+\mathrm{cost}(D,\mu_1)}{2\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))}.$$

We clearly have

$$\mathsf{LR}(f) \geq \max_{\mu_0,\mu_1} \min_D \frac{\min\{\mathrm{cost}(D,\mu_0),\mathrm{cost}(D,\mu_1)\}}{\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))}.$$

Lemma 7. *For any fixed μ_0 and μ_1, we have*

$$\min_D \frac{\mathrm{cost}(D,\mu_1)}{\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))} \leq 6 \min_D \frac{\mathrm{cost}(D,\mu_0)}{\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))}.$$

Proof. Let D minimize the right-hand side. Suppose by contradiction that D makes a lot more queries against μ_1 than it does against μ_0. The idea is to cut off paths in D that are too long, since in those paths we essentially already know we are running on μ_1 instead of μ_0. The new truncated tree D' will still have roughly the same total variation distance between the transcripts when run on μ_0 and μ_1, but it won't make too many more queries on μ_1 than D made on μ_0.

Formally, we define D' to be the same tree D except that we cut off an internal node u of D (making it a leaf in D') if $\Pr_{\mu_1}[u]/\Pr_{\mu_0}[u] > 3$. The new tree D' makes fewer queries on every input than D did, so we clearly have $\mathrm{cost}(D',\mu_b) \leq \mathrm{cost}(D,\mu_b)$ for $b = 0, 1$. Moreover, note that $\mathrm{cost}(D,\mu)$ is the sum, over all internal nodes of D, of the probability that D reaches that node when run on μ. Now, for all internal nodes of D', we know that $\Pr_{\mu_1}[u] \leq 3\Pr_{\mu_0}[u]$, and hence we have $\mathrm{cost}(D',\mu_1) \leq 3\,\mathrm{cost}(D',\mu_0) \leq 3\,\mathrm{cost}(D,\mu_0)$.

We next want to show that the distance $\mathrm{TV}(\mathrm{tran}(D',\mu_0),\mathrm{tran}(D',\mu_1))$ is not much smaller than the distance $\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))$. To see this, let V be the set of all leaves in D' that are also leaves in D, and let $*$ denote the event that a cutoff occurred in D'. Then

$$\begin{aligned}
\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1)) &= \frac{1}{2}\sum_v |\Pr_{D,\mu_0}[v] - \Pr_{D,\mu_1}[v]| \\
&\leq \frac{1}{2}\sum_{v\in V} |\Pr_{D,\mu_0}[v] - \Pr_{D,\mu_1}[v]| + \frac{1}{2}\sum_{v\notin V} \max\{\Pr_{D,\mu_0}[v], \Pr_{D,\mu_1}[v]\} \\
&\leq \frac{1}{2}\sum_{v\in V} |\Pr_{D,\mu_0}[v] - \Pr_{D,\mu_1}[v]| + \frac{\Pr_{D,\mu_0}[*]+\Pr_{D,\mu_1}[*]}{2},
\end{aligned}$$

while

$$\begin{aligned}
\mathrm{TV}(\mathrm{tran}(D',\mu_0),\mathrm{tran}(D',\mu_1)) &= \frac{1}{2}\sum_v |\Pr_{D',\mu_0}[v] - \Pr_{D',\mu_1}[v]| \\
&= \frac{1}{2}\sum_{v\in V} |\Pr_{D,\mu_0}[v] - \Pr_{D,\mu_1}[v]| + \frac{1}{2}\sum_{v\notin V} \frac{\Pr_{D,\mu_1}[v]-\Pr_{D,\mu_0}[v]}{\Pr_{D,\mu_1}[v]+\Pr_{D,\mu_0}[v]} \cdot (\Pr_{D,\mu_1}[v] + \Pr_{D,\mu_0}[v]) \\
&\geq \frac{1}{2}\sum_{v\in V} |\Pr_{D,\mu_0}[v] - \Pr_{D,\mu_1}[v]| + \frac{1}{4}\sum_{v\notin V} \Pr_{D,\mu_1}[v] + \Pr_{D,\mu_0}[v] \\
&= \frac{1}{2}\sum_{v\in V} |\Pr_{D,\mu_0}[v] - \Pr_{D,\mu_1}[v]| + \frac{\Pr_{D,\mu_0}[*]+\Pr_{D,\mu_1}[*]}{4}.
\end{aligned}$$

Hence the worst possible ratio between them is $1/2$, and the greatest possible ratio between the left-hand side and right-hand side of the original lemma is 6, completing the proof. □

Corollary 8.

$$\max_{\mu_0,\mu_1} \min_D \frac{\min\{\mathrm{cost}(D,\mu_0),\mathrm{cost}(D,\mu_1)\}}{\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))} \leq \mathsf{LR}(f) \leq 6 \max_{\mu_0,\mu_1} \min_D \frac{\min\{\mathrm{cost}(D,\mu_0),\mathrm{cost}(D,\mu_1)\}}{\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))}.$$

One useful property of LR complexity is that up to a multiplicative factor of 2, we can consider only randomised decision trees that always query at least one bit of their input.

Lemma 9. *For every non-constant partial function g, there is a randomised decision tree A that always queries at least one bit of g's input and satisfies, for every x,*

$$\frac{\mathrm{cost}(A,x)}{\mathrm{bias}(A,x)} \le 2 \cdot \mathsf{LR}(g).$$

Proof. Let R be a randomised tree that achieves the minimum for $\mathsf{LR}(g)$. Write $R = \lambda_0 T_0 + \lambda_1 T_1 + \lambda_2 R_{\ge 1}$, $\lambda_0 + \lambda_1 + \lambda_2 = 1$, $\lambda_i \ge 0$, where T_i is the cost-0 deterministic tree that makes no queries and outputs i, and where $R_{\ge 1}$ is a randomised tree that always makes at least 1 query. Note that $\lambda_0, \lambda_1 < 1/2$ as otherwise the tree would answer incorrectly with probability $\ge 1/2$ on some input. Let us assume wlog that $\lambda_0 \le \lambda_1$. Re-weight R by defining a new randomised tree $R' \coloneqq \lambda_1' T_1 + \lambda_2' R_{\ge 1}$ where $\lambda_1' \coloneqq (\lambda_1 - \lambda_0)/(1 - 2\lambda_0)$ and $\lambda_2' \coloneqq \lambda_2/(1-2\lambda_0)$. Then, for all x, using $\mathrm{cost}((T_0+T_1)/2, x) = \mathrm{bias}((T_0+T_1)/2, x) = 0$,

$$\frac{\mathrm{cost}(R,x)}{\mathrm{bias}(R,x)} = \frac{2\lambda_0 \,\mathrm{cost}((T_0+T_1)/2,x) + (1-2\lambda_0)\,\mathrm{cost}(\lambda_1' T_1 + \lambda_2' R_{\ge 1}, x)}{2\lambda_0 \mathrm{bias}((T_0+T_1)/2,x) + (1-2\lambda_0)\mathrm{bias}(\lambda_1' T_1 + \lambda_2' R_{\ge 1}, x)} = \frac{\mathrm{cost}(R',x)}{\mathrm{bias}(R',x)}.$$

Note that $\lambda_1' < 1/2$ and thus $\mathrm{cost}(R',x) \ge 1/2$. Consider finally the tree $A \coloneqq \lambda_1' T_1' + \lambda_2' R_{\ge 1}$ where T_1' is the cost-1 tree that makes one (arbitrary) query and then outputs 1. We have $\mathrm{cost}(A,x)/\mathrm{bias}(A,x) \le (\mathrm{cost}(R',x) + 1/2)/\mathrm{bias}(R',x) \le 2\,\mathrm{cost}(R',x)/\mathrm{bias}(R',x) = 2 \cdot \mathsf{LR}(g)$. □

As a corollary, we obtain a universal lower bound on the LR complexity of every non-constant function.

Corollary 10. *For every non-constant partial function g, $\mathsf{LR}(g) \ge \frac{1}{2}$.*

Proof. By Lemma 9, there exists a randomised decision tree A that always queries at least one bit of its input and satisfies $\mathrm{cost}(A,x)/\mathrm{bias}(A,x) \le 2 \cdot \mathsf{LR}(g)$ for all x in the domain of g. But since A always makes at least one query, $\mathrm{cost}(A,x) \ge 1$. And by definition, $\mathrm{bias}(A,x) \le 1$, so the cost-bias ratio of A is always bounded below by 1. □

Part I

Composition Theorem

In this Part I, we prove our inner-optimal composition theorem, Theorems 1 and 2 restated below, along with related results.

Theorem 1. $\mathsf{R}(f \circ g) \ge \Omega(\mathsf{R}(f)\mathsf{LR}(g))$ *for all partial boolean functions f, g.*

Theorem 2. *Theorem 1 is optimal: If M is any complexity measure such that $\mathsf{R}(f \circ g) \ge \Omega(\mathsf{R}(f)\mathsf{M}(g))$ for all partial f, g, then $\mathsf{LR}(g) \ge \Omega(\mathsf{M}(g))$ for all partial g.*

The heart of the proof of Theorem 1 is a simulation theorem showing that for any two distributions μ_0 and μ_1 and any decision tree T, it is possible to simulate T on inputs drawn from μ_b for some initially unknown $b \in \{0,1\}$ while querying the actual value of b with probability bounded by the total variation distance between the two distributions μ_0 and μ_1. This result, Theorem 11, is established in Section 3.

In Section 4, we use the simulation theorem to complete the proof of the main composition theorem, Theorem 13, a slightly more general version of Theorem 1. We also use the simulation theorem to establish the perfect composition for LR complexity, Theorem 3, in this section.

The proof of Theorem 2 is completed in Section 5. Finally, in Section 6, we establish the separation between LR complexity and max-conflict complexity of Lemma 4.

3 Decision tree simulation theorem

An *online decision tree simulator* is a randomised algorithm that is given two distributions μ_0 and μ_1 on inputs $\{0,1\}^n$, oracle access to a bit $b \in \{0,1\}$, and a stream of queries $i_1, \dots, i_k \in [n]$ that represent the queries made by a decision tree T that is not known to the algorithm. The goal of an online decision tree simulator is to answer the queries according to the distribution μ_b while querying the value of b itself with as small probability as possible. We think of this protocol as having a big red button that gives b, and it tries to pretend to have a sample from μ_b without pressing the button for as long as possible.

Theorem 11. *There exists an online decision tree simulator that simulates the queries of T on μ_b while querying the value of b with probability* $\mathrm{TV}\big(\mathrm{tran}(T,\mu_0), \mathrm{tran}(T,\mu_1)\big)$.

The algorithm that satisfies the theorem is stated below. In the algorithm, $x \in \{0,*,1\}^n$ is a partially defined boolean string: the coordinates labelled with $*$ are undefined. Given a string $x \in \{0,*,1\}^n$, an index $i \in [n]$, and a value $a \in \{0,1\}$, the notation $x^{(i\leftarrow a)}$ denotes the string y which equals x on all coordinates except i, where it takes the value $y_i = a$.

```
Algorithm 1: OnlineQuerySimulator($\mu_0, \mu_1$)
for all $x \in \{0,*,1\}^n$ do
    $\mu_{\min}(x) \leftarrow \min\{\mu_0(x), \mu_1(x)\}$;
$x \leftarrow *^n$;
$b \leftarrow *$;
while more queries remain do
    $i \leftarrow$ NextQuery;
    $u \leftarrow \mu_{\min}(x^{(i\leftarrow 0)}) + \mu_{\min}(x^{(i\leftarrow 1)})$;
    if $b = *$ then
        With probability $1 - u/\mu_{\min}(x)$, query the value of $b$;
    if $b = *$ then
        $x_i \leftarrow \mathrm{Ber}\big(\mu_{\min}(x^{(i\leftarrow 1)})/u\big)$;
    else
        $x_i \leftarrow \mathrm{Ber}\left(\frac{\mu_b(x^{(i\leftarrow 1)}) - \mu_{\min}(x^{(i\leftarrow 1)})}{\mu_b(x) - u}\right)$;
```

Note that each vertex in a decision tree T corresponds to the partial string $x \in \{0,1,*\}^n$ of the values revealed on the path to that vertex in T. Our main task is to show that each vertex in T (including each leaf) is reached with probability $\mu_b(x)$ in the algorithm and that the probability that we reach x *and* don't reveal b along the way is $\mu_{\min}(x)$.

Lemma 12. *For every $x \in \{0,1,*\}^n$, when we run the* OnlineQuerySimulator, *then*

1. *We reach the vertex x with probability $\mu_b(x)$, and*
2. *We reach the vertex x* and *don't query the value b on the way to x with probability $\mu_{\min}(x)$.*

Proof. We prove the claim by induction on the number of defined coordinates on x. The base case corresponds to $x = *^n$, which trivially satisfies both conditions of the claim.

Consider now any $x \neq *^n$. Let z be the parent of x in the decision tree T, and let i denote the coordinate where $z_i = *$ and $x_i \neq *$. Define also y to be x's sibling in T. Let us assume that $x_i = 1$. (The case where $x_i = 0$ is essentially identical.)

By the induction hypothesis, the probability that we reach z and don't query the value b is $\mu_{\min}(z)$. With probability $\big(\mu_{\min}(x) + \mu_{\min}(y)\big)/\mu_{\min}(z)$, we don't query the value of b while processing the query i either. And when this occurs the algorithm next reaches x with probability $\mu_{\min}(x)/\big(\mu_{\min}(x) + \mu_{\min}(y)\big)$. So the overall probability that we reach x without querying b along the way is

$$\mu_{\min}(z) \cdot \frac{\mu_{\min}(x) + \mu_{\min}(y)}{\mu_{\min}(z)} \cdot \frac{\mu_{\min}(x)}{\mu_{\min}(x) + \mu_{\min}(y)} = \mu_{\min}(x).$$

Next, by the induction hypothesis again the probability that we query the value of b either on the way to z or while processing the query i is

$$(\mu_b(z) - \mu_{\min}(z)) + \mu_{\min}(z) \cdot \left(1 - \frac{\mu_{\min}(x) + \mu_{\min}(y)}{\mu_{\min}(z)}\right) = \mu_b(z) - \big(\mu_{\min}(x) + \mu_{\min}(y)\big).$$

Then the probability we output x conditioned on having revealed b is

$$\frac{\mu_b(x) - \mu_{\min}(x)}{\mu_b(z) - (\mu_{\min}(x) + \mu_{\min}(y))},$$

so that the overall probability that we reach x and reveal b along the way is $\mu_b(x) - \mu_{\min}(x)$. Therefore, the overall probability that we reach x is $\mu_b(x)$. □

The proof of Theorem 11 is now essentially complete, as it just requires combining the lemma with a simple identity on total variation distance.

Proof of Theorem 11. Lemma 12 implies that the OracleQuerySimulator indeed reaches each leaf with the correct probability $\mu_b(x)$. And the probability that it queries the value of b is $1 - \sum_{\ell \in T} \min\{\mu_0(\ell), \mu_1(\ell)\}$, which is the total variation distance between $\operatorname{tran}(T, \mu_0)$ and $\operatorname{tran}(T, \mu_1)$. □

4 Composition theorems

The inner-optimal composition theorem, Theorem 1, is established in Section 4.1. In fact, we establish a slight generalization of that theorem, stated below in Theorem 13. Then the perfect composition theorem for LR complexity, Theorem 3, is established in Section 4.2.

4.1 Composition for randomised query complexity

For a boolean string $y \in \{0,1\}^n$ and a pair of distributions μ_0, μ_1, we define $y \circ (\mu_0, \mu_1)$ to be the product distribution $\bigotimes_{i=1}^n \mu_{y_i}$. In particular, if μ_0 and μ_1 are hard distributions for the 0- and 1-inputs of g respectively, and if y is an input to f, then $y \circ (\mu_0, \mu_1)$ will give a distribution over the inputs to the composed $f \circ g$ (all of which correspond to the same f-input y).

We prove the following composition theorem, which is a slightly more general version of Theorem 1.

Theorem 13. *Let Σ_I and Σ_O be finite alphabets, and let $n, m \in \mathbb{N}$. Let $f \subseteq \{0,1\}^n \times \Sigma_O$ be a (possibly partial) relation on n bits, and let $g\colon \operatorname{Dom}(g) \to \{0,1\}$ be a (possibly partial) boolean function, with $\operatorname{Dom}(g) \subseteq \Sigma_I^m$. Let $\epsilon \in [0,1/2)$. Then*

$$\overline{\mathsf{R}}_\epsilon(f \circ g) \geq \overline{\mathsf{R}}_\epsilon(f)\mathsf{LR}(g)/6.$$

Proof. Let μ_0 and μ_1 be distributions over the 0-inputs and 1-inputs to g, respectively, that maximize the expression in the right-hand side of Corollary 8. Let Π be the online decision tree simulator from Theorem 11. Let R be a randomised algorithm that computes $f \circ g$ to error ϵ using $\overline{\mathsf{R}}_\epsilon(f \circ g)$ expected queries. We describe a randomised algorithm R' for computing f on worst-case inputs.

Given input $y \in \{0,1\}^n$, the algorithm R' will instantiate n copies of Π, which we denote $\Pi_1, \Pi_2, \dots, \Pi_n$, one for each bit of the input; if protocol Π_i presses the button, it gets y_i (and this causes R' to make a real query to the real input). Each of these copies of Π will assume the distributions to be simulated are μ_0 and μ_1. Then R' will run R, and whenever R makes a query (i, j) (corresponding to querying bit j inside of the i-th copy of g), the algorithm R' will ask Π_i to give an answer to query j, and it will use that answer to determine the next query of R.

Note that since the protocols Π_i are guaranteed to be sound, the outcome of the simulation of R made by R' is precisely the same (in distribution) as the outcome of running R on an input sampled from $y \circ (\mu_0, \mu_1)$. Therefore, by the correctness guarantee of R, the output will be a valid output for $f(y)$ except with error probability ϵ. It remains to show that for each $y \in \operatorname{Dom}(f)$, the expected number of real queries R' makes when run on y is at most $6\overline{\mathsf{R}}_\epsilon(f \circ g)/\mathsf{LR}(g)$.

Fix any $y \in \mathrm{Dom}(f)$. Now, when R' is run on y, let T be the expected number of fake queries it makes; in other words, let $T = \mathrm{cost}(R, y \circ (\mu_0, \mu_1)) \leq \overline{\mathsf{R}}_\epsilon(f \circ g)$. For each i, let T_i be the expected number of queries to Π_i that R' makes when run on y, so that $T_1 + T_2 + \cdots + T_n = T$. Let p_i the overall probability that Π_i presses the button when R' runs on y; the sum $q = p_1 + p_2 + \cdots + p_n$ is therefore the expected number of real queries made by R' on y. We would like to show that $q \leq 6T/\mathsf{LR}(g)$, or equivalently, $T/q \geq \mathsf{LR}(g)/6$.

Since $T/q = (T_1 + \cdots + T_n)/(p_1 + \cdots + p_n)$, there must be some i such that $T/q \geq T_i/p_i$. It will therefore suffice to show that $T_i/p_i \geq \mathsf{LR}(g)/6$ for all $i \in [n]$. Fix such i, and recall that T_i is the number of (fake) queries R' makes to Π_i when run on y, and p_i is the probability that Π_i presses the button when R' is run on y. Consider the algorithm $R_{y,i}$ which takes in an input x in $\mathrm{Dom}(g)$, generates $n-1$ additional fake inputs to g from the distributions μ_{y_ℓ} for $\ell \neq i$, places the real input x as the i-th input among the n inputs to g, and runs R on this tuple (treating it as an input to $f \circ g$). Note that when $R_{y,i}$ is run on an input from μ_{y_i}, its behavior is exactly the same as the behavior of R when run on $y \circ (\mu_0, \mu_1)$; therefore, it makes T_i expected queries. Consider running $R_{y,i}$ with query answers generated by Π instead of by making real queries; then when Π uses the hidden bit y_i and simulates the distributions μ_0, μ_1, the behavior of $R_{y,i}$ is the same as when we run it on μ_{y_i}, and hence the expected number of queries it makes to Π is T_i and the probability that Π presses the button is exactly p_i.

Now, by Theorem 11, we know that Π presses the button with probability $\mathrm{TV}(\mathrm{tran}(D, \mu_0), \mathrm{tran}(D, \mu_1))$ when simulating a deterministic decision tree D. For a random decision tree such as the one given by $R_{y,i}$, the probability p_i of the button being pressed will be the mixture of the values $\mathrm{TV}(\mathrm{tran}(D, \mu_0), \mathrm{tran}(D, \mu_1))$ for the deterministic decision trees D in the support of $R_{y,i}$. Also, the expected number of queries T_i that $R_{y,i}$ makes is a matching mixture of the expected number of queries made by the decision trees D in the support of $R_{y,i}$; the latter is $\mathrm{cost}(D, \mu_{y_i})$. Hence to lower bound T_i/p_i, it will suffice to lower bound $\frac{\mathrm{cost}(D,\mu_{y_i})}{\mathrm{TV}(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))}$ for all deterministic decision trees D acting on inputs in $\mathrm{Dom}(g)$. We now write

$$\frac{\mathrm{cost}(D, \mu_{y_i})}{\mathrm{TV}(\mathrm{tran}(D, \mu_0), \mathrm{tran}(D, \mu_1))} \geq \frac{\min\{\mathrm{cost}(D, \mu_0), \mathrm{cost}(D, \mu_1)\}}{\mathrm{TV}(\mathrm{tran}(D, \mu_0), \mathrm{tran}(D, \mu_1))} \geq \mathsf{LR}(g)/6$$

(using Corollary 8). The desired result follows. □

4.2 Composition for LR complexity

Theorem 3. $\mathsf{LR}(f \circ g) \geq \Omega(\mathsf{LR}(f)\mathsf{LR}(g))$ *for all partial boolean functions* f, g.

We actually prove the more explicit result $\mathsf{LR}(f \circ g) \geq \mathsf{LR}(f)\mathsf{LR}(g)/6$.

Proof. The proof is similar to that of Theorem 13. We fix hard distributions μ_0 and μ_1 for $\mathsf{LR}(g)$, and we fix a randomised algorithm R for $f \circ g$ such that $\max_z \mathrm{cost}(R, z)/\mathrm{bias}(R, z) \leq \mathsf{LR}(f \circ g)$. We then define a randomised algorithm R' for f; this time, unlike in the proof of Theorem 13, we want R' to solve f in the $\mathsf{LR}(f)$ sense instead of being a randomised algorithm that solves f to error ϵ. We define R' as before: on input $y \in \mathrm{Dom}(f)$, R' instantiates n protocols Π_i, one for each bit of y; it instantiates each with the distributions (μ_0, μ_1), and gives Π_i the hidden bit y_i if it presses the button. Then R' will run R, and whenever R makes a query (i, j) (to the bit j inside the i-th input to g), R' will ask Π_i for bit j.

Note that by the soundness of the protocols Π_i, we have $\mathrm{bias}(R', y) = \mathrm{bias}(R, y \circ (\mu_0, \mu_1))$. We will next show that $\mathrm{cost}(R', y) \leq 6\,\mathrm{cost}(R, y \circ (\mu_0, \mu_1))/\mathsf{LR}(g)$; This way, we will have

$$\mathsf{LR}(f) = \max_y \frac{\mathrm{cost}(R', y)}{\mathrm{bias}(R', y)} \leq \frac{6}{\mathsf{LR}(g)} \max_y \frac{\mathrm{cost}(R, y \circ (\mu_0, \mu_1))}{\mathrm{bias}(R, y \circ (\mu_0, \mu_1))} \leq \frac{6}{\mathsf{LR}(g)} \max_z \frac{\mathrm{cost}(R, z)}{\mathrm{bias}(R, z)} \leq \frac{6\mathsf{LR}(f \circ g)}{\mathsf{LR}(g)}.$$

Fix any $y \in \mathrm{Dom}(f)$; it remains to show that $\mathrm{cost}(R, y \circ (\mu_0, \mu_1))/\mathrm{cost}(R', y) \geq \mathsf{LR}(g)/6$. For every $i \in [n]$, let T_i be the expected number of queries R makes to the i-th input on $y \circ (\mu_0, \mu_1)$, and let p_i be the probability that R' queries the i-th bit when run on input y. Let $T = T_1 + \cdots + T_n$, and let $q = p_1 + \cdots + p_n$. We wish to show $T/q \geq \mathsf{LR}(g)/6$. This precise statement was shown in the proof of Theorem 13, which completes this proof as well. □

5 Optimality of the composition theorem

We complete the proof of Theorem 2 in this section.

Theorem 2. *Theorem 1 is optimal: If* M *is any complexity measure such that* $\mathsf{R}(f \circ g) \geq \Omega(\mathsf{R}(f)\mathsf{M}(g))$ *for all partial* f, g*, then* $\mathsf{LR}(g) \geq \Omega(\mathsf{M}(g))$ *for all partial* g*.*

The proof of Theorem 2 is obtained by a characterization of LR complexity in terms of the complexity of functions composed with the *approximate index* partial function $\textsc{ApproxIndex}_k : \{0,1\}^k \times \{0,1,2\}^{2^k} \to \{0,1,*\}$ defined by

$$\textsc{ApproxIndex}_k(a,y) = \begin{cases} y_a & \text{if } y_a \in \{0,1\}, \\ & \quad y_b = y_a \text{ for all } |b-a| \leq \frac{k}{2} - 2\sqrt{k \log k}, \text{ and} \\ & \quad y_b = 2 \text{ for all other } b \\ * & \text{otherwise.} \end{cases}$$

The randomised query complexity of the approximate index function is as follows.

Lemma 14 ([BB20b, Lemma 27])**.** $\mathsf{R}(\textsc{ApproxIndex}_k) = \Theta\left(\sqrt{k \log k}\right)$.

The key to the proof of Theorem 2 is the following characterization of LR complexity in terms of composition with the approximate index function.

Lemma 15. *For every partial boolean function* $g : \Sigma^m \to \{0,1,*\}$*, when* $k \in \mathbb{N}$ *satisfies* $\frac{k}{\log k} \geq (36m)^2$ *then*

$$\mathsf{LR}(g) = \Theta\left(\frac{\mathsf{R}(\textsc{ApproxIndex}_k \circ g)}{\mathsf{R}(\textsc{ApproxIndex}_k)}\right).$$

Proof. The lemma trivially holds when g is a constant function. For the rest of the proof, fix g to be any non-constant partial function. Theorem 1 implies the upper bound

$$\mathsf{LR}(g) = O\left(\frac{\mathsf{R}(\textsc{ApproxIndex}_k \circ g)}{\mathsf{R}(\textsc{ApproxIndex}_k)}\right).$$

The goal of the remainder of the proof is to establish a matching lower bound by showing that

$$\overline{\mathsf{R}}(\textsc{ApproxIndex}_k \circ g) = O\left(\sqrt{k \log k} \cdot \mathsf{LR}(g)\right).$$

This bound suffices to complete the proof because $\overline{\mathsf{R}}(f) = \Theta(\mathsf{R}(f))$ for every partial function f.

Let R denote a randomised algorithm that satisfies

$$\mathrm{cost}(R,x) \leq 2 \cdot \mathsf{LR}(g) \cdot \mathrm{bias}(R,x)$$

for all x in the domain of g and always queries at least one bit of its input. Such an algorithm is guaranteed to exist by Lemma 9. We define a new randomised algorithm A that proceeds as follows: it runs the algorithm R sequentially on the first instances $x_1, x_2, \ldots, x_\ell$ of g which correspond to the initial address bits of the input to $\textsc{ApproxIndex}_k$. It continues this process until the total number of queries made to the underlying inputs exceeds $36\sqrt{k \log k} \cdot \mathsf{LR}(g)$. By the choice of k and the trivial bound $\mathsf{LR}(g) \leq m$, this process terminates when R has computed the first ℓ instances of g with some biases $b_1, \ldots, b_\ell$ for some $\ell \leq k$. The algorithm A then guesses the value of the remaining $k - \ell$ bits of the address. It finally computes the value of g on the instance corresponding to the address obtained with error probability at most $\frac{1}{9}$ and returns that value.

Let $c_1, \ldots, c_\ell$ denote the query cost incurred by R when running on the ℓ computed instances of g. The random variables $(X_i)_{i \leq k}$ defined by $X_i = \sum_{j \leq i} c_j - \mathrm{cost}(R, x_j)$ form a discrete-time martingale and ℓ is the stopping time of this martingale. By the optional stopping theorem, $\mathrm{E}[X_\ell] = 0$. So $\mathrm{E}[\sum_{i \leq \ell} c_i] = \sum_{i \leq \ell} \mathrm{cost}(R, x_i)$. By Markov's inequality, the probability that the total cost exceeds 6 times the expected cost on the same inputs is at most $1/6$; let us consider from now on only the case when this does not occur. In this case,

$$\sum_{i=1}^{\ell} \mathrm{cost}(R, x_i) \geq \frac{1}{6}\sum_{i=1}^{\ell} c_i \geq 6\sqrt{k\log k}\cdot \mathsf{LR}(g).$$

By our choice of R, the biases $\beta_1, \ldots, \beta_\ell$ on the values $g(x_1), \ldots, g(x_\ell)$ satisfy $\sum_{i=1}^{\ell} \beta_i \geq 3\sqrt{k\log k}$ and so if we let $b \in \{0,1\}^k$ denote the address computed by the algorithm, we observe that

$$\mathrm{E}\big[|b-a|\big] = \sum_{i=1}^{k} \Pr[b_i \neq g(x_i)] \leq \frac{k}{2} - 3\sqrt{k\log k}.$$

Furthermore, each of the k events $b_i \neq g(x_i)$ are independent. So by Hoeffding's bound the probability that more than $\frac{k}{2} - 2\sqrt{k\log k}$ of these events occur is at most $e^{-2\log^2 k}$, which is less than $\frac{1}{9}$ when $k \geq 3$. When this event does not occur, the address b computed by the algorithm satisfies $x_b = x_a$. Since A lastly computes $g(x_b)$ with error at most $\frac{1}{9}$, in total it computes $\textsc{ApproxIndex}_k \circ g$ with error at most $\frac{1}{3}$.

It remains to show that the expected query cost of the algorithm A satisfies the desired bound. The first round of the algorithm uses at most $36\sqrt{k\log k}\cdot \mathsf{LR}(g)$ queries plus the number of queries of the instance of R run on x_ℓ. In expectation, this additional number of queries is at most $\mathrm{cost}(R, x_\ell) \leq \mathsf{LR}(g)$. And then computing $g(x_b)$ requires another $\mathsf{R}(g) \leq m < \sqrt{k}$ queries. So the overall expected query complexity of A is at most $(36\sqrt{k\log k}+1)\cdot \mathsf{LR}(g) + \sqrt{k}$. By Corollary 10, $\mathsf{LR}(g) \geq \frac{1}{2}$ for every non-constant function g so this query complexity is bounded above by $O\big(\sqrt{k\log k}\cdot \mathsf{LR}(g)\big)$, as required. □

The proof of Theorem 2 now follows easily from Lemma 15.

Proof of Theorem 2. Let M be a measure that satisfies the condition of the theorem. Then, choosing f to be the $\textsc{ApproxIndex}_k$ function for a large enough value of k and applying Lemma 15, we obtain

$$M(g) = O\left(\frac{\mathsf{R}(\textsc{ApproxIndex}_k \circ g)}{\mathsf{R}(\textsc{ApproxIndex}_k)}\right) = O\left(\mathsf{LR}(g)\right). \qquad \square$$

6 Separation from $\overline{\chi}$

In this section, we exhibit a polynomial separation between LR and $\overline{\chi}$, the max conflict complexity introduced by Gavinsky, Lee, Santha and Sanyal in [GLSS19] (see Section 6.1 for a formal definition of $\overline{\chi}$).

Lemma 4. *There exists a partial f such that $\mathsf{LR}(f) \geq \Omega(\overline{\chi}(f)^{1.5})$.*

Proof. The function f we build takes input of size $n^2 + \sqrt{n}$ with format $(x_1, \ldots, x_{n^2}, a_1, \ldots, a_{\sqrt{n}})$. The function value is given as the parity of $\textsc{GapMaj}(x)$ and $\textsc{Xor}(a)$, i.e.:

$$f(x_1, \ldots, x_{n^2}, a_1, \ldots, a_{\sqrt{n}}) = \textsc{GapMaj}^{n^2}_{n^{-1/2}}(x_1, \ldots, x_{n^2}) \oplus \textsc{Xor}_{\sqrt{n}}(a_1, \ldots, a_{\sqrt{n}})$$

$\textsc{GapMaj}^{n^2}_{n^{-1/2}}(x)$ is the majority function on n^2 bits with promise that $|x| \notin [n^2/2 - n^{3/2}, n^2/2 + n^{3/2}]$ so that returning the value of a random index holds bias at least $n^{-1/2}$. Thus, f is a partial function whose domain is constrained by the gap majority instance. Lemma 16 shows that $\mathsf{LR}(f) \geq \Omega\left(n^{3/4}\right)$ and Lemma 17 that $\overline{\chi}(f) \leq O(n^{1/2})$, as desired. □

Lemma 16. $\mathsf{LR}(F) \geq \Omega\left(n^{3/4}\right)$

Proof. To obtain the lower-bound, we define a pair of distributions P^0, P^1 over $f^{-1}(0), f^{-1}(1)$ and use the minimax theorem for LR (see (7)):

$$\mathsf{LR}(f) = \max_{\mu}\min_{D}\frac{\mathrm{cost}(D,\mu)}{\mathrm{bias}_f(D,\mu)} \geq \min_{D}\frac{\mathrm{cost}(D,P)}{\mathrm{TV}(\mathrm{tran}(D,P^0),\mathrm{tran}(D,P^1))} \quad \text{where} \quad P := (P_1+P_2)/2$$

Let μ^0 be the hard distribution for no-instances of $\textsc{GapMaj}$, i.e μ^0 is uniform over all strings of Hamming weight $n^2 - n^{3/2}$. Similarly, we let μ^1 be the uniform distribution of all strings of Hamming weight $n^2 + n^{3/2}$.

Define further ν^0, respectively ν^1 to be the uniform distribution over even-parity, respectively odd-parity strings of size $n^{1/2}$. With those base distribution in hand, we define P^0 and P^1:

$$P^0 := \frac{\mu^0 \times \nu^0}{2} + \frac{\mu^1 \times \nu^1}{2} \quad \text{and} \quad P^1 := \frac{\mu^1 \times \nu^0}{2} + \frac{\mu^0 \times \nu^1}{2}$$

Fix now any decision tree D and let us argue that $\mathrm{cost}(D, P)/\mathrm{TV}(\mathrm{tran}(D, P^0), \mathrm{tran}(D, P^1)) \geq \Omega\left(n^{3/4}\right)$. Without loss of generality, we may assume that D has depth bounded by $n^{3/4}$ (see Lemma 41). Observe that any leaf $\ell \in \mathcal{L}(D)$ can be written as $\ell = \ell_x \circ \ell_a$ where ℓ_x is exclusively over variables $x_1, \ldots, x_{n^2}$ and ℓ_a over $a_1, \ldots, a_{\sqrt{n}}$. We further let $\mathcal{L}^s := \{\ell_x \circ \ell_a \in \mathcal{L}(D) : |\ell_a| = \sqrt{n}\}$ be the set of leaves that solve the Xor instance. Observe that for any $\ell \notin \mathcal{L}^s$, $\left|P^0[\ell] - P^1[\ell]\right| = 0$, indeed $\nu^0[\ell_a] = \nu^1[\ell_a] = 1/2$ so that:

$$\left|P^0[\ell] - P^1[\ell]\right| = \frac{1}{2} \cdot \left|\mu^0[\ell_x]\nu^0[\ell_a] + \mu^1[\ell_x]\nu^1[\ell_a] - \mu^1[\ell_x]\nu^0[\ell_a] - \mu^0[\ell_x]\nu^1[\ell_a]\right| = 0$$

We can therefore focus on bounding the bias contribution of leaves in $\mathcal{L}^s$. To that end, fix Γ to be the set of all conjunction of size $n^{1/2}$ over $a_1, \ldots, a_{\sqrt{n}}$ and for $\gamma \in \Gamma$, fix $\mathcal{L}_\gamma$ to be the set of leaves ℓ over $x_1, \ldots, x_n$ such that $\ell \circ \gamma \in \mathcal{L}(D)$ (note that it is possible that $\mathcal{L}_\gamma = \emptyset$ for some γ). The bias of D on P can now be re-expressed as:

$$\mathrm{TV}(\mathrm{tran}(D, P^0), \mathrm{tran}(D, P^1)) = \frac{1}{2} \sum_{\gamma \in \Gamma} \sum_{\ell \in \mathcal{L}_\gamma} \left|P^0[\ell \circ \gamma] - P^1[\ell \circ \gamma]\right| = \sum_{\gamma \in \Gamma} \nu[\gamma] \sum_{\ell \in \mathcal{L}_\gamma} \left|\mu^0[\ell] - \mu^1[\ell]\right| \tag{9}$$

Observe that $\mathcal{L}_\gamma$ can be seen as the leaves of D_γ, the tree which is obtained from *compressing* D with γ. For instance if $\gamma_1 = 1$, we swap any node querying a_1 with its children sub-tree corresponding to $a_1 = 1$. The inner sums can thus be interpreted as the bias D_γ holds in distinguishing μ^0 from μ^1. To bound those bias, it is convenient to replace μ (which is hyper-geometric) with a binomial variant $\tilde{\mu}$. We let $\tilde{\nu} := (\tilde{\nu}^0 + \tilde{\nu}^1)/2$, where $\tilde{\mu}^0$, $\tilde{\mu}^1$ yield n^2 iid Bernoulli$(1/2 - n^{-1/2})$, respectively Bernoulli$(1/2 + n^{-1/2})$ random variables. Because all D_γ have depth $\leq n^{3/4}$, we may relate μ to $\tilde{\mu}$ and use the hardness of Section 9.2 to get:

$$\begin{aligned}
\textstyle\sum_{\ell \in \mathcal{L}(D_\gamma)} \left|\mu^0[\ell] - \mu^1[\ell]\right| &= \textstyle\sum_{\ell \in \mathcal{L}(D_\gamma)} \left|\tilde{\mu}^0[\ell] - \mu^0[\ell]\right| + \left|\tilde{\mu}^0[\ell] - \tilde{\mu}^1[\ell]\right| + \left|\tilde{\mu}^1[\ell] - \mu^1[\ell]\right| \\
&\leq \textstyle\sum_{\ell \in \mathcal{L}(D_\gamma)} \left|\tilde{\mu}^0[\ell] - \tilde{\mu}^1[\ell]\right| + 24n^{-1/2}\left(\tilde{\mu}^0[\ell] + \tilde{\mu}^1[\ell]\right) && \text{(by Lemma 38)} \\
&= \mathrm{TV}(\mathrm{tran}(D_\gamma, \tilde{\mu}^0), \mathrm{tran}(D_\gamma, \tilde{\mu}^1)) + 48n^{-1/2} \\
&\leq O(1) \cdot n^{-1/2}\sqrt{\mathrm{cost}(D_\gamma, \tilde{\mu})} && \text{(by Theorem 22 with } b := n^{-1/2}\text{)} \\
&\leq O(1) \cdot n^{-1/2}\sqrt{\mathrm{cost}(D_\gamma, \mu)} && \text{(by Lemma 38)}
\end{aligned}$$

Note that by Lemma 38, we can extend (9) by using this bound and Cauchy-Schwarz inequality:

$$\begin{aligned}
\mathrm{TV}(\mathrm{tran}(D, P^0), \mathrm{tran}(D, P^1)) &\leq O(1) \cdot n^{-1/2} \sum_{\gamma \in \Gamma} \nu[\gamma]\sqrt{\mathrm{cost}(D_\gamma, \mu)} \\
&\leq O(1) \cdot n^{-1/2} \sqrt{\textstyle\sum_{\gamma \in \Gamma} \nu[\gamma]} \sqrt{\textstyle\sum_{\gamma \in \Gamma} \nu[\gamma]\,\mathrm{cost}(D_\gamma, \mu)}
\end{aligned}$$

The cost of D on P is easily lower-bounded by only taking leaves of $\mathcal{L}^s$ into account:

$$\mathrm{cost}(D, P) \geq \textstyle\sum_{\gamma \in \Gamma} \sum_{\ell \in \mathcal{L}_\gamma} \left(\sqrt{n} + |\ell|\right) \nu[\gamma]\mu[\ell] = \sqrt{n} \sum_{\gamma \in \Gamma} \nu[\gamma] + \sum_{\gamma \in \Gamma} \nu[\gamma]\,\mathrm{cost}(D_\gamma, \mu)$$

Combining both, we get the desired bound on the LR ratio of D:

$$\begin{aligned}
\frac{\mathrm{cost}(D, P)}{\mathrm{TV}(\mathrm{tran}(D, P^0), \mathrm{tran}(D, P^1))} &\geq \Omega(1) \cdot \max\left\{\frac{n \cdot \sqrt{\sum_{\gamma \in \Gamma} \nu[\gamma]}}{\sqrt{\sum_{\gamma \in \Gamma} \nu[\gamma]\,\mathrm{cost}(D_\gamma, \mu)}}, \frac{n^{1/2} \cdot \sqrt{\sum_{\gamma \in \Gamma} \nu[\gamma]\,\mathrm{cost}(D_\gamma, \mu)}}{\sqrt{\sum_{\gamma \in \Gamma} \nu[\gamma]}}\right\} \\
&\geq \Omega(n^{3/4})
\end{aligned}$$

□

6.1 An upper bound for $\overline{\chi}$

We recall here the definition of max conflict complexity (but see [GLSS19] for an in-depth treatment of the measure). Let f be a fixed boolean function, μ^0, μ^1 a pair of distribution over $f^{-1}(0)$ and $f^{-1}(1)$ respectively and D a deterministic decision tree solving f. For each node v in D, we let $\mu^0|_v, \mu^1|v$ be the distributions conditioned on reaching v and $q(v)$ be the index queried at node v. Furthermore, we associate to each $v \in \mathcal{N}(D)$ a number $R_\mu^D(v)$ inductively. If v is the root of D, we let $R_\mu^D(v) = 1$ and if v is the child of w which is reached when the query answer to $q(w)$ is $b \in \{0,1\}$:

$$R_\mu^D(v) = R_\mu^D(w) \cdot \min\left\{\Pr_{x\sim\mu^0|_w}[x_{q(w)} = b], \Pr_{x\sim\mu^1|_w}[x_{q(w)} = b]\right\}$$

Finally, we define $\Delta_\mu^D(v)$ for each $v \in \mathcal{N}(D)$ with:

$$\Delta_\mu^D(v) \coloneqq \left|\Pr_{x\sim\mu^0|_w}[x_{q(v)} = 0] - \Pr_{x\sim\mu^1|_w}[x_{q(v)} = 0]\right|$$

$R_\mu^D(v)$ can be interpreted as the probability of reaching node v in a random walk that starts at the root and with probability $\min\{\Pr_{x\sim\mu^0|_v}[x_i = 0], \Pr_{x\sim\mu^1|_v}[x_i = 0]\}$ moves left, with probability $\min\{\Pr_{x\sim\mu^0|_v}[x_i = 1], \Pr_{x\sim\mu^1|_v}[x_i = 1]\}$ moves right and with remaining probability $\Delta_\mu^D(v)$ stops. As such, it holds that $\sum_{v\in\mathcal{N}(D)} \Delta_\mu^D(v) R_\mu^D(v) = 1$ and that for any partition Γ of $\{0,1\}^n$ we have $\sum_{\gamma\in\Gamma} R_\mu^D(\gamma) \leq 1$. The max conflict complexity $\overline{\chi}(f)$ is defined as:

$$\overline{\chi}(f) \coloneqq \max_{\mathcal{Q}} \min_{D\in\mathsf{D}(f)} \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}} \left[\sum\nolimits_{v\in\mathcal{N}(D)} |v| \Delta_\mu^D(v) R_\mu^D(v)\right]$$

Where $\mathcal{Q}$ ranges over distributions of pairs of distributions over $f^{-1}(0)$ and $f^{-1}(1)$ and $\mathsf{D}(f)$ is the set of all decision tree solving f correctly.

Lemma 17. $\overline{\chi}(F) \leq O(n^{1/2})$

Proof. Let $\mathcal{Q}$ be the witness distribution over pairs of distribution for $\overline{\chi}(F)$ so that:

$$\overline{\chi}(F) = \min_{D\in\mathsf{D}(f)} \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}} \left[\sum\nolimits_{v\in\mathcal{N}(D)} |v| \Delta_\mu^D(v) R_\mu^D(v)\right]$$

We build a decision tree $D \in \mathsf{D}(F)$ and show that it witnesses $\overline{\chi}(F) \leq O(n^{1/2})$. Let Γ be the set of all conjunction of size $n^{1/2}$ over $a_1, \ldots, a_{\sqrt{n}}$. D starts by querying all the Xor variables $a_1, \ldots, a_{\sqrt{n}}$. We then append to each branch $\gamma \in \Gamma$ a decision tree D_γ on variables $x_1, \ldots, x_{n^2}$ depending on $\mathcal{Q}$. For each $\gamma \in \Gamma$, let $f|_\gamma$ be the function f with a set to γ and let us define a distribution $\mathcal{Q}_\gamma$ over distributions on $f|_\gamma^{-1}(0)$ and $f|_\gamma^{-1}(1)$. For each $\nu \in \mathrm{supp}(\mathcal{Q})$, we put the conditioned distribution $\nu|_\gamma$ in $\mathcal{Q}_\gamma$ and set its probability mass as:

$$\mathcal{Q}_\gamma[\nu|_\gamma] = \frac{\mathcal{Q}[\nu] R_\nu^D(\gamma)}{z_\gamma} \quad \text{where} \quad z_\gamma = \sum\nolimits_{\nu\in\mathrm{supp}(\mathcal{Q})} \mathcal{Q}[\nu] R_\nu^D(\gamma)$$

While D refers to the whole tree, z_γ does not actually depend on the choice of D_γ so that we are still free to choose it as

$$D_\gamma \coloneqq \mathop{\arg\min}_{D'\in\mathsf{D}(f|_\gamma)} \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}_\gamma} \left[\sum\nolimits_{v\in\mathcal{N}(D')} |v| \Delta_\mu^{D'}(v) R_\mu^{D'}(v)\right]$$

We now show that D witnesses $\overline{\chi}(F) \leq \sqrt{n}$. Indeed, we have:

$$\begin{aligned}
\overline{\chi}(F) &\leq \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}} \left[\sum_{v\in\mathcal{N}(D)} |v| \Delta_\mu^D(v) R_\mu^D(v)\right] \\
&= \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}} \left[\sum_{|v|<\sqrt{n}} |v| \Delta_\mu^D(v) R_\mu^D(v) + \sum_{\gamma\in\Gamma} \sum_{w\in\mathcal{N}(D_\gamma)} |\gamma \circ w| \Delta_\mu^D(\gamma \circ w) R_\mu^D(\gamma \circ w)\right]
\end{aligned}$$

$$
\begin{aligned}
&\le \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}}\left[\sum_{|v|<\sqrt{n}} \sqrt{n}\Delta_\mu^D(v)R_\mu^D(v) + \sum_{\gamma\in\Gamma}\sum_{w\in\mathcal{N}(D_\gamma)}(\sqrt{n}+|w|)\Delta_\mu^D(\gamma\circ w)R_\mu^D(\gamma\circ w)\right]\\
&\le \mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}}\left[\sum_{v\in\mathcal{N}(D)} \sqrt{n}\Delta_\mu^D(v)R_\mu^D(v) + \sum_{\gamma\in\Gamma}\sum_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_\mu^D(\gamma\circ w)R_\mu^D(\gamma\circ w)\right]\\
&\le \sqrt{n} + \sum_{\gamma\in\Gamma}\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}}\left[\sum_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_\mu^D(\gamma\circ w)R_\mu^D(\gamma\circ w)\right]\\
&= \sqrt{n} + \sum_{\gamma\in\Gamma}\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}}\left[\sum_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_{\mu|_\gamma}^{D_\gamma}(w)R_{\mu|_\gamma}^{D_\gamma}(w)R_\mu^D(\gamma)\right]\\
&= \sqrt{n} + \sum_{\gamma\in\Gamma}\sum_{\nu\in\operatorname{supp}\mathcal{Q}}\mathcal{Q}[\nu]R_\nu^D(\gamma)\sum_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_{\nu|_\gamma}^{D_\gamma}(w)R_{\nu|_\gamma}^{D_\gamma}(w)\\
&= \sqrt{n} + \sum_{\gamma\in\Gamma}z_\gamma\sum_{\nu\in\operatorname{supp}\mathcal{Q}}\mathcal{Q}_\gamma[\nu|_\gamma]\sum_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_{\nu|_\gamma}^{D_\gamma}(w)R_{\nu|_\gamma}^{D_\gamma}(w)\\
&= \sqrt{n} + \sum_{\gamma\in\Gamma}z_\gamma\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}_\gamma}\left[\sum_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_\mu^{D_\gamma}(w)R_\mu^{D_\gamma}(w)\right]
\end{aligned}
$$

Observe that for any $\gamma\in\Gamma$, $f|_\gamma\in\{\textsc{GapMaj},\neg\textsc{GapMaj}\}$, hence following our choice of D_γ, we have:

$$
\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}_\gamma}\left[\sum\nolimits_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_\mu^{D_\gamma}(w)R_\mu^{D_\gamma}(w)\right] = \min_{D'\in\mathsf{D}(f|_\gamma)}\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}_\gamma}\left[\sum\nolimits_{w\in\mathcal{N}(D)}|w|\Delta_\mu^{D'}(w)R_\mu^{D'}(w)\right]\le\overline{\chi}(f|_\gamma)
$$

Finally, note that $\mathsf{LR}(\textsc{GapMaj})\le O(n^{1/2})$, as witnessed by the algorithm that makes one random query and returns the result. Combining this observation together with Theorem 2 and the fact that $\overline{\chi}$ is inner-optimal yields $\overline{\chi}(f|_\gamma)\le O(n^{1/2})$ for every $\gamma\in\Gamma$ so that:

$$
\begin{aligned}
\overline{\chi}(F) &\le \sqrt{n}+\sum\nolimits_{\gamma\in\Gamma}z_\gamma\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}_\gamma}\left[\sum\nolimits_{w\in\mathcal{N}(D_\gamma)}|w|\Delta_\mu^{D_\gamma}(w)R_\mu^{D_\gamma}(w)\right]\\
&\le \sqrt{n}+O(\sqrt{n})\cdot\sum\nolimits_{\gamma\in\Gamma}z_\gamma\\
&= \sqrt{n}+O(\sqrt{n})\cdot\sum\nolimits_{\gamma\in\Gamma}\mathop{\mathbb{E}}_{\mu\sim\mathcal{Q}}\left[R_\mu^D(\gamma)\right]\\
&\le O(\sqrt{n}) && (\Gamma\text{ partitions the input space})\qquad\square
\end{aligned}
$$

Part II
Small-Bias Minimax

7 Overview

In this Part II, we prove the separation between LR and ULR in Theorem 6, restated below.

Theorem 6. *There is an n-bit partial function f such that $\mathsf{ULR}(f)\ge\Omega(\mathsf{LR}(f)^{5/4})\ge n^{\Omega(1)}$.*

We start, in this section, by giving an outline of the proof. Without further delay, we define the separating function f below. For convenience, we use $N=n^{O(1)}$ to denote the input length. It will be easy to show an upper bound $\mathsf{LR}(f)\le O(n)$ below in Lemma 19. The hard part is to show a lower bound $\mathsf{ULR}(f)\ge\Omega(n^{5/4})$, which will occupy us for Sections 8–11.

7.1 Choice of hard function

Definition 18 (Separating function)**.** *We define a partial function $f\colon \{0,1\}^N \to \{0,1,*\}$ using parameters n, B and $b_X \le o(b_Y) \le o(1)$. The input string is of length $N \coloneqq Bn$ and we think of it as being composed of B blocks, each of size n. The "type" of a block $(x_1, x_2, \dots, x_n)$ is determined by $x_1 \in \{0,1\}$:*

- *If $x_1 = 0$, we say the block is* easy *and the value of the block is x_2.*
- *If $x_1 = 1$, we say the block is* hard *and the value of the block is $\textsc{Xor}_{n-1}(x_2, \dots, x_n)$.*

The value of f is the majority value of the blocks. To make f partial, we promise that the number of hard block is either 0 or $4Bb_Y$. Moreover:

- *If there are 0 hard blocks, we guarantee there are at least $B/2 + Bb_X$ blocks with the same value.*
- *If there are $4Bb_Y$ hard blocks, we guarantee there are at least $B/2 + Bb_Y$ blocks with the same value.*

We will set $b_X \coloneqq n^{-1}$, $b_Y \coloneqq n^{-3/4}$, $B \coloneqq 8n^{9/2}$, so that effectively $N = 8n^{11/2}$.

Lemma 19. $\mathsf{LR}(f) \le O(n)$

Proof. Let R be the randomised tree that picks one block at random and outputs its value (i.e., easy blocks take two queries, hard blocks take n). Consider any input $x \in \{0,1\}^{Bn}$. Suppose first that the number of hard blocks in x is 0. In this case, the number of queries R makes is 2 and the bias is at least b_X so that the cost/bias ratio is $O(n)$. On the other hand, if the number of hard blocks is $4Bb_Y$, the bias is at least b_Y while the expected query cost is $\mathrm{cost}(R,x) = n \cdot (4Bb_Y)/B + 2 \cdot (1 - (4Bb_Y)/B) \le 5b_Y n$. □

7.2 Choice of hard input distributions

Our goal is now to prove a lower bound $\mathsf{ULR}(f) \ge \Omega(n^{5/4})$. To this end, we use the following minimax characterisation, which can be derived using the minimax theorem for ratios of bilinear functions [BB20a]

$$\mathsf{ULR}(f) \;\coloneqq\; \min_R \max_{x,y} \frac{\mathrm{cost}(R,x)}{\mathrm{bias}(R,y)} \;=\; \max_{\mu,\nu} \min_D \frac{\mathrm{cost}(D,\mu)}{\mathrm{bias}(D,\nu)}. \tag{10}$$

We are thus faced with the task of finding a pair of hard distributions μ and ν. To do so, we first define distributions X and Y, over inputs with no hard blocks and with $4Bb_Y$ hard blocks, respectively. Naturally, these distributions combine the hard distribution for $\textsc{GapMaj}$ and $\textsc{Xor}$. Crucially, we define Y as hiding the correct value of f inside the hard blocks—this way, an algorithm that never bothers to solve hard blocks would see only block values *negatively* biased against the correct value of f. Finally, we will define μ and ν as appropriate mixtures of X and Y.

Specifically, we define X^0 as the result of setting all blocks to easy and $B/2 + Bb_X$ of them have value 0. The process of picking the $B/2 + Bb_X$ many 0-blocks is made at uniformly at random (without replacement). Observe that creating an easy 0-block simply amounts to setting the underlying variables to 0^n. The distribution X^1 is defined analogously but with $B/2 + Bb_X$ many 1-blocks. Note that $f(X^0) = 0$ and $f(X^1) = 1$. The definition of Y^0 is more interesting. We first select the location of $4Bb_Y$ hard blocks whose values are set to 0 using the hard distribution for $\textsc{Xor}_{n-1}$ (uniform over $\textsc{Xor}_{n-1}^{-1}(0)$). The remainder of the blocks are easy and $B/2 - 3Bb_Y$ are set to 0 while $B/2 - Bb_Y$ are set to 1. Observe that the values of the easy blocks are indeed negatively biased toward the right answer: the probability of getting the correct value by sampling one easy block is $(B/2 - 3Bb_Y)/(B - 4Bb_y) \le 1/2 - b_Y$. The distribution Y^1 is defined similarly. We now let

$$\begin{aligned} X &\coloneqq \tfrac{1}{2}X^0 + \tfrac{1}{2}X^1, \\ Y &\coloneqq \tfrac{1}{2}Y^0 + \tfrac{1}{2}Y^1, \\ \mu &\coloneqq \tfrac{1}{2}X + \tfrac{1}{2}Y, \\ \nu &\coloneqq (1-\lambda)X + \lambda Y, \end{aligned}$$

where we will set $\lambda \coloneqq \Theta(b_X/b_Y) = \Theta(n^{-1/4})$ with a carefully chosen implicit constant (see Section 9.1).

Intuition. Here is the intuition behind our definition of the hard pair μ, ν. First, hard blocks are much more likely to be found in μ than in ν. Hence, if an algorithm D wants to keep $\mathrm{cost}(D, \mu)$ low, it should not solve hard blocks very often. On the other hand, consider the following algorithm D that solves no hard blocks: Query a block, if it is easy, output its value; if it is hard, output a random guess. What is the bias of this algorithm? We have $\mathrm{bias}(D, X) = b_X$ and $\mathrm{bias}(D, Y) = -b_Y$ and thus the two biases cancel each other out: $\mathrm{bias}(D, \nu) = (1 - \lambda)b_X - \lambda b_Y \approx 0$. The challenge for us will be to rule out every such decision tree that does not solve hard blocks. However, this analysis will become subtle. For example, if the algorithm witnesses a hard block (but does not solve it), it still learns that the input comes from Y and it then knows that the rest of the blocks are negatively biased.

7.3 Lower-bound plan

Now that the hard distributions μ, ν are chosen, our goal is to show for every deterministic tree D,

$$\frac{\mathrm{cost}(D, \mu)}{\mathrm{bias}(D, \nu)} \geq \Omega(n^{5/4}). \tag{11}$$

We now outline our lower-bound strategy that we will carry out in the upcoming Sections 8–11.

(§8.1) **Simplification I.** We start by making two simplifications. First, we rule out any decision tree whose strategy is to solve hard blocks (querying all n bits of a block) with noticeable probability. Such trees have too high a cost relative to μ (which includes hard blocks with probability 1/2). We will henceforth assume that a given tree D never solves a hard block. Moreover, we are left with a challenge of proving an LR-style cost/bias trade-off relative to ν only. (We forget μ from now on.)

(§8.2) **Simplification II.** Second, we simplify the analysis of decision trees by switching to a *infinite-stream* model of computation. Instead of the N-bit input distributions such as $X = \frac{1}{2}X^0 + \frac{1}{2}X^1$ we model them as *infinite streams* $\tilde{X} = \frac{1}{2}\tilde{X}^0 + \frac{1}{2}\tilde{X}^1$ that record the values of the blocks, with the original biases, but with more independence. For example, $\tilde{X}^1$ is an infinite sequence of iid Bernoulli$(1/2 + b_X)$. Moreover, each of $\tilde{X}, \tilde{Y}, \tilde{\nu}$ is a mixture of at most 4 streams of iid symbols.

(§9) **Two basic hardness results.** At this point, it remains to prove a cost/bias trade-off for stream $\tilde{\nu}$. In preparation for this, we establish some very basic query lower bounds that are well-known in the worst-case setting, but which have not been yet proved in the expected-cost setting. For example, every algorithm trying to distinguish between an iid stream of Bernoulli$(1/2+b)$ and an iid stream of Bernoulli$(1/2 + b)$ must have bias at most $O(b\sqrt{\mathrm{cost}})$. Because the cost is measured in expectation, these basic facts turn out to be somewhat tricky to prove.

(§10) **Lower bound for $\boldsymbol{\nu}$-stream.** We now have the tools to analyse the cost/bias trade-off for $\tilde{\nu}$. The analysis here is rather intricate, as there are several decision tree strategies that we need to defeat. Some technical calculations are relegated to Section 11 and Appendix A.

8 Two simplifications

Having chosen two hard pair of distribution μ, ν to be plugged in the minimax characterisation (10) for ULR, it remains to be shown that $\mathrm{cost}(D, \mu)/\mathrm{bias}_f(D, \nu) \geq \Omega(n^{5/4})$ for all decision tree D to prove Theorem 6. The purpose of this section is to make two simplifying steps (as sketched in Section 7.3) and reduce the lower-bound task to showing

$$\frac{\mathrm{cost}(D, \tilde{\nu})}{\mathrm{TV}(\mathrm{tran}(D, \tilde{\nu}^0), \mathrm{tran}(D, \tilde{\nu}^1))} \geq \Omega(n^{5/4}) \quad \text{for all deterministic decision trees } D. \tag{12}$$

Here D will be a ternary decision tree over $\{0, 1, \mathtt{H}\}$ (where $\mathtt{H}$ models a hard block that the algorithm did not solve) and $\tilde{\nu}$ an infinite stream over $\{0, 1, \mathtt{H}\}$ representing ν. The goal of this section is to define and justify precisely the reduction, leaving the proof of (12) for Section 10.

8.1 Simplification I: No hard blocks

The first simplification step rules out any decision tree whose strategy is to solve blocks completely. In that regard, having $\lambda \ll 1$ is critical. Let us say that D *solves a hard block* on input x, denoted $S(D, x)$, if the leaf reached by x on D contains all the n bits of some block with first bit equal to 1. Observe that $S(D, x)$ can only hold when x actually contains a hard block, i.e., when it is coming from Y. Note that equating "solving" with "querying the whole block" is justified since the value v of a hard block is hidden by the uniform distribution $\textsc{Xor}_{n-1}^{-1}(v)$, offering zero bias until the very last query. Let $\nu[S]$ and $Y[S]$ be the probability that D solves a hard block when the input is drawn from ν, respectively Y. If $\nu[S] \geq \mathrm{bias}(t, \nu)/3$, the desired bound $\Omega(n^{5/4})$ in (11) is already attained because $\lambda = \Theta(b_X/b_Y)$ and

$$\mathrm{cost}(D, \mu) \geq \frac{1}{2} \cdot \mathrm{cost}(D, Y) \geq \frac{n}{2} \cdot Y[S] \geq \frac{n}{2\lambda} \cdot \nu[S] \geq \frac{n}{6\lambda} \cdot \mathrm{bias}(D, \nu) \geq \Omega(\mathrm{bias}(D, \nu) \cdot n^{5/4}).$$

Thus we may assume $\nu[S] \leq \mathrm{bias}(t, \nu)/3$ henceforth. We let D' be the copy of D that stops whenever it is about to solve a hard block, i.e., instead of querying the last remaining bit of a hard block, D' stops and outputs the most likely answer. Note that $\mathrm{cost}(D', \mu) \leq \mathrm{cost}(D, \mu)$, but D' still enjoys a constant fraction of the bias that D has against ν:

$$\begin{aligned}\mathrm{bias}(D', \nu) = 2 \Pr_{x\sim\nu}[D'(x) = f(x)] - 1 &\geq 2 \Pr_{x\sim\nu}[D'(x) = f(x) \text{ and } \neg S(D, x)] - 1 \\ &= 2 \Pr_{x\sim\nu}[\neg S(D, x)] \Pr_{x\sim\nu}[D(x) = f(x)] - 1 \\ &\geq \mathrm{bias}(D, \nu)/3\end{aligned}$$

As $\mathrm{cost}(D', \mu) \geq \mathrm{cost}(D', \nu)/2$, we have reduced (11) to showing that $\mathrm{cost}(D, \nu)/\mathrm{bias}(D, \nu) \geq \Omega(n^{5/4})$ for any decision tree D that never solves hard blocks completely. *Note that this is an* LR*-style lower bound relative to a single hard distribution ν and against a restricted class of algorithms.*

We may restrict the class of admissible algorithms even further. Observe first that a decision tree D that does not solve any hard block can be simulated by a *randomised* tree R that never queries anything outside the first two variables $\{x_1, x_2\}$ of any given block. Indeed, if a block is easy, the values beyond the two first bits are fixed to 0^{n-2} and need not be queried. If a block is hard, then the marginal distribution of any proper subset of the variables $x_2, \ldots, x_n$ is uniform, and R can simulate this distribution itself with internal randomness. We may now derandomise such an R back into a deterministic tree: if R has $\mathrm{cost}(R, \nu)/\mathrm{bias}_f(R, \nu) \leq C$, then there must be a deterministic decision tree D' in its support that also achieves $\mathrm{cost}(D', \nu)/\mathrm{bias}_f(D', \nu) \leq C$.

After the above simplification, we only need to show that for any tree D that is constrained to read at most variables $\{x_1, x_2\}$ within each block, $\mathrm{cost}(D, \nu)/\mathrm{bias}(D, \nu) \geq \Omega(n^{5/4})$. At the cost of losing a constant factor in the bound, we may also assume that whenever D queries x_2, it first queries x_1. Observe that such a D only sees three types of events: easy 0-block, easy 1-block or hard block with undisclosed value. D can thus be interpreted as a *ternary* decision tree over the alphabet $\{0, 1, \textsc{h}\}$ (where $\textsc{h}$ means hard block with undisclosed value) trying to solve a ternary analogue $\overline{f}$ of f against inputs sampled from the ternary analogue $\overline{\nu}$ of ν. More precisely, $\overline{f}$ maps $\{0, 1, \textsc{h}\}^B \to \{0, 1, *\}$ and $\overline{\nu}$ a distribution over $\{0, 1, \textsc{h}\}^B$ defined as before but now based on $\overline{X}$ and $\overline{Y}$:

1. $\overline{X}^0$: $B/2 + Bb_X$ random locations are set to 0, the rest are 1.
2. $\overline{X}^1$: $B/2 + Bb_X$ random locations are set to 1, the rest are 0.
3. $\overline{Y}^0$: $4Bb_Y$ locations are set to $\textsc{h}$, $B/2 - 3Bb_Y$ locations to 0 and $B/2 - Bb_Y$ locations to 1.
4. $\overline{Y}^1$: $4Bb_Y$ locations are set to $\textsc{h}$, $B/2 - 3Bb_Y$ locations to 1 and $B/2 - Bb_Y$ locations to 0.

Following this discussion, to prove (11), we only need to show:

$$\frac{\mathrm{cost}(D, \overline{\nu})}{\mathrm{bias}_{\overline{f}}(D, \overline{\nu})} \geq \Omega(n^{5/4}) \quad \text{for all ternary decision tree } D \tag{13}$$

Ternary decision tree. We use an intuitive notion of ternary decision tree, where each query node has three children corresponding to each possible query response in $\{0, 1, \textsc{h}\}$. Akin to their binary analogues, we will see leaves of ternary tree as conjunction over literals with three possible states: positive, negative or *witnessing*. This new witnessing literal type amounts to checking whether the variable equals $\textsc{h}$. Following this, we will say that a leaf ℓ is witnessing, in short $\textsc{h} \in \ell$, if it holds a witnessing literal.

8.2 Simplification II: Streaming model

Following Section 8.1, we are left with the task of showing (13). One of the main technical annoyances in working with $\overline{\nu}$ is that its entries are generated using a hypergeometric distribution, where, even in $\overline{X}^0$, the bits are not iid. To overcome this issue and simplify the calculations, we propose to replace $\overline{\nu}$ with a a multinomial variant $\tilde{\nu}$. This change requires attention on two counts. First, we need to ensure that D is shallow enough and B is large enough, thus asserting that the behavior of D on $\tilde{\nu}$ is close to the behavior of D on $\overline{\nu}$. Second and most importantly, observe that it is possible that $\widetilde{\nu}^0$ yields a 1-input (and $\widetilde{\nu}^1$ a 0-input). This makes the usual notion of bias moot, but we can still resort to the total-variation distance interpretation of the bias. Indeed, assuming without loss of generality that D is optimally labelled for distinguishing $\overline{\nu}^0$ from $\overline{\nu}^1$:

$$\text{bias}_{f'}(D, \overline{\nu}) = \text{TV}(\text{tran}(D, \overline{\nu}^0), \text{tran}(D, \overline{\nu}^1)) = \frac{1}{2} \cdot \sum\nolimits_{\ell \in \mathcal{L}(D)} \left| \overline{\nu}^0[\ell] - \overline{\nu}^1[\ell] \right|$$

Thus, (13) can actually be re-written swapping $\text{TV}(\text{tran}(D, \overline{\nu}^0), \text{tran}(D, \overline{\nu}^1)$ in place of $\text{bias}_{f'}(T, \overline{\nu})$ and this relaxed formulation opens the floor for the desired multinomial variant. $\tilde{\nu}$ is defined analogously to $\overline{\nu}$, by mixing appropriately the distributions $\tilde{X}^0$, $\tilde{X}^1$, $\tilde{Y}^0$ and $\tilde{Y}^1$:

- $\tilde{X}^0$: B iid Bernoulli$(1/2 - b_X)$ random variables.
- $\tilde{X}^1$: B iid Bernoulli$(1/2 + b_X)$ random variables.
- $\tilde{Y}^0$: B iid random variables taking value 0 with probability $1/2 - 3b_Y$, 1 with probability $1/2 - b_Y$ and $\textsc{h}$ with remaining probability $4b_Y$.
- $\tilde{Y}^1$: B iid random variables taking value 0 with probability $1/2 - b_Y$, 1 with probability $1/2 - 3b_Y$ and $\textsc{h}$ with remaining probability $4b_Y$.

Lemma 20. *If* $\text{cost}(D, \tilde{\nu})/\text{TV}(\text{tran}(D, \tilde{\nu}^0), \text{tran}(D, \tilde{\nu}^1) \geq \Omega(n^{5/4})$ *for all* D*, then* (13) *holds.*

Proof. Fix some ternary decision tree D, and let us show that (13) holds for D. We may assume without loss of generality that D has depth bounded by $n^{5/4}$, as else D would essentially already have LR ratio $\geq \Omega(n^{5/4})$ (see Lemma 41 for details). The claim would also be vacuously true if $\text{bias}_{f'}(D, \overline{\nu}) \leq n^{-5/4}$. Fix now any leaf $\ell \in \mathcal{L}(D)$ and let q_0, q_1 and $q_{\textsc{h}}$ be the number of negative, positive and witnessing literals in ℓ so that $|\ell| = q_0 + q_1 + q_{\textsc{h}}$. Recalling that $B = 8n^{9/2}$, we have $|\ell| \ll \sqrt{B}$, so that using Lemma 38 for X and Lemma 37 for Y:

$$\left| \tilde{X}^d[\ell] - \overline{X}^d[\ell] \right| \leq \frac{12|\ell|^2}{B} \cdot \tilde{X}^d[\ell] \quad \text{and} \quad \left| \tilde{Y}^d[\ell] - \overline{Y}^d[\ell] \right| \leq \frac{2|\ell|^2}{Bb_Y} \cdot \tilde{Y}^d \quad \forall d \in \{0, 1\}$$

In short, $|\tilde{\nu}^d[\ell] - \overline{\nu}^d[\ell]| \leq n^{-5/4}\tilde{\nu}^d[\ell]/4$ and so $\text{cost}(D, \overline{\nu}) \geq \text{cost}(D, \tilde{\nu})/2$. Furthermore, both total variation distance are close:

$$\begin{aligned}
\text{bias}_{f'}(D, \overline{\nu}) = \sum\nolimits_{\ell \in \mathcal{L}(D)} \left| \overline{\nu}^0[\ell] - \overline{\nu}^1[\ell] \right| &\leq \sum\nolimits_{\ell \in \mathcal{L}(D)} \left| \overline{\nu}^0[\ell] - \tilde{\nu}^0[\ell] \right| + \left| \tilde{\nu}^0[\ell] - \tilde{\nu}^1[\ell] \right| + \left| \overline{\nu}^1[\ell] - \tilde{\nu}^1[\ell] \right| \\
&\leq \sum\nolimits_{\ell \in \mathcal{L}(D)} \left| \tilde{\nu}^0[\ell] - \tilde{\nu}^1[\ell] \right| + \sum\nolimits_{\ell \in \mathcal{L}(t)} n^{-5/4}\tilde{\nu}/2 \\
&\leq n^{-5/4}/2 + \sum\nolimits_{\ell \in \mathcal{L}(D)} \left| \tilde{\nu}^0[\ell] - \tilde{\nu}^1[\ell] \right| \\
&\leq \text{bias}_{f'}(D, \overline{\nu})/2 + \sum\nolimits_{\ell \in \mathcal{L}(D)} \left| \tilde{\nu}^0[\ell] - \tilde{\nu}^1[\ell] \right|
\end{aligned}$$

Thus, if $\text{cost}(D, \tilde{\nu})/\text{TV}(\text{tran}(D, \tilde{\nu}^0), \text{tran}(D, \tilde{\nu}^1) \geq \Omega(n^{5/4})$ holds for D, we have, as desired:

$$\frac{\text{cost}(D, \overline{\nu})}{\text{bias}_{f'}(D, \overline{\nu})} \geq \Omega(1) \cdot \frac{\text{cost}(D, \tilde{\nu})}{\text{TV}(\text{tran}(D, \tilde{\nu}^0), \text{tran}(D, \tilde{\nu}^1))} \geq \Omega(n^{5/4}) \qquad \square$$

As an ultimate simplification, instead of multinomial distributions over $\{0,1,\textsc{h}\}^B$, we will see $\tilde{X}^0$, $\tilde{X}^1$, $\tilde{Y}^0$ and $\tilde{Y}^1$ as *infinite* iid streams and allow D to be have unbounded length. Because of Lemma 41, this generalization adds no real power but this stream framework is generally nicer to work with. In particular, the parameter B is not relevant anymore and as such we will only use b_X and b_Y for later sections. Following our reduction, proving $\mathsf{ULR}f \geq \Omega(n^{5/4})$ now amounts to proving the following:

$$\frac{\operatorname{cost}(D,\tilde{\nu})}{\operatorname{TV}(\operatorname{tran}(D,\tilde{\nu}^0),\operatorname{tran}(D,\tilde{\nu}^1))} \geq \Omega(n^{5/4}) \quad \text{for all infinite ternary decision tree } D \tag{14}$$

8.2.1 Some thoughts on the LR streaming model

Note that by the iid nature of random streams, the indices queried by decision trees do not matter. Actually, we could even force D to query variable x_1 at level 1, x_2 at level 2 and so forth. This however doesn't prevent D from adopting an adaptive strategy: D can indeed decide *when to stop* depending on past query answers. This allows for some wildly unbalanced decision trees that need to be ruled out, and as such it is a challenge to prove our results in the *expected* query cost setting.

The lower bound (14) can also be seen as a distribution testing hardness result. Consider the problem in which a secret φ is sampled amongst $(\tilde{X}^0,\tilde{X}^1,\tilde{Y}^0,\tilde{Y}^1)$ with prior $(\lambda/2,\lambda/2,(1-\lambda)/2,(1-\lambda)/2)$ and one needs to decide whether $\varphi \in \{\tilde{X}^0,\tilde{Y}^0\}$ or $\varphi \in \{\tilde{X}^1,\tilde{Y}^1\}$ by making repeated queries to the stream φ. Then, (14) says that any decision tree D accomplishing this task must have LR ratio $\geq \Omega(n^{5/4})$.

As a technicality, we will need to resort to LR-streaming bounds for randomised decision tree. For this, we define the total variation distance on transcript distribution naturally with $\operatorname{TV}(\operatorname{tran}(R,P^0),\operatorname{tran}(R,P^1)) = \mathbb{E}_{D\sim R}[\operatorname{TV}(\operatorname{tran}(D,P^0),\operatorname{tran}(D,P^1))]$ (where P^0 and P^1 are two distributions). In particular, this is still consistent with the view that if R is labelled optimally by $\{P^0,P^1\}$, then $\operatorname{TV}(\operatorname{tran}(R,P^0),\operatorname{tran}(R,P^1)) = \Pr_{x\sim P^0}[R(x)=P^0] - \Pr_{x\sim P^1}[T(x)=P^0]$

9 Two basic hardness results

Before tackling the proof of (14) and ultimately showing $\mathsf{ULR}(f) \geq \Omega(n^{5/4})$, we first establish in this section a couple of basic results in the LR-streaming model. Those self-contained results are a good starting point to get acquainted with LR-style lower bounds and have the added benefit of being a key component of our main technical result Theorem 24.

9.1 Source of hardness I: Bernoulli mixtures

As a first step toward bounding the hardness of distinguishing $\tilde{\nu}^0$ from $\tilde{\nu}^1$ with a decision tree, we study an idealized version with no hard blocks. More precisely, for two parameters $b_X < o(b_Y) < o(1)$, we let X^0, X^1, Z^0 and Z^1 be iid random stream of Bernoulli$(1/2-b_X)$, Bernoulli$(1/2+b_X)$, Bernoulli$(1/2+b_Y)$ and Bernoulli$(1/2-b_Y)$ random variables and define the mixtures of random streams M^0 and M^1 with:

$$\begin{aligned} M^0 &\coloneqq (1-\lambda)X^0+\lambda Z^0 \\ M^1 &\coloneqq (1-\lambda)X^1+\lambda Z^1 \end{aligned} \qquad \text{where} \qquad \frac{1-\lambda}{\lambda} = \frac{\ln\left(\frac{1+2b_Y}{1-2b_Y}\right)}{\ln\left(\frac{1+2b_X}{1-2b_X}\right)} \tag{15}$$

Finally, we let $M \coloneqq (M^0+M^1)/2$. We will show that $\operatorname{TV}(\operatorname{tran}(D,M^0),\operatorname{tran}(D,M^1)) \leq O(b_X b_Y \operatorname{cost}(D,M))$ for any deterministic D. The fine-tuning of λ in (15) will turn out to be a necessary technicality, but $\lambda = \Theta(b_X/b_Y)$ as Lemma 43 shows. Following the initial plan, distinguishing M^0 from M^1 can be interpreted as distinguishing $\tilde{\nu}^0$ from $\tilde{\nu}^1$ *with no* $\textsc{h}$, in particular Z carries overall *negative bias*. This particular source of hardness will be used to analyse the bias brought by *small* leaves, whose behavior for $\tilde{\nu}^0$ versus $\tilde{\nu}^1$ is close to M^0 versus M^1. As a secondary goal, the companion proof exemplifies one of the simplest way to obtain a LR lower bound, namely finding a *hybrid* distribution—in this case the uniform distribution U—and apply a corruption bound.

Theorem 21. *For any tree D, we have* $\operatorname{TV}(\operatorname{tran}(D,M^0),\operatorname{tran}(D,M^1)) \leq O(b_X b_Y \operatorname{cost}(D,M))$.

Proof. Following Lemma 41, we may assume without loss of generality that D has depth bounded by $1/9b_Xb_Y$. Let U be the random stream of Bernoulli$(1/2)$ variables and decompose the bias of D as:

$$2 \cdot \mathrm{TV}(\mathrm{tran}(D, M^0), \mathrm{tran}(D, M^1)) = \sum_{\ell \in \mathcal{L}(D)} \left|M^0[\ell] - M^1[\ell]\right| \leq \sum_{\ell \in \mathcal{L}(D)} \left|M^0[\ell] - U[\ell]\right| + \sum_{\ell \in \mathcal{L}(D)} \left|M^1[\ell] - U[\ell]\right|$$

We focus on the first sum as the bound on the second follows by a symmetrical argument. The first sum can be interpreted as the bias held by D in distinguishing U from M^0. Letting $\mathcal{L}^U = \{\ell \in \mathcal{L}(D) : U[\ell] \geq M^0[\ell]\}$:

$$\begin{aligned}
\textstyle\sum_{\ell \in \mathcal{L}(D)} \left|M^0[\ell] - U[\ell]\right| &= 2 \textstyle\sum_{\ell \in \mathcal{L}^U} U[\ell] - M^0[\ell] \\
&\leq O(1) \cdot b_X b_Y \textstyle\sum_{\ell \in \mathcal{L}^U} |\ell| \cdot U[\ell] && \text{(by Lemma 32)} \\
&\leq O(1) \cdot b_X b_Y \textstyle\sum_{\ell \in \mathcal{L}^U} |\ell| \cdot M^0[\ell] && \text{(by Lemma 32 with } |\ell| \leq 1/9b_Xb_Y\text{)} \\
&\leq O(1) \cdot b_X b_Y \,\mathrm{cost}(D, M)
\end{aligned}$$

□

9.2 Source of hardness II: Bernoulli with opposite bias

Our second hardness result tackles the problem of distinguishing a random stream B^0 of iid Bernoulli$(1/2-b)$ variables from the symmetric random stream B^1 with variables sampled from Bernoulli$(1/2+b)$, where $b \in o(1)$ is a parameter. This basic result will be employed several times in later section. For instances, witnessing leaves need to solve the $\tilde{Y}^0$ versus $\tilde{Y}^1$, which is essentially B^0 versus B^1 with $b \coloneqq b_Y$. In what follows, we let $B \coloneqq (B^0 + B^1)/2$.

Theorem 22. *For any randomised decision tree R,* $\mathrm{TV}(\mathrm{tran}(R, B^0), \mathrm{tran}(R, B^1)) \leq O(b\sqrt{\mathrm{cost}(R, B)})$

Note that the theorem statement features a randomised decision tree instead of a deterministic one, a generalization needed for later use. There are essentially three ways to prove a result similar to the one of Theorem 22. The first is to use a direct corruption bound, akin to Theorem 21 - but this would only give the weaker bound $\mathrm{TV}(\mathrm{tran}(R, B^0), \mathrm{tran}(R, B^1)) \leq O(b\,\mathrm{cost}(R, B)))$. A second would be to leverage the machinery of later sections and especially Lemma 34 to get a bias bound *per leaf*, ultimately yielding $\mathrm{TV}(\mathrm{tran}(R, B^0), \mathrm{tran}(R, B^1)) \leq O(b\sqrt{\mathrm{cost}(R, B)}\mathrm{polylog}(\mathrm{cost}(R, B)))$. Even though this is enough to cover the desired polynomial separation, Theorem 22 is optimal and we believe, interesting on its own. The crux is to re-use an idea brought by Sherstov [She12] in the context of bounding the communication complexity of the gap-Hamming function. As a first step, we prove some hardness result in distinguishing B from U.

Lemma 23. *For any randomised decision tree R,* $\mathrm{TV}(\mathrm{tran}(R, U), \mathrm{tran}(R, B)) \leq 4b^2\,\mathrm{cost}(R, U/2 + B/2)$.

Proof. We show this for a deterministic D instead of R as the randomised case follows by linearity of expectation. Let us begin by showing that for any $\ell \in \mathcal{L}(D)$, $B[\ell]/U[\ell] \geq 1 - 2|\ell|b^2$. Let $k \coloneqq |\ell|$ and supposing that ℓ has $k/2 + m$ positive literals (so $m \in [-k/2, k/2]$):

$$\frac{B[\ell]}{U[\ell]} = \frac{1}{2} \cdot \left(1 - 4b^2\right)^{k/2} \cdot \left[\left(\frac{1+2b}{1-2b}\right)^m + \left(\frac{1-2b}{1+2b}\right)^m\right]$$

The quantity within the square bracket is a function of m and we find its minimum by setting its derivative equal to zero, yielding $m = 0$. Note that this shows that the most separating leaves have as many positive literals as negative ones. In any case, it holds that $B[\ell]/U[\ell] \geq (1 - 4b^2)^{k/2}$ so that by defining $\mathcal{L}^U \coloneqq \{\ell \in \mathcal{L}(D) : U[\ell] \geq B[\ell]\}$ and using the approximation of Lemma 44:

$$\mathrm{TV}(\mathrm{tran}(D, U), \mathrm{tran}(D, B)) = \sum_{\ell \in \mathcal{L}^U} U[\ell] - B[\ell] \leq \sum_{\ell \in \mathcal{L}^U} U[\ell] \cdot \left(1 - (1 - 4b^2)^{|\ell|/2}\right) \leq \sum_{\ell \in \mathcal{L}^U} U[\ell] \cdot 2b^2|\ell|$$

We may finally conclude:

$$\frac{\mathrm{cost}(D, U/2 + B/2)}{\mathrm{TV}(\mathrm{tran}(D, U), \mathrm{tran}(D, B))} \geq \frac{\mathrm{cost}(D, U)}{4b^2 \cdot \mathrm{cost}(D, U)} = 1/4b^2$$

□

Proof of Theorem 22. As a first step, we prove the claim for deterministic D and extend it to randomised ones at the end. Following an argument similar to the one of Lemma 41, we may assume that D has depth bounded by $1/5b^2$. It will be convenient to see D as having leaves labelled optimally by $\{B^0, B^1\}$. Let $\delta \coloneqq \mathrm{TV}(\mathrm{tran}(D, B^0), \mathrm{tran}(D, B^1)) = \mathrm{Pr}_{x\sim B^0}[D(x) = B^0] - \mathrm{Pr}_{x\sim B^1}[D(x) = D^0]$ be the bias of D in B^0 against B^1 and note that we can mix D with the trivial tree that always accept (or always reject) to get a randomised decision tree R with centred acceptance probability:

$$\mathrm{Pr}_{x\sim B^0}[R(x) = B^0] = 1/2 + \xi \quad \text{and} \quad \mathrm{Pr}_{x\sim B^1}[R(x) = B^0] = 1/2 - \xi \quad \text{where} \quad \xi \geq \delta/6$$

The construction of R is detailed in Lemma 42, but R has worst-case depth bounded by $1/5b^2$ too. We will use R to build another randomised decision tree $R^\star$ which can distinguish B from U, effectively reducing the hardness of B^0 versus B^1 to distinguishing B from U. Define R_{neg} as a copy of R with *inverted* labels, i.e. leaves labelled with B^0 are now labelled with B^1 and reciprocally. The tree $R^\star$ we build will have $\mathrm{TV}(\mathrm{tran}(R^\star, U), \mathrm{tran}(R^\star, B)) \geq \Omega(\xi^2)$ and small cost. The construction of $R^\star$ depends on the value of $\alpha \coloneqq \mathrm{Pr}_{x\sim U}[R(x) = B^0] - 1/2$. If $\alpha \in [-\delta/7, \delta/7]$, we let $R^\star$ execute R and R_{neg} in turn on independent bits and output U if both run output B^0 and B else. The independent runs can be performed by off-setting the query indices of R_{neg} by a large number, e.g. $\lceil 1/b^2 \rceil$. We have:

$$\begin{aligned}
\Pr_{x\sim B}[R^\star(x) = U] &= \frac{1}{2} \cdot \Pr_{x\sim B^0}[R^\star(x) = U] + \frac{1}{2} \cdot \Pr_{x\sim B^1}[R^\star(x) = U] \\
&= \frac{1}{2} \cdot \Pr_{x,x'\sim B^0}[R(x) = R_{\mathrm{neg}}(x') = B^0] + \frac{1}{2} \cdot \Pr_{x,x'\sim B^1}[R(x') = R_{\mathrm{neg}}(x) = B^0] \\
&= \left(\frac{1}{2} + \xi\right) \cdot \left(\frac{1}{2} - \xi\right) \\
&\leq \frac{1}{4} - \frac{\delta^2}{36}
\end{aligned}$$

On the other hand, $\mathrm{Pr}_{x\sim U}[R^\star(x) = U] = (1/2 + \alpha) \cdot (1/2 - \alpha) \geq 1/4 - \delta^2/49$ and hence for this regime of α, $R^\star$ achieves bias $\mathrm{TV}(\mathrm{tran}(R^\star, U), \mathrm{tran}(R^\star, B)) \geq 13\delta^2/1764$. Finally, if $\alpha \geq \delta/7$ we pick $R^\star \coloneqq R$ and since R is centred, $\mathrm{Pr}_{x\sim B}[R(x) = U] = 1/2$. On the other hand, we have $\mathrm{Pr}_{x\sim U}[R(x) = U] \geq 1/2 + \xi/7$. The case $\alpha \leq \delta/7$ is analogous and requires picking $R^\star \coloneqq R_{\mathrm{neg}}$. In any case, we get a construction $R^\star$ with $\mathrm{TV}(\mathrm{tran}(R^\star, U), \mathrm{tran}(R^\star, A)) \geq \Omega(\delta^2)$ and worst-case depth $2/5b^2$, implying that for any D' in the support of $R^\star$:

$$\mathrm{cost}(D', B) = \sum_{\ell\in\mathcal{L}(D')} |\ell| \cdot B[\ell] \geq \sum_{\ell\in\mathcal{L}(D')} |\ell| \cdot U[\ell] \cdot \left(1 - 2|\ell|b^2\right) \geq \frac{1}{5} \sum_{\ell\in\mathcal{L}(D')} |\ell| \cdot U[\ell] = \frac{\mathrm{cost}(D', U)}{5}$$

The first inequality holds because of the analysis in Lemma 23. Applying linearity of expectation, we have that $\mathrm{cost}(R^\star, U/2 + B/2) \leq 3\,\mathrm{cost}(R^\star, B)$. Observe that the only non-trivial tree in the support of $R^\star$ is the initial D so that $\mathrm{cost}(D, B) \geq \Omega(\mathrm{cost}(R^\star), U/2 + B/2)$ and hence using Lemma 23:

$$\frac{\sqrt{\mathrm{cost}(D, B)}}{\delta} \geq \Omega\left(\frac{\mathrm{cost}(R^\star, U/2 + B/2)}{\mathrm{TV}(\mathrm{tran}(R^\star, U), \mathrm{tran}(R^\star, B))}\right)^{1/2} \geq \Omega(1/b)$$

To obtain the claim for randomised decision tree, we resort to Jensen's inequality:

$$\begin{aligned}
\mathrm{TV}(\mathrm{tran}(R, B^0), \mathrm{tran}(R, B^1)) &= \mathop{\mathbb{E}}_{D\sim R}\left[\mathrm{TV}(\mathrm{tran}(D, B^0), \mathrm{tran}(D, B^1))\right] \\
&\leq O(1) \cdot \mathop{\mathbb{E}}_{D\sim R}\left[b\sqrt{\mathrm{cost}(D, B)}\right] \\
&\leq O(1) \cdot b\sqrt{\mathrm{cost}(R, B)}
\end{aligned}$$

□

10 Lower bound for ν-stream

In this section, we finally prove that $\mathsf{ULR}(f) \geq \Omega(n^{5/4})$. Following Section 8, it suffices to prove (14). To keep things as general as possible, we will not work with $\tilde{\nu}$ directly but with an asymptotically equivalent

Stream	0	1	H
X^0	$.5 + b_X$	$.5 - b_X$	0
X^1	$.5 - b_X$	$.5 + b_X$	0
Y^0	$(.5 - b_Y)(1 - p_{\mathrm{H}})$	$(.5 + b_Y)(1 - p_{\mathrm{H}})$	p_{H}
Y^1	$(.5 + b_Y)(1 - p_{\mathrm{H}})$	$(.5 - b_Y)(1 - p_{\mathrm{H}})$	p_{H}
Z^0	$.5 - b_Y$	$.5 + b_Y$	0
Z^1	$.5 + b_Y$	$.5 - b_Y$	0
U	$.5$	$.5$	0

$$\begin{aligned}\nu^0 &:= (1-\lambda)X^0 + \lambda Y^0\\ \nu^1 &:= (1-\lambda)X^1 + \lambda Y^1\\ \nu &:= (\nu^0 + \nu^1)/2\end{aligned}$$

$$\begin{aligned}M^0 &:= (1-\lambda)X^0 + \lambda Z^0\\ M^1 &:= (1-\lambda)X^1 + \lambda Z^1\\ M &:= (M^0 + M^1)/2\end{aligned}$$

Table 1: A summary of the random streams used in this section. It should be read as e.g. the stream Z^1 has probability $1/2 - b_Y$ of producing a 1. The mixture parameter λ is set following (15) with b_X and b_Y so that $\lambda = \Theta(b_X/b_Y)$.

version ν and drop the tilde notation which is too heavy, so that e.g. Y is now a random stream over $\{0, 1, \mathrm{H}\}$ instead of an hyper-geometric distribution over $\{0,1\}^{Bn}$. All the random streams used for the remainder of the paper are summarised in Table 1. Note that the stream Y of Table 1 and the stream $\tilde{Y}$ of Section 8 are indeed asymptotically equivalent, as the proof of Theorem 25 shows. Let us now state our main technical result and show how to use it to get $\mathsf{ULR}(f) \geq \Omega(n^{5/4})$.

Theorem 24. *For any $b_X < o(b_Y) < o(1)$, $p_{\mathrm{H}} \in [3b_Y, 4b_Y]$ and deterministic decision tree D,*

$$\frac{\mathrm{cost}(D,\nu)}{\mathrm{TV}(\mathrm{tran}(D,\nu^0), \mathrm{tran}(D,\nu^1))} \geq \Omega(1) \cdot \min\left\{\frac{1}{b_X^{1/2} b_Y}, \frac{1}{b_X b_Y^{1/3}}\right\}$$

Theorem 25. $\mathsf{ULR}(f) \geq \Omega(n^{5/4})$

Proof. Following Section 8, in order to prove $\mathsf{ULR}(f) \geq \Omega(n^{5/4})$, it is sufficient to prove (14). To this end, fix any deterministic ternary decision tree D and let $\hat{b}_X$ and $\hat{b}_Y$ be the original parameters of Section 7, which were set to n^{-1} and $n^{-3/4}$ respectively. By setting $b_X := \hat{b}_X$, $b_Y := \hat{b}_Y/(1 - 4\hat{b}_Y)$ and $p_{\mathrm{H}} = 4\hat{b}_Y$, we have $\tilde{\nu} = \nu$, $b_Y = \Theta(\hat{b}_Y)$ and $p_{\mathrm{H}} \in [3b_Y, 4b_Y]$ (for n large enough), so that we may apply Theorem 24 directly:

$$\frac{\mathrm{cost}(D,\tilde{\nu})}{\mathrm{TV}(\mathrm{tran}(D,\tilde{\nu}^0), \mathrm{tran}(D,\tilde{\nu}^1))} = \frac{\mathrm{cost}(D,\nu)}{\mathrm{TV}(\mathrm{tran}(D,\nu^0), \mathrm{tran}(D,\nu^1))} \geq \Omega(1) \cdot \min\left\{\frac{1}{b_X^{1/2} b_Y}, \frac{1}{b_X b_Y^{1/3}}\right\}$$

This is $\Omega(n^{5/4})$ for our initial setting of $\hat{b}_X$ and $\hat{b}_Y$. □

10.1 Bias decomposition

A decision tree trying to distinguish ν^0 from ν^1 may pick one of several different strategies or even a mixture of those. To encompass all types of strategy, we will split the bias contribution of each leaf depending on their length and witnessing qualities. For instance, leaves that witness (i.e. $\mathrm{H} \in \ell$) have to solve the Y^0 versus Y^1 problem, whereas leaves that do not witness have the harder task of distinguishing M^0 from M^1. This effect however wears-off with an increasing depth: if some input reaches a very long non-witnessing leaf, then most likely the distribution was not Y to start with and this leaf can thus focus on distinguishing X^0 from X^1. To formalize this idea, let us partition the leaves of a decision tree D with $\mathcal{L}(D) = \mathcal{L}^{\mathrm{wit}}(D) \cup \mathcal{L}^{\overline{\mathrm{wit}}}(D)$ where $\mathcal{L}^{\mathrm{wit}}(D)$ contains all witnessing leaves and $\mathcal{L}^{\overline{\mathrm{wit}}}(D)$ the non-witnessing ones. For a pair of streams P^0 and P^1, we define the witnessing and non-witnessing bias with:

$$\begin{aligned}\mathrm{bias}^{\mathrm{wit}}(D, P^0, P^1) &:= \textstyle\sum_{\ell \in \mathcal{L}^{\mathrm{wit}}(D)} \left|P^0[\ell] - P^1[\ell]\right|\\ \mathrm{bias}^{\overline{\mathrm{wit}}}(D, P^0, P^1) &:= \textstyle\sum_{\ell \in \mathcal{L}^{\overline{\mathrm{wit}}}(D)} \left|P^0[\ell] - P^1[\ell]\right|\end{aligned}$$

In particular, $2 \cdot \mathrm{TV}(\mathrm{tran}(D, P^0), \mathrm{tran}(D, P^1)) = \mathrm{bias}^{\mathrm{wit}}(D, P^0, P^1) + \mathrm{bias}^{\overline{\mathrm{wit}}}(D, P^0, P^1)$. This distinction between witnessing and non-witnessing leaves allows for a quick proof of Theorem 24, provided that we have matching hardness result for the witnessing and non-witnessing trade-offs.

Proof of Theorem 24. Using Theorem 26 and Theorem 30 to bound the witnessing and non-witnessing trade-off,

$$\begin{aligned}\frac{\mathrm{cost}(D,\nu)}{\mathrm{TV}(\mathrm{tran}(D,\nu^0),\mathrm{tran}(D,\nu^1))} &\geq \min\left\{\frac{\mathrm{cost}(D,\nu)}{\mathrm{bias}^{\mathrm{wit}}(D,\nu^0,\nu^1)}, \frac{\mathrm{cost}(D,\nu)}{\mathrm{bias}^{\overline{\mathrm{wit}}}(D,\nu^0,\nu^1)}\right\} \\ &\geq \Omega(1)\cdot\min\left\{\frac{1}{b_X^{1/2}b_Y}, \frac{1}{b_X b_Y^{1/3}}\right\}\end{aligned}$$ □

In the following sections, it will be convenient to use $\Delta(\ell)$ for the absolute difference between the number of positive and negative literal in the conjunction ℓ. $\Delta(\ell)$ will be directly related to the distinguishing qualities of a leaf. For instance, if $\Delta(\ell) = 0$, then ℓ has no bias in distinguishing X^0 from X^1.

10.2 Trade-off for witnessing leaves

Since the distribution X^0 and X^1 never output H, witnessing leaves can only be reached if the distribution is Y^0 or Y^1. Hence, we can focus on bounding $\mathrm{bias}^{\mathrm{wit}}(D, Y^0, Y^1)$ as $\mathrm{bias}^{\mathrm{wit}}(D, \nu^0, \nu^1) = \lambda \cdot \mathrm{bias}^{\mathrm{wit}}(D, Y^0, Y^1)$. The main idea is to bound $\mathrm{bias}^{\mathrm{wit}}(D, Y^0, Y^1)$ using the hardness of distinguishing B^0 from B^1 with $b := b_Y$, with care needed as Y is ternary but B is of binary nature. Note that under Y, we expect to see a H after about $\Theta(1/p_{\mathrm{H}})$ queries and to simplify the argument, we will assume that any leaf witnesses a H if its length is larger than $\Theta(1/p_{\mathrm{H}})$. This relaxed notion of witnessing can be understood as *helping* the tree: if it has already made about $\Theta(1/p_{\mathrm{H}})$ queries, then we give it a H *for free*. To express this syntactically, let $L_{\mathrm{cut}} = \lceil 1/3p_{\mathrm{H}} \rceil$ and define the set of stopping nodes $\mathcal{S}(D) \subseteq \mathcal{L}(D)$ as any node which witnesses for the first time or has not witnessed but is at depth L_{cut}. Observe that the parent of any stopping node corresponds to a conjunction with no H and that any witnessing leaf has a unique stopping node as ancestor. For each $s \in \mathcal{S}(D)$, let D_s be the decision tree rooted at node s. With this notation in hand, we have:

$$\mathrm{bias}^{\mathrm{wit}}(D, Y^0, Y^1) \leq \sum\nolimits_{s\in\mathcal{S}(D)} \sum\nolimits_{\ell\in\mathcal{L}(D_s)} \left|Y^0[s\circ\ell] - Y^1[s\circ\ell]\right| \tag{16}$$

The inequality instead of strict equality comes from our *relaxed* notion of witnessing leaf, i.e. some large non-witnessing leaves are now also accounted for; but this is counterbalanced by the fact that those leaves tend to contribute greatly toward the expected cost. The connection between cost and bias will be made through $Y[\mathcal{S}(D)]$, the probability that a node from $\mathcal{S}(D)$ is reached by an input $x \sim Y$. Shallow trees will tend to have $Y[\mathcal{S}(D)]$ close to zero while tall trees will tend to have this probability close to 1. We now state and prove our bound for the witnessing trade-off.

Theorem 26. *For any deterministic decision tree D,*

$$\frac{\mathrm{cost}(D,\nu)}{\mathrm{bias}^{\mathrm{wit}}(D,\nu^0,\nu^1)} \geq \Omega(1)\cdot\min\left\{\frac{1}{b_X^{1/2}b_Y}, \frac{1}{b_X b_Y^{1/3}}\right\}$$

Proof. Using Lemma 27 and Lemma 29, we have:

$$\frac{\mathrm{cost}(D,\nu)}{\mathrm{bias}^{\mathrm{wit}}(D,\nu)} = \frac{\mathrm{cost}(D,\nu)}{\lambda\cdot\mathrm{bias}^{\mathrm{wit}}(D,Y)} \geq \Omega(1)\cdot\frac{\max\left\{\mathrm{cost}(D,X),\ \lambda\,\mathrm{cost}(D,Y),\ Y[\mathcal{S}(D)]/b_Y\right\}}{\lambda\cdot\max\left\{b_Y^{4/3}\,\mathrm{cost}(t,X),\ b_Y\sqrt{Y[\mathcal{S}(D)]\cdot\mathrm{cost}(D,Y)}\right\}}$$

If $b_Y^{4/3}\,\mathrm{cost}(D,X) \geq Y[\mathcal{S}(D)]\cdot\mathrm{cost}(D,Y)$, then we have:

$$\frac{\mathrm{cost}(D,\nu)}{\mathrm{bias}^{\mathrm{wit}}(D,\nu)} \geq \Omega(1)\cdot\frac{\mathrm{cost}(D,X)}{\lambda b_Y^{4/3}\,\mathrm{cost}(D,X)} = \Omega\left(\frac{1}{b_X b_Y^{1/3}}\right)$$

The last inequality holds because $\lambda = \Theta(b_X/b_Y)$. If the other case holds, then we have:

$$\frac{\mathrm{cost}(D,\nu)}{\overline{\mathrm{bias}^{\mathrm{wit}}}(D,\nu)} \geq \Omega(1)\cdot\max\left\{\frac{\sqrt{\mathrm{cost}(D,Y)}}{b_Y\sqrt{Y[\mathcal{S}(D)]}}, \frac{\sqrt{Y[\mathcal{S}(D)]}}{\lambda b_Y^2\sqrt{\mathrm{cost}(D,Y)}}\right\} \geq \Omega\left(\frac{1}{b_X^{1/2}b_Y}\right) \qquad \square$$

Lemma 27. *For any deterministic decision tree D:*

$$\mathrm{bias}^{\mathrm{wit}}(D,Y^0,Y^1) \leq O(1)\cdot\max\left\{b_Y\sqrt{\mathrm{cost}(D,Y)\cdot Y[\mathcal{S}(D)]},\, b_Y^{4/3}\,\mathrm{cost}(D,X)\right\}$$

Proof. Continuing on (16), we further split the bias contribution by leaves that are stopping nodes and leaves which have an ancestor stopping node.

$$\begin{aligned}
\mathrm{bias}^{\mathrm{wit}}(D,Y^0,Y^1) &\leq \sum_{s\in\mathcal{S}(D)}\sum_{\ell\in\mathcal{L}(D_s)}\left|Y^0[s\circ\ell]-Y^1[s\circ\ell]\right| \\
&= \sum_{s\in\mathcal{S}(D)}\sum_{\ell\in\mathcal{L}(D_s)}\left|Y^0[s]Y^0[\ell]-Y^1[s]Y^1[\ell]\right| \\
&= \sum_{s\in\mathcal{S}(D)}\sum_{\ell\in\mathcal{L}(D_s)} 2Y[s]\left|\left(\frac{1}{2}+\delta(s)\right)Y^0[\ell]-\left(\frac{1}{2}-\delta(s)\right)Y^1[\ell]\right| \qquad \delta(s) := \frac{Y^0[s]-Y^1[s]}{4Y[s]} \\
&\leq \sum_{s\in\mathcal{S}(D)}\sum_{\ell\in\mathcal{L}(D_s)} Y[s]\cdot\left|Y^0[\ell]-Y^1[\ell]\right| + 2Y[s]\cdot|\delta(s)|\cdot\left(Y^0[\ell]+Y^1[\ell]\right) \\
&= \sum_{s\in\mathcal{S}(D)}\sum_{\ell\in\mathcal{L}(D_s)} Y[s]\cdot\left|Y^0[\ell]-Y^1[\ell]\right| + \underbrace{\textstyle\sum_{s\in\mathcal{S}(D)}\left|Y^0[s]-Y^1[s]\right|}_{\text{cash-out bias } b_C}
\end{aligned}$$

Lemma 28 shows that $b_C \leq O(1)\cdot b_Y^{4/3}\,\mathrm{cost}(D,X)$ and we now argue that the first sum is bounded by $O(1)\cdot b_Y\sqrt{\mathrm{cost}(D,Y)\cdot Y[\mathcal{S}(D)]}$, thus finishing the claim. To do so, fix some $s\in\mathcal{S}(D)$ and let us assume that D_s is optimally labeled with $\{Y^0,Y^1\}$ so that:

$$\textstyle\sum_{\ell\in\mathcal{L}(D_s)}\left|Y^0[\ell]-Y^1[\ell]\right| = 2\cdot\left(\Pr_{x\sim Y^0}[D_s(x)=Y^0]-\Pr_{x\sim Y^1}[D_s(x)=Y^0]\right)$$

Now, D_s can be transformed into a *randomised* decision tree R_s that solves Z^0 versus Z^1: R_s runs D_s as usual and swap a H for query answers in $\{0,1\}$ with probability p_{H}. R_s further has Y^0 leaves relabeled by Z^0 and Y^1 leaves by Z^1, implying:

$$\begin{aligned}
\textstyle\sum_{\ell\in\mathcal{L}(D_s)}\left|Y^0[\ell]-Y^1[\ell]\right| &= 2\cdot\left(\Pr_{x\sim Z^0}[R_s(x)=Z^0]-\Pr_{x\sim Z^1}[R_s(x)=Z^0]\right) \\
&= 2\cdot\mathrm{TV}(\mathrm{tran}(R_s,Z^0),\mathrm{tran}(R_s,Z^1)) \\
&\leq O(1)\cdot b_Y\sqrt{\mathrm{cost}(R_s,Z)} \qquad \text{(by Theorem 22 with } b := b_Y\text{)} \\
&= O(1)\cdot b_Y\sqrt{\mathrm{cost}(D_s,Y)}
\end{aligned}$$

Applying this bound to each $s\in\mathcal{S}(D)$, we get the desired bound on the left sum:

$$\begin{aligned}
\sum_{s\in\mathcal{S}(D)}\sum_{\ell\in\mathcal{L}(D_s)} Y[s]\cdot\left|Y^0[\ell]-Y^1[\ell]\right| &\leq O(1)\cdot b_Y\sum_{s\in\mathcal{S}(D)} Y[s]\sqrt{\mathrm{cost}(D_s,Y)} \\
&\leq O(1)\cdot b_Y\sqrt{Y[\mathcal{S}(D)]}\cdot\sqrt{\sum_{s\in\mathcal{S}(D)} Y[s]\cdot\mathrm{cost}(D_s,Y)} \\
&\leq O(1)\cdot b_Y\sqrt{Y[\mathcal{S}(D)]}\cdot\sqrt{\mathrm{cost}(D,Y)}
\end{aligned}$$

Where Cauchy-Schwarz inequality was used for the second inequality. □

Lemma 28. *Following the notation of Lemma 27, $b_C \leq O(1)\cdot b_Y^{4/3}\,\mathrm{cost}(D,X)$ for any deterministic decision tree D.*

Proof. Without loss of generality, we may assume that D has depth bounded by L_{cut} as there is no stopping node with depth greater than that. As a first step, we partition $\mathcal{S}(D) = \mathcal{S}^{\text{H}} \cup \mathcal{S}^{\text{cut}}$ where $\mathcal{S}^{\text{H}}$ contains all the witnessing stopping nodes in D and $\mathcal{S}^{\text{cut}}$ the nodes that are stopping only because of their length being L_{cut}. Note that any $s \in \mathcal{S}^{\text{H}}$ has parent node $p(s)$ representing a conjunction with no H. Therefore, we may substitute Z to Y as follows:

$$\begin{aligned} b_C &= \sum\nolimits_{s\in\mathcal{S}^{\text{cut}}} \left|Y^0[s] - Y^1[s]\right| + \sum\nolimits_{s\in\mathcal{S}^{\text{H}}} \left|Y^0[s] - Y^1[s]\right| \\ &\le \sum\nolimits_{s\in\mathcal{S}^{\text{cut}}} \left|Z^0[s] - Z^1[s]\right| + p_{\text{H}} \sum\nolimits_{s\in\mathcal{S}^{\text{H}}} \left|Z^0[p(s)] - Z^1[p(s)]\right| \end{aligned}$$

The second sum bears a nice interpretation. Indeed, if D' is a copy of D that stops at depth $L_{\text{cut}} - 1$ and has all witnessing paths removed then the second sum is equal to $\sum_{n\in\mathcal{N}(D')} \left|Z^0[n] - Z^1[n]\right|$, i.e. the bias D' gets in distinguishing Z^0 from Z^1 while having the ability to *cash-out* the current bias it has at each node. Now, if D' happens to be close to the complete binary tree, then we may apply the hardness of distinguishing Z^0 from Z^1 at each level and get a global bound on the cash-out bias. This approach will however fail if the tree is largely unbalanced and we thus develop a more robust argument, which we re-use in later results. We first partition $\mathcal{S}^{\text{H}} = \bigcup \mathcal{S}^k$ where $\mathcal{S}^k := \{s \in \mathcal{S}^{\text{H}} : |s| = k\}$ for $k \in [L_{\text{cut}}]$. Fix now some $k \in [L_{\text{cut}}]$ and observe that,

$$\begin{aligned} \sum_{s\in\mathcal{S}^k} \left|Z^0[p(s)] - Z^1[p(s)]\right| &\le 8b_Y \sum_{s\in\mathcal{S}^k} \Delta(p(s)) Z[p(s)] && \text{(by Lemma 34)} \\ &\le 16 b_Y k^{2/3} \sum_{\Delta(s)\le 2k^{2/3}} Z[p(s)] + 8b_Y k \sum_{\Delta(s)\ge 2k^{2/3}} Z[p(s)] \\ &\le 16 b_Y k^{2/3} \sum_{\Delta(s)\le 2k^{2/3}} Z[p(s)] + 16 b_Y k e^{-k^{1/3}/48} && \text{(by Lemma 45)} \\ &\le 16 b_Y k^{2/3} \sum_{\ell\in\mathcal{L}(D):|\ell|\ge k} Z[\ell] + 16 b_Y k e^{-k^{1/3}/48} \\ &\le 16 b_Y k^{-1/3} \operatorname{cost}(D, Z) + 16 b_Y k e^{-k^{1/3}/48} && \text{(Markov's inequality)} \\ &\le O(1) \cdot b_Y k^{-1/3} \operatorname{cost}(D, Z) \end{aligned}$$

We made the arbitrary choice to split between small and large Δ with cutoff parameter $2k^{2/3}$. Setting it to the limiting $k^{1/2}\text{polylog}(n)$ would have slightly improved the LR ratio against ν but this would ultimately yield $\lambda = \Theta(1/\text{polylog}(n))$, thus *worsening* the step in which we reduce to decision trees not solving hard blocks (see Section 8). Observing that the above chain of inequalities also holds for $\mathcal{S}^{\text{cut}}$ with level $k = L_{\text{cut}}$, we have:

$$\begin{aligned} b_C &\le \sum_{s\in\mathcal{S}^{\text{cut}}} \left|Z^0[s] - Z^1[s]\right| + p_{\text{H}} \sum_{s\in\mathcal{S}^*} \left|Z^0[p(s)] - Z^1[p(s)]\right| \\ &\le O(1) \cdot b_Y L_{\text{cut}}^{-1/3} \operatorname{cost}(D, Z) + O(1) \cdot p_* b_Y \operatorname{cost}(D, Z) \sum\nolimits_{k\in[L_{\text{cut}}]} k^{-1/3} \\ &\le O(1) \cdot b_Y L_{\text{cut}}^{-1/3} \operatorname{cost}(D, Z) + O(1) \cdot p_* b_Y {L_{\text{cut}}}^{2/3} \operatorname{cost}(D, Z) \\ &\le O(1) \cdot b_Y^{4/3} \operatorname{cost}(t, Z) && (L_{\text{cut}} = \lceil 1/3p_* \rceil \text{ and } p_* = \Theta(b_Y)) \\ &\le O(1) \cdot b_Y^{4/3} \operatorname{cost}(t, X) && \text{(by Lemma 33)} \quad \square \end{aligned}$$

Lemma 29. *For any deterministic decision tree D, $\operatorname{cost}(D, \nu) \ge \Omega(Y[\mathcal{S}(D)]/p_{\text{H}})$.*

Proof. Without loss of generality, we may assume that D stops whenever it reaches a stopping node. Let us first argue that $\operatorname{cost}(D, \nu) \ge \Omega(\operatorname{cost}(D, Y))$. Fix $d \in \{0, 1\}$ and observe that sampling from Y^d is the same as sampling from Z^d while *salting* the query answers by replacing them with a star with independent probability p_{H} so that:

$$\operatorname{cost}(D, Y^d) = \mathop{\mathbb{E}}_{x\sim Z^d}\left[\mathop{\mathbb{E}}_{\text{salt}}\left[q(D, \text{salt}(x))\right]\right] \le \mathop{\mathbb{E}}_{x\sim Z^d}\left[q(D, x)\right] = \operatorname{cost}(D, Z^d) \le 78 \operatorname{cost}(D, X) \tag{17}$$

The first inequality stems from the fact that flipping some answer with a star makes the leaf witness and D thus directly stops. The last one is due by Lemma 33, recalling that D has depth bounded by L_{cut}. Using (17) and the definition of ν, we have $\text{cost}(D, \nu) \geq \Omega(\text{cost}(D, Y))$. Recall that there are two types of stopping nodes in $\mathcal{S}(D)$. The first type are nodes that witnesses for the first time and the second type are nodes that never witness but have reached depth L_{cut}. We split $Y[\mathcal{S}(D)]$, according to both type:

$$p_1 \coloneqq \Pr_{x\sim Y}[D \text{ stops on } \textsc{h}] \quad \text{and} \quad p_2 \coloneqq \Pr_{x\sim Y}[D \text{ stops because of } L_{\text{cut}}]$$

We may thus write $Y[\mathcal{S}[D]] = p_1 + p_2$, as a simple sanity check, note that it is possible to have $Y[\mathcal{S}(D)] \ll 1$, e.g. if D has many small non-witnessing leaves. If $p_2 \geq p_1$, then $p_2 \geq Y[\mathcal{S}(D)]/2$ and so:

$$\text{cost}(D, Y) \geq \sum\nolimits_{|\ell| = L_{\text{cut}}} |\ell| Y[\ell] = p_2 L_{\text{cut}} \geq \Omega(Y[\mathcal{S}(D)]/p_{\textsc{h}})$$

Now, if $p_1 \geq Y[\mathcal{S}(D)]$, we let D' be a decision tree that runs D in turn until it witnesses a $\textsc{h}$ in the stream but for at most $\lceil 2/p_1 \rceil$ times. We ensure that the runs are independent by offsetting the query indices by a multiple of a large number (e.g. $10L_{\text{cut}}$). With that number of runs, D' has a constant probability of witnessing:

$$\Pr_{x\sim Y}[D' \text{ witnesses}] = 1 - (1-p_1)^{\lceil 2/p_1 \rceil} \geq 1 - e^{-2} \geq 3/4$$

Observe further that D' never queries after witnessing and that $\text{cost}(D, Y) \geq \Omega(\text{cost}(D', Y) Y[\mathcal{S}(D)])$. Fix now $\mathcal{L}^1 = \{\ell \in \mathcal{L}(D') : \textsc{h} \in \ell \text{ and } |\ell| \leq L_{\text{cut}}\}$ and $\mathcal{L}_2 = \{\ell \in \mathcal{L}(D') : \textsc{h} \in \ell \text{ and } |\ell| > L_{\text{cut}}\}$ and note that $Y[\mathcal{L}^1] + Y[\mathcal{L}^2] \geq 3/4$. Because leaves of D' can only have a $\textsc{h}$ as their last literal we have $Y[\ell] = (1-p_{\textsc{h}})^{|\ell|} p_{\textsc{h}} Z[p(\ell)]$ where $p(\ell)$ is the parent node of ℓ and hence:

$$Y[\mathcal{L}^1] = \sum_{k=1}^{L_{\text{cut}}} \sum_{\substack{\ell \in \mathcal{L}_1: \\ |\ell| = k}} Y[\ell] \leq 2p_{\textsc{h}} \sum_{k=1}^{L_{\text{cut}}} (1-p_{\textsc{h}})^{k-1} = 1 - (1-p_{\textsc{h}})^{L_{\text{cut}}} \leq p_{\textsc{h}} L_{\text{cut}} \leq 2/3$$

This shows that $Y[\mathcal{L}^2] \geq 1/12$ and thus:

$$\text{cost}(D, Y) \geq \Omega(1) \cdot Y[\mathcal{S}(D)]\, \text{cost}(D', Y) \geq \Omega(1) \cdot Y[\mathcal{S}(D)] Y[\mathcal{L}^2] L_{\text{cut}} \geq \Omega(Y[\mathcal{S}(D)/p_{\textsc{h}}]) \qquad \square$$

10.3 Trade-off for non-witnessing leaves

Non-witnessing leaves can be reached both by the X and Y distribution and as such have the harder task of distinguishing ν^0 from ν^1, unlike witnessing leaves that only need to solve Y^0 versus Y^1. As previously noted, this effect wears off following the depth of the tree: if the unknown stream reaches a long leaf with no $\textsc{h}$, then most likely the stream is part of $\{X^0, X^1\}$ and not $\{Y^0, Y^1\}$. To account for this, we will break the analysis into small and large leaves with cutoff parameter $L_{\text{cut}} = \lceil 1/b_Y \rceil$. For a decision tree D, let $\mathcal{L}^{\text{small}}(D) = \{\ell \in \mathcal{L}^{\overline{\text{wit}}}(D) : |\ell| \leq L_{\text{cut}}\}$ and $\mathcal{L}^{\text{large}}(D) = \{\ell \in \mathcal{L}^{\overline{\text{wit}}}(D) : |\ell| > L_{\text{cut}}\}$. We will argue that the bias brought by $\mathcal{L}^{\text{small}}(D)$ is capped by the hardness of the M^0 versus M^1 problem (see Section 9.1) and that the bias brought by $\mathcal{L}^{\text{large}}(D)$ can bounded using the B^0 versus B^1 problem with $b \coloneqq b_X$ (see Section 9.2).

Theorem 30. *For any deterministic decision tree D, $\text{cost}(D, \nu)/\text{bias}^{\overline{\text{wit}}}(D, \nu^0, \nu^1) \geq \Omega(1/b_X b_Y^{1/3})$*

Proof. Using the notation introduced above, we can split $\text{bias}^{\overline{\text{wit}}}(D, \nu^0, \nu^1)$ with:

$$\begin{aligned} \text{bias}^{\overline{\text{wit}}}(D, \nu^0, \nu^1) &= \sum_{\ell \in \mathcal{L}^{\text{small}}} \big|\nu^0[\ell] - \nu^1[\ell]\big| + \sum_{\ell \in \mathcal{L}^{\text{large}}} \big|\nu^0[\ell] - \nu^1[\ell]\big| \\ &\leq \sum_{\ell \in \mathcal{L}^{\text{small}}} \big|\nu^0[\ell] - \nu^1[\ell]\big| + O(1) \cdot \sum_{\ell \in \mathcal{L}^{\text{large}}} \big|X^0[\ell] - X^1[\ell]\big| + b_Y^3 \qquad \text{(by Lemma 35)} \end{aligned}$$

Lemma 31 shows that the the second sum is bounded by $O(\text{cost}(D, X) b_X b_Y^{1/3})$ so that we only need to show that the first sum is also bounded by the same amount, which we do next. To analyse this sum, we may assume without loss of generality that D is a binary decision tree of maximum depth L_{cut}. Indeed, $\mathcal{L}^{\text{small}}$

contains small non-witnessing leaves only. This implies that for all $\ell \in \mathcal{L}(D)$, $Y[\ell] = (1 - p_{\text{H}})^{|\ell|} Z[\ell]$ and so we may further break the sum into the contribution of M^0 versus M^1 and the one of Z^0 versus Z^1:

$$\begin{aligned}
\left|\nu^0[\ell] - \nu^1[\ell]\right| &= \left|M^0[\ell] - \lambda Z^0[\ell] + \lambda Y^0[\ell] - M^1[\ell] + \lambda Z^1[\ell] - \lambda Y^1[\ell]\right| \\
&\leq \left|M^0[\ell] - M^1[\ell]\right| + \lambda \cdot \left|Y^0[\ell] - Y^1[\ell] - Z^0[\ell] + Z^1[\ell]\right| \\
&= \left|M^0[\ell] - M^1[\ell]\right| + \lambda \cdot (1 - (1 - p_{\text{H}})^{|\ell|}) \cdot \left|Z^0[\ell] - Z^1[\ell]\right| \\
&\leq \left|M^0[\ell] - M^1[\ell]\right| + \lambda p_{\text{H}} |\ell| \cdot \left|Z^0[\ell] - Z^1[\ell]\right| && \text{(by Lemma 44)}
\end{aligned}$$

Using this insight, we can finally exploit the hardness of distinguishing M^0 from M^1:

$$\begin{aligned}
\sum_{\ell \in \mathcal{L}^{\text{small}}} \left|\nu^0[\ell] - \nu^1[\ell]\right| &\leq \sum_{\ell \in \mathcal{L}(D)} \left|M^0[\ell] - M^1[\ell]\right| + \lambda p_{\text{H}} \sum_{\ell \in \mathcal{L}(D)} |\ell| \cdot \left|Z^0[\ell] - Z^1[\ell]\right| \\
&= 2 \cdot \text{TV}(\text{tran}(D, M^0), \text{tran}(D, M^1)) + \lambda p_{\text{H}} \sum_{\ell \in \mathcal{L}(D)} |\ell| \cdot \left|Z^0[\ell] - Z^1[\ell]\right| \\
&\leq O(b_X b_Y \,\text{cost}(D, M)) + \lambda p_{\text{H}} \sum_{\ell \in \mathcal{L}(D)} |\ell| \cdot \left|Z^0[\ell] - Z^1[\ell]\right| && \text{(by Theorem 21)} \\
&\leq O(b_X b_Y \,\text{cost}(D, X)) + \lambda p_{\text{H}} \underbrace{\textstyle\sum_{\ell \in \mathcal{L}(D)} |\ell| \cdot \left|Z^0[\ell] - Z^1[\ell]\right|}_{\text{weighted bias } b_W} && \text{(by Lemma 33)}
\end{aligned}$$

Since $\lambda = \Theta(b_X / b_Y)$ and $p_{\text{H}} = \Theta(b_Y)$, the only thing left to prove is that $b_W \leq O(b_Y^{1/3} \,\text{cost}(D, X))$. We resort to an analysis similar to the one employed in Lemma 28, except that this time we need to take into account the size of the leaf. To that end, let $\mathcal{L}^k \coloneqq \{\ell \in \mathcal{L}(D) : |\ell| = k\}$ for $k \in [L_{\text{cut}}]$ and let us bound the bias brought by level k:

$$\begin{aligned}
\textstyle\sum_{\ell \in \mathcal{L}^k} k \cdot \left|Z^0[\ell] - Z^1[\ell]\right| &\leq 8 b_Y k \textstyle\sum_{\ell \in \mathcal{L}^k} \Delta(\ell) Z[\ell] && \text{(by Lemma 34)} \\
&\leq 16 b_Y k^{5/3} \textstyle\sum_{\Delta(\ell) \leq 2k^{2/3}} Z[\ell] + 8 b_Y k^2 \textstyle\sum_{\Delta(\ell) \geq 2k^{2/3}} Z[\ell] \\
&\leq 16 b_Y k^{5/3} \textstyle\sum_{\Delta(\ell) \leq 2k^{2/3}} Z[\ell] + 16 b_Y k^2 e^{-k^{1/3}/48} && \text{(by Lemma 45)}
\end{aligned}$$

Recalling that D has depth bounded by $L_{\text{cut}} = \lceil 1/b_Y \rceil$, we have $k^{5/3} \leq b_Y^{-2/3} k$ and hence:

$$\begin{aligned}
b_W &= \textstyle\sum_{k \in [L_{\text{cut}}]} \sum_{\ell \in \mathcal{L}^k} k \cdot \left|Z^0[\ell] - Z^1[\ell]\right| \\
&\leq 16 b_Y^{1/3} \textstyle\sum_{\ell \in \mathcal{L}(t)} |\ell| \cdot Z[\ell] + 16 b_Y \sum_{k \in [L_{\text{cut}}]} k^2 e^{-k^{1/3}/48} \\
&\leq 16 b_Y^{1/3} \,\text{cost}(D, Z) + O(b_Y) && \text{(by ratio test)} \\
&\leq O(1) \cdot b_Y^{1/3} \,\text{cost}(D, X) && \text{(by Lemma 33)} \quad \square
\end{aligned}$$

Lemma 31. *For any deterministic decision tree D, $\sum_{\ell \in \mathcal{L}^{\text{large}}(D)} \left|X^0[\ell] - X^1[\ell]\right| \leq O(1) \cdot b_X b_Y^{1/3} \,\text{cost}(D, X)$*

Proof. We may assume without loss of generality that D is over $\{0, 1\}$ only as any witnessing conjunction ℓ has $\left|X^0[\ell] - X^1[\ell]\right| = 0$. We let $\mathcal{C} \coloneqq \{c \in \mathcal{N}(t) : |c| = L_{\text{cut}}\}$ be the set of node at height L_{cut} and for each $c \in \mathcal{C}$, define D_c to be the sub-tree of D rooted at c. Re-cycling the decomposition used in the proof of Lemma 27:

$$\textstyle\sum_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| \leq \underbrace{\textstyle\sum_{c \in \mathcal{C}} X[c] \sum_{\ell \in \mathcal{L}(D_c)} \left|X^0[\ell] - X^1[\ell]\right|}_{\text{large leaves bias } b_L} + \underbrace{\textstyle\sum_{c \in \mathcal{C}} \left|X^0[c] - X^1[c]\right|}_{\text{cut-off bias } b_C}$$

We show next that $b_L, b_C \leq O(b_X b_Y^{1/3} \,\text{cost}(D, X))$. To get a bound on b_L, observe that for any $c \in \mathcal{C}$, $\sum_{\ell \in \mathcal{L}(D_c)} \left|X^0[\ell] - X^1[\ell]\right| = 2 \cdot \text{TV}(\text{tran}(D_c, X^0), \text{tran}(D_c, X^1))$ which can be bounded by $O(b_X \sqrt{\text{cost}(D_c, X)})$

using Theorem 22 with $b \coloneqq b_Y$ so that:

$$\begin{aligned}
b_L &\le O(1) \cdot b_X \sum_{c \in \mathcal{C}} X[c] \sqrt{\operatorname{cost}(D_c, X)} \\
&= O(1) \cdot b_X \sqrt{X[\mathcal{C}]} \cdot \sqrt{\textstyle\sum_{c \in \mathcal{C}} X[c] \cdot \operatorname{cost}(D_c, X)} && \text{(by Cauchy-Schwarz inequality)} \\
&\le O(1) \cdot b_X b_Y^{1/2} \sqrt{\operatorname{cost}(D, X)} \cdot \sqrt{\textstyle\sum_{c \in \mathcal{C}} X[c] \cdot \operatorname{cost}(D_c, X)} && \text{(by Markov's inequality and } L_{\text{cut}} = \lceil 1/b_Y \rceil) \\
&\le O(1) \cdot b_X b_Y^{1/2} \operatorname{cost}(D, X)
\end{aligned}$$

We can finally bound b_C, using the fact that all $c \in \mathcal{C}$ have length $L_{\text{cut}} = \lceil 1/b_Y \rceil$.

$$\begin{aligned}
b_C &\le O(1) \cdot b_X \textstyle\sum_{c \in \mathcal{C}} \Delta(c) X[c] && \text{(by Lemma 34)} \\
&= O(1) \cdot b_X L_{\text{cut}}^{2/3} \textstyle\sum_{\Delta(c) \le 2L_{\text{cut}}^{2/3}} X[c] + O(1) \cdot b_X L_{\text{cut}} \textstyle\sum_{\Delta(c) \ge 2L_{\text{cut}}^{2/3}} X[c] \\
&= O(1) \cdot b_X L_{\text{cut}}^{2/3} \textstyle\sum_{\Delta(c) \le 2L_{\text{cut}}^{2/3}} X[c] + O(1) \cdot b_X L_{\text{cut}} e^{-{L_{\text{cut}}}^{1/3}/48} && \text{(by Lemma 45)} \\
&\le O(1) \cdot 2b_X L_{\text{cut}}^{2/3} \textstyle\sum_{\Delta(c) \le 2L_{\text{cut}}^{2/3}} X[c] + O(1) \cdot b_X {L_{\text{cut}}}^{-1/3} \\
&\le O(1) \cdot b_X {L_{\text{cut}}}^{-1/3} \operatorname{cost}(D, X) + O(1) \cdot b_X {L_{\text{cut}}}^{-1/3} && \text{(by Markov's inequality)} \\
&\le O(1) \cdot b_X b_Y^{1/3} \operatorname{cost}(D, X)
\end{aligned}$$

□

11 Technical lemmas

11.1 Some corruption bounds

Lemma 32. *For any non-witnessing conjunction ℓ, $M^0[\ell]/U[\ell] \ge 1 - 8|\ell| b_X b_Y$*

Proof. Fix $k \coloneqq |\ell|$ and let ℓ have $k/2 + q$ positive variables for some $q \in [-k/2, k/2]$. As a function of q, the ratio $M^0[\ell]/U[\ell]$ can be expressed as:

$$r(q) \coloneqq (1 - \lambda)\left(1 - 4b_X^2\right)^{k/2} \left(\frac{1 - 2b_X}{1 + 2b_X}\right)^q + \lambda \left(1 - 4b_Y^2\right)^{k/2} \left(\frac{1 + 2b_Y}{1 - 2b_Y}\right)^q$$

Using Lemma 39, the minimizer of the ratio (extended to $\mathbb{R}$) is $q^\star \in [kb_Y/7, kb_Y]$, so we may lower-bound the above with:

$$\begin{aligned}
\min_{q \in [-k/2, k/2]} r(q) &\ge (1 - \lambda)\left(1 - 4b_X^2\right)^{k/2} \left(\frac{1 - 2b_X}{1 + 2b_X}\right)^{q^\star} + \lambda \left(1 - 4b_Y^2\right)^{k/2} \left(\frac{1 + 2b_Y}{1 - 2b_Y}\right)^{q^\star} \\
&\ge (1 - \lambda)\left(1 - 4b_X^2\right)^{k/2} (1 - 4b_X)^{kb_Y} + \lambda \left(1 - 4b_Y^2\right)^{k/2} \\
&\ge (1 - \lambda)\left(1 - 2kb_X^2\right)(1 - 4kb_X b_Y) + \lambda \left(1 - 2kb_Y^2\right) \\
&\ge (1 - \lambda)(1 - 6kb_X b_Y) + \lambda \left(1 - kb_Y^2\right) \\
&\ge 1 - 6kb_X b_Y - \lambda k b_Y^2
\end{aligned}$$

Recalling that $\lambda \le 2b_X/b_Y$, we get that $M^0[\ell]/U[\ell] \ge 1 - 8|\ell| b_X b_Y$, as desired. □

Lemma 33 (Z vs. X)**.** *For any non-witnessing conjunction ℓ with $|\ell| \le 1/b_Y$, $Z^d[\ell] \le 78X[\ell]$ for $d \in \{0, 1\}$.*

Proof. We prove the claim for $d = 0$, the other case being symmetric. Fix some conjunction ℓ and let $L \coloneqq \lceil 1/b_Y \rceil$. We first demonstrate that $Z^0[\ell] \le 26U[\ell]$:

$$\frac{Z^0[\ell]}{U[\ell]} \le (1 + 2b_Y)^L = \sum_{k=0}^{L} \binom{L}{k} (2b_Y)^k \le 1 + \sum_{k=1}^{L} \left(\frac{2eLb_Y}{k}\right)^k \le 1 + \sum_{k=1}^{\infty} \left(\frac{2e}{k}\right)^k \le 26$$

Where the first inequality is due to the worst-case consisting of a conjunction with L positive literal and zero negative ones. We now show that $U[\ell] \leq 3X[\ell]$, thus finishing the claim. Observe that the conjunction maximizing the ratio $X[\ell]/U[\ell]$ has $L/2$ positive literal and $L/2$ negative one, hence:

$$\frac{X[\ell]}{U[\ell]} \geq \frac{X^1[\ell]}{2U[\ell]} \overset{(b)}{\geq} \frac{(1+2b_X)^{L/2}(1-2b_X)^{L/2}}{2} = \frac{1-2b_X^2 L}{2} \geq \frac{1}{3} \qquad \square$$

Lemma 34 (Z^0 vs. Z^1)**.** *For any leaf ℓ, $\left|Z^0[\ell] - Z^1[\ell]\right| \leq 8b_Y \Delta(\ell) Z[\ell]$*

Proof. If ℓ is witnessing, the claim is trivially true so let us assume that $\textsc{h} \notin \ell$. By symmetry, we may assume that ℓ has more negative literals than positive ones so that $Z^1[\ell] \geq Z^0[\ell]$, thus:

$$\left|Z^0[\ell] - Z^1[\ell]\right| = Z^1[\ell] - Z^0[\ell] \leq 2Z[\ell] \cdot \left(1 - Z^0[\ell]/Z^1[\ell]\right)$$

Now, letting q_0, respectively q_1 be the number of negative, respectively positive, literals in ℓ and using the definition of Z^0 and Z^1, we have:

$$\frac{Z^0[\ell]}{Z^1[\ell]} = \frac{(1+2b_Y)^{q_0}(1-2b_Y)^{q_1}}{(1+2b_Y)^{q_1}(1-2b_Y)^{q_0}} = \left(\frac{1-2b_Y}{1+2b_Y}\right)^{\Delta(\ell)} \geq (1-4b_Y)^{\Delta(\ell)}$$

Combining both observations, we get:

$$\left|Z^0[\ell] - Z^1[\ell]\right| \leq 2Z[\ell] \cdot \left(1 - (1-4b_Y)^{\Delta(\ell)}\right) \leq 8b_Y \Delta(\ell) Z[\ell] \qquad \square$$

11.2 Some bias transfers

Lemma 35 (ν to X bias transfer)**.** *Letting $\mathcal{L}^{\text{large}}$ be defined as in Theorem 30, it holds that:*

$$\sum\nolimits_{\ell \in \mathcal{L}^{\text{large}}} \left|\nu^0[\ell] - \nu^1[\ell]\right| \leq b_Y^2 + O(1) \cdot \sum\nolimits_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right|$$

Proof. Recall that $\mathcal{L}^{\text{large}}$ only contains non-witnessing leaf of size at least $L_{\text{cut}} = \lceil 1/b_Y \rceil$. Using the definition of ν and the triangle inequality, we have:

$$\sum\nolimits_{\ell \in \mathcal{L}^{\text{large}}} \left|\nu^0[\ell] - \nu^1[\ell]\right| \leq \sum\nolimits_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| + \lambda \sum\nolimits_{\ell \in \mathcal{L}^{\text{large}}} \left|Y^0[\ell] - Y^1[\ell]\right|$$

Hence, we need to focus on the second sum only. To bound it, we partition $\mathcal{L}^{\text{large}}$ into balanced and unbalanced leaves:

$$\mathcal{L}^{\text{large}} = \bigcup_{k=L_{\text{cut}}}^{\infty} \mathcal{B}^k \cup \mathcal{U}^k \quad \text{where} \quad \begin{array}{l} \mathcal{B}^k := \left\{\ell \in \mathcal{L}^{\text{large}} : |\ell| = k \quad \text{and} \quad \Delta(\ell) \leq |\ell|/2\right\} \\ \mathcal{U}^k := \left\{\ell \in \mathcal{L}^{\text{large}} : |\ell| = k \quad \text{and} \quad \Delta(\ell) > |\ell|/2\right\} \end{array}$$

For leaves in $\mathcal{B}^k$, one can apply Lemma 36 to get that $\lambda \left|Y^0[\ell] - Y^1[\ell]\right| \leq O(1) \cdot \left|X^0[\ell] - X^1[\ell]\right|$ so that we have:

$$\begin{aligned} \lambda \sum_{\ell \in \mathcal{L}^{\text{large}}} \left|Y^0[\ell] - Y^1[\ell]\right| &\leq O(1) \cdot \sum_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| + \lambda \sum_{k=L_{\text{cut}}}^{\infty} \sum_{\ell \in \mathcal{U}^k} \left|Y^0[\ell] - Y^1[\ell]\right| \\ &\leq O(1) \cdot \sum_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| + \sum_{k=L_{\text{cut}}}^{\infty} \sum_{\ell \in \mathcal{U}^k} Y[\ell] \\ &\leq O(1) \cdot \sum_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| + \sum_{k=L_{\text{cut}}}^{\infty} e^{-k/100} && (*) \\ &\leq O(1) \cdot \sum_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| + 2e^{-L_{\text{cut}}/100} \\ &\leq O(1) \cdot \sum_{\ell \in \mathcal{L}^{\text{large}}} \left|X^0[\ell] - X^1[\ell]\right| + b_Y^2 && \text{(for } b_Y \text{ small enough)} \end{aligned}$$

Where (*) is obtained by a slight modification of the proof of Lemma 45. □

Lemma 36. *For any non-witnessing leaf ℓ with $|\ell| \geq 1/b_Y$ and $\Delta(\ell) \leq |\ell|/2$,*

$$\lambda \cdot \left|Y^0[\ell] - Y^1[\ell]\right| \leq O\left(\left|X^0[\ell] - X^1[\ell]\right|\right)$$

Proof. The rationale behind the statement is that the leaf ℓ maximizing the ratio between $\left|Y^0[\ell] - Y^1[\ell]\right|$ and $\left|X^0[\ell] - X^1[\ell]\right|$ has $\Delta(\ell)$ maximized (e.g. ℓ with $3|\ell|/4$ positive literals). This is however technically challenging to prove directly. Thus, we will split the proof in two cases: first with $\Delta(\ell) \leq 16/10b_Y$ and then with $\Delta(\ell) \geq 16/10b_Y$. For the second regime, we will actually be able to prove that the most-separating leaves have $\Delta(\ell) = |\ell|/2$. Fix now any non-witnessing leaf ℓ, let $k := |\ell|$ and $q := \Delta(\ell)/2$. Since $\textsc{h} \notin \ell$, we have:

$$\left|Y^0[\ell] - Y^1[\ell]\right| = \left(1/4 - b_Y^2\right)^{k/2} \cdot \left(r_Y^q - r_Y^{-q}\right) \cdot (1 - p_*)^k \qquad \text{where} \quad r_Y := \frac{1 + 2b_Y}{1 - 2b_Y}$$

$$\left|X^0[\ell] - X^1[\ell]\right| = \left(1/4 - b_X^2\right)^{k/2} \cdot \left(r_X^q - r_X^{-q}\right) \qquad \text{where} \quad r_X := \frac{1 + 2b_X}{1 - 2b_X}$$

As $b_X \leq b_Y$, we will ignore the terms $\left(1/4 - b_Y^2\right)^{k/2}$ and $\left(1/4 - b_X^2\right)^{k/2}$. For the first regime $q \leq 16/10b_Y$, we don't even need the *dampening* term $(1 - p_{\textsc{h}})^k$ and we simply show that:

$$r_Y^q - r_Y^{-q} \leq \frac{10000}{\lambda} \cdot \left(r_X^q - r_X^{-q}\right) \tag{18}$$

Using the series representation of the exponential function and Lemma 44 to bound $\ln(r_Y)$, we have:

$$r_Y^q - r_Y^{-q} = 2 \sum_{t \geq 0 \text{ odd}} \frac{\ln(r_Y)^t q^t}{t!} \leq e^{\ln(r_Y)q} - 1 \leq e^{6b_Y q} - 1$$

Now, using the fact that $\lambda \leq 3b_X/b_Y$ (see Lemma 43), have:

$$\frac{10000}{\lambda} \cdot \left(r_X^q - r_X^{-q}\right) = \frac{20000}{\lambda} \sum_{t \geq 0 \text{ odd}} \frac{\ln(r_X)^t q^t}{t!} \geq \frac{20000}{\lambda} \cdot \ln(r_X) q \geq \frac{80000}{3} \cdot b_Y q$$

Hence, 18 holds if $b_Y q \leq 1.6$ so that the claim holds in the first regime. In the regime where $q \geq 16/10b_Y$, we will show that:

$$\lambda(1 - p_{\textsc{h}})^k \Phi(q) \leq 3 \quad \text{where} \quad \Phi(q) := \frac{r_Y^q}{r_X^q - r_X^{-q}} \tag{19}$$

Since Φ (as a real function over q) is increasing on the interval $[16/10b_Y, \infty)$ (see Lemma 40), the maximum of the left-hand side of equation 19 is attained at the boundary of the domain, i.e. for $q = k/4$. Therefore, in the second regime:

$$\lambda(1 - p_{\textsc{h}})^k \Phi(q) \leq \lambda(1 - p_{\textsc{h}})^k \Phi(k/4) = \frac{\lambda}{r_X^{k/4} - r_X^{-k/4}} \cdot \left[(1 - p_{\textsc{h}})^4 \cdot r_Y\right]^{k/4} \leq \frac{\lambda}{r_X^{k/4} - r_X^{-k/4}}$$

The last inequality holds because the quantity in the square bracket is ≤ 1 (recall that $p_{\textsc{h}} \in [3b_Y, 4b_Y]$). Now, note that $\mu - \mu^{-1} \geq \lambda/3$ if $\mu \geq 1 + \lambda/3$ but since $k \geq 1/b_Y$ and $\lambda \leq 3b_X/b_Y$, we have:

$$\mu := r_X^{k/4} \geq \left(\frac{1 + 2b_X}{1 - 2b_X}\right)^{k/4} \geq (1 + 4b_X)^{k/4} \geq 1 + kb_X \geq 1 + \lambda/3$$

So that 19 holds. □

A Appendix

A.1 Hypergeometric vs. multinomial

We let $H_3(N_a, N_b, N_c, k)$ be the hypergeometric distribution where one sample k objects without replacement where there are N_a, N_b and N_c objects of type a, b and c, respectively. This distribution is not independent and thus hard to work with. However, when k is small enough, the hyper-geometric distribution becomes very close to the multinomial distribution $M(N_a/N, N_b/N, N_c/N, k)$ where $N = N_a + N_b + N_c$.

Lemma 37. *If $q_a, q_b, q_c \in \mathbb{N}$ are such that $q_a + q_b + q_c = k$, $k \leq \sqrt{N}/2$, $q_a \leq N_a/2$, $q_b \leq N_b/2$ and $q_c \leq N_c/2$, then:*

$$\left(1 - \frac{2q_a^2}{N_a} - \frac{2q_b^2}{N_b} - \frac{2q_c^2}{N_c}\right) \cdot \Pr_M[q_a, q_b, q_c] \leq \Pr_{H_3}[q_a, q_b, q_c] \leq \left(1 + \frac{4k^2}{N}\right) \cdot \Pr_M[q_a, q_b, q_c]$$

Proof. Using the definition of the hyper-geometric distribution, we have:

$$\Pr_{H_3}[q_a, q_b, q_c] = \frac{\binom{N_a}{q_a}\binom{N_b}{q_b}\binom{N_c}{q_c}}{\binom{N}{q_a+q_b+q_c}} = \frac{k!}{q_a! q_b! q_c!} \cdot \prod_{i=0}^{q_a-1} \frac{N_a - i}{N - i} \cdot \prod_{i=0}^{q_b-1} \frac{N_b - i}{N - q_a - i} \cdot \prod_{i=0}^{q_c-1} \frac{N^1 - i}{N - q_a - q_b - i}$$

Let us denote by P the three products. We proceed by upper-bounding it:

$$P \leq \left(\frac{N_a}{N}\right)^{q_a} \cdot \left(\frac{N_b}{N - q_a}\right)^{q_b} \cdot \left(\frac{N_c}{N - q_a - q_b}\right)^{q_c} \leq \left(\frac{N_a}{N}\right)^{q_a} \cdot \left(\frac{N_b}{N}\right)^{q_b} \cdot \left(\frac{N_c}{N}\right)^{q_c} \cdot \left(\frac{1}{1 - k/N}\right)^{k}$$

Recall that $k \leq \sqrt{B}/2$, hence:

$$\left(\frac{1}{1 - k/N}\right)^{k} \leq \left(1 + \frac{2k}{N}\right)^{k} \leq e^{2k^2/N} \leq 1 + \frac{4k^2}{N}$$

The upper bound therefore follows. We give a lower bound to P as follows:

$$\begin{aligned} P &\geq \prod_{i=0}^{q_a-1} \frac{N_a - i}{N} \cdot \prod_{i=0}^{q_b-1} \frac{N_b - i}{N} \cdot \prod_{i=0}^{q_c-1} \frac{N_c - i}{N} \\ &= \left(\frac{N_a}{N}\right)^{q_a} \cdot \left(\frac{N_b}{N}\right)^{q_b} \cdot \left(\frac{N_c}{N}\right)^{q_c} \cdot \prod_{i=0}^{q_a-1} 1 - \frac{i}{N_a} \cdot \prod_{i=0}^{q_b-1} 1 - \frac{i}{N_c} \cdot \prod_{i=0}^{q_c-1} 1 - \frac{i}{N_c} \\ &\geq \left(\frac{N_a}{N}\right)^{q_a} \cdot \left(\frac{N_b}{N}\right)^{q_b} \cdot \left(\frac{N_c}{N}\right)^{q_c} \cdot \left(1 - \frac{q_a}{N_a}\right)^{q_a} \cdot \left(1 - \frac{q_b}{N_b}\right)^{q_b} \cdot \left(1 - \frac{q_c}{N_c}\right)^{q_c} \end{aligned}$$

Recalling that $q_a \leq N_a/2$, $q_b \leq N_b/2$ and $q_c \leq N_c/2$, we get the desired lower bound:

$$\begin{aligned} \left(1 - \frac{q_a}{N_a}\right)^{q_a} \cdot \left(1 - \frac{q_b}{N_b}\right)^{q_b} \cdot \left(1 - \frac{q_c}{N_c}\right)^{q_c} &\geq \exp\left(-\frac{2q_a^2}{N_a} - 2\frac{q_b^2}{N_b} - \frac{2q_c^2}{N_c}\right) \\ &\geq 1 - \frac{2q_a^2}{N_a} - \frac{2q_b^2}{N_b} - \frac{2q_c^2}{N_c} \end{aligned}$$ □

The same holds in the case were there are two classes of objects. More specifically, we let $H_2(N_a, N_b, k)$ be the hyper-geometric distributions which amounts to sampling without replacement from a population with N_a objects of type a and N_b objects of type b and define $B(N_a/N, N_b/N, k)$ to be the classical binomial distribution (with replacement). Again, if k is small enough then the distributions can be interchanged.

Lemma 38. *If $q_a, q_b \in \mathbb{N}$ are such that $q_a + q_b = k$, $k \leq \sqrt{N}/2$, $q_a \leq N_a/2$ and $q_b \leq N_b/2$, then:*

$$\left(1 - \frac{2q_a^2}{N_a} - \frac{2q_b^2}{N_b}\right) \cdot \Pr_M[q_a, q_b] \leq \Pr_{H_2}[q_a, q_b] \leq \left(1 + \frac{4k^2}{N}\right) \cdot \Pr_M[q_a, q_b]$$

Proof. Similar to the proof of Lemma 37. □

A.2 Properties of some functions

Lemma 39. *The function $r(q)$ in the proof of Lemma 32 has minimizer $q^\star \in [kb_Y/7, kb_Y]$.*

Proof. Setting $\partial r/\partial q$ equal to zero yields an equation for the minima:

$$(1-\lambda)\left(1-4b_X^2\right)^{k/2}\ln\left(\frac{1-2b_X}{1+2b_X}\right)\left(\frac{1-2b_X}{1+2b_X}\right)^q a+\lambda\left(1-4b_Y^2\right)^{k/2}\ln\left(\frac{1+2b_Y}{1-2b_Y}\right)\left(\frac{1+2b_Y}{1-2b_Y}\right)^q=0$$

Shuffling around:

$$(1-\lambda)\left(1-4b_X^2\right)^{k/2}\ln\left(\frac{1+2b_X}{1-2b_X}\right)\left(\frac{1-2b_X}{1+2b_X}\right)^q=\lambda\left(1-4b_Y^2\right)^{k/2}\ln\left(\frac{1+2b_Y}{1-2b_Y}\right)\left(\frac{1+2b_Y}{1-2b_Y}\right)^q$$

Shuffling around and using the precise definition of λ (see (15)):

$$\left(\frac{1+2b_X}{1-2b_X}\cdot\frac{1+2b_Y}{1-2b_Y}\right)^q=\frac{(1-\lambda)\left(1-4b_X^2\right)^{k/2}\ln\left(\frac{1+2b_X}{1-2b_X}\right)}{\lambda\left(1-4b_Y^2\right)^{k/2}\ln\left(\frac{1+2b_Y}{1-2b_Y}\right)}=\left(\frac{1-4b_X^2}{1-4b_Y^2}\right)^{k/2}$$

This allows to isolate $q^\star$ and using the bounds of Lemma 44, we have:

$$q^\star=\frac{k}{2}\cdot\frac{\ln(1-4b_X^2)-\ln(1-4b_Y^2)}{\ln\left(\frac{1+2b_X}{1-2b_X}\cdot\frac{1+2b_Y}{1-2b_Y}\right)}\implies\frac{k}{2}\cdot\frac{4b_Y^2-8b_X^2}{6b_X+6b_Y}\le q^\star\le\frac{k}{2}\cdot\frac{8b_Y^2-4b_X^2}{4b_X+4b_Y}$$

Finally, recall that $b_X \in o(b_Y) \in o(1)$ so that $q^\star \in [kb_Y/7, kb_Y]$ □

Lemma 40. *The function $\Phi(q)$ of Lemma 36 is increasing for $q \in [16/10b_Y, \infty)$.*

Proof. The derivative of Φ is:

$$\frac{\partial\Phi}{\partial q}=\frac{\ln(r_Y)r_Y^q\left(r_X^q-r_X^{-q}\right)-\ln(r_X)r_Y^q\left(r_X^q+r_X^{-q}\right)}{\left(r_X^q+r_X^{-q}\right)^2}$$

Thus, we only need to show that $\ln(r_Y)\left(r_X^q-r_X^{-q}\right)\ge\ln(r_X)\left(r_X^q+r_X^{-q}\right)$ for $q\in[16/10b_Y,\infty)$. Using the series representation of the exponential function and various bounds of Lemma 44, we have:

$$\begin{aligned}\ln(r_Y)\left(r_X^q-r_X^{-q}\right)&\ge\textstyle\sum_{t\ge0\text{ odd}}c(t)q^t\quad\text{where}\quad c(t):=8b_Y\ln(r_X)^t/t!\\\ln(r_X)\left(r_X^q+r_X^{-q}\right)&\le\textstyle\sum_{t\ge0\text{ even}}d(t)q^t\quad\text{where}\quad d(t):=12b_X\ln(r_X)^t/t!\end{aligned}$$

Observe that under the hypothesis that $q\ge16/10b_Y$ it holds that $c(1)q^1/2\ge d(0)q^0$. Thus, it only remains to show that for any $t\ge1$:

$$\underbrace{c(t-1)q^{t-1}/2}_{o_1}+\underbrace{c(t+1)q^{t+1}/2}_{o_2}\ge d(t)q^t$$

If $o_2\ge d(t)q^t$, the claim is already good. If not, then we have: $t\ge b_Y\ln(r_X)q/3b_X-1$ so that:

$$\frac{o_1}{d(t)q^t}=\frac{tb_Y}{3b_X\ln(r_X)q}\ge\left(\frac{b_Y\ln(r_X)q}{3b_X}-1\right)\cdot\frac{b_Y}{3b_X\ln(r_X)q}=\frac{b_Y^2}{9b_X^2}-\frac{b_Y}{3b_X\ln(r_X)q}\ge\frac{b_Y^2}{144b_X^2}$$

Where the last inequality is due to the fact that $\ln(r_X)\ge2b_X$ and $q\ge16/10b_Y$. Since $b_X<o(b_Y)$, we have $o_1\ge d(t)q^t$ and thus Φ is indeed increasing on $[16/10b_Y,\infty)$. □

A.3 Properties of trees

Lemma 41. *Let $P = (P^0 + P^1)/2$ be a distribution and D a decision tree together with some $L \in \mathbb{N}$. If D' is the version of D that stops after L queries, then:*

$$\frac{\operatorname{cost}(D', P)}{\operatorname{TV}(\operatorname{tran}(D', P^0), \operatorname{tran}(D', P^1))} \geq L \implies \frac{\operatorname{cost}(D, P)}{\operatorname{TV}(\operatorname{tran}(D, P^0), \operatorname{tran}(D, P^1))} \geq \frac{L}{3}$$

Proof. Define $\mathcal{L}^{\text{large}}(D) \coloneqq \{\ell \in \mathcal{L}(D) : |\ell| \geq L\}$ and let $P[\mathcal{L}^{\text{large}}(D)]$ be the probability that a leaf of $\mathcal{L}^{\text{large}}$ is reached by $x \sim P$ in D. Observe that:

$$P[\mathcal{L}^{\text{large}}(D)] \geq \operatorname{TV}(\operatorname{tran}(D, P^0), \operatorname{tran}(D, P^1))/3 \implies \operatorname{cost}(D, P) \geq L \cdot \operatorname{TV}(\operatorname{tran}(D, P^0), \operatorname{tran}(D, P^1))/3$$

Hence, we may assume for the remainder of the proof that $P[\mathcal{L}^{\text{large}}(D)] \leq \operatorname{TV}(\operatorname{tran}(t, P^0), \operatorname{tran}(t, P^1))/3$. If $P[\mathcal{L}^{\text{large}}(D)]$ is small, it must be that D' holds a constant fraction of the bias of D. Indeed, letting $\mathcal{L}^0(D) \coloneqq \{\ell \in \mathcal{L}(D) : P^0[\ell] \geq P^1[\ell]\}$:

$$\begin{aligned}\operatorname{TV}(\operatorname{tran}(D', P^0), \operatorname{tran}(D', P^1)) &= \textstyle\sum_{\ell \in \mathcal{L}^0(D')} P^0[\ell] - P^1[\ell] \\ &\geq \textstyle\sum_{\ell \in \mathcal{L}^0(D)} P^0[\ell] - P^1[\ell] - \sum_{\ell \in \mathcal{L}^{\text{large}}(D)} P^0[\ell] + P^1[\ell] \\ &= \operatorname{TV}(\operatorname{tran}(D, P^0), \operatorname{tran}(D, P^1)) - 2 \cdot P[\mathcal{L}^{\text{large}}(D)] \\ &= \operatorname{TV}(\operatorname{tran}(D, P^0), \operatorname{tran}(D, P^1))/3\end{aligned}$$

Finally, as D' is a truncated copy of D, it holds that $\operatorname{cost}(D, P) \geq \operatorname{cost}(D', P)$ and the claim follows. □

Lemma 42 (Acceptance centring)**.** *If D is a deterministic decision tree over $\{0,1\}^*$ labelled by $\{B^0, B^1\}$ with bias $\delta = \Pr_{x\sim B^0}[D(x) = B^0] - \Pr_{x\sim B^1}[D(x) = B^0]$, then there exists a randomised decision tree R with* $\operatorname{cost}(R, B) \leq \operatorname{cost}(D, B)$ *and* $\operatorname{depth}(R) \leq \operatorname{depth}(D)$ *such that:*

$$\Pr_{x\sim B^0}[R(x) = B^0] = \frac{1}{2} + \xi \quad \textit{and} \quad \Pr_{x\sim B^1}[R(x) = B^0] = \frac{1}{2} - \xi \quad \textit{where} \quad \xi \geq \delta/6$$

Proof. Let $p \coloneqq \Pr_{x\sim B^0}[t(x) = B^0]$ and suppose by symmetry that $p \leq 1/2$. For some $\alpha \in [0,1]$ that we fix later, we define R to query nothing and output B^0 with probability α and run D with remaining probability $1 - \alpha$. As such, $\Pr_{x\sim B^0}[R(x) = B^0] = \alpha + (1-\alpha) \cdot p$ and $\Pr_{x\sim B^1}[R(x) = B^0] = \alpha + (1-\alpha) \cdot (p - \delta)$. Thus, by setting $\alpha \coloneqq 1 - 1/(2 - 2p + \delta)$, we have $\xi = \delta/(4 - 4p + 2\delta) \geq \delta/6$ and the desired acceptance probabilities. □

A.4 Some inequalities

Lemma 43. *The mixture parameter λ defined in (15) satisfies $\lambda \in (b_X/b_Y) \cdot [2/3, 3]$.*

Proof. Immediate by recalling that $b_X, b_Y \in o(1)$ and using inequalities (20) and (22) of Lemma 44. □

Lemma 44. *For any $x \in [0, 0.5]$, $y \in [0,1]$ and $k \in [0, \infty]$,*

$$x \leq \frac{x}{1-x} \leq 2x \tag{20}$$

$$-2x \leq \ln(1-x) \leq -x \tag{21}$$

$$2x \leq \ln\left(\frac{1+x}{1-x}\right) \leq 3x \tag{22}$$

$$1 - (1-y)^k \leq ky \tag{23}$$

Proof. Inequality (20) holds by inspection while (23) is proven in Lemma 3 of [BB20a]. The upper bound of (21) is due to a truncation of the Taylor series whereas the lower bound comes from:

$$x - \ln(1-x) = \sum_{n=2}^{\infty} \frac{x^n}{n} \leq x \cdot \sum_{n=1}^{\infty} \frac{x^n}{n} = -x\ln(1-x) \leq -\ln\left(\frac{1}{2}\right)x \implies \ln(1-x) \geq -2x$$

The lower bound in (22) is again due to a truncation of the Taylor series while the upper bound is a combination of (21) and the identity $1+x \leq e^x$, i.e. $\ln((1+x)/(1-x)) = \ln(1+x) - \ln(1-x) \leq x + 2x$. □

Lemma 45. *If D is a decision tree and $\mathcal{U}^k = \{\ell \in \mathcal{L}(D) : |\ell| = k \text{ and } \Delta(\ell) \geq 2k^{2/3}\}$ for all $k \in \mathbb{N}$, then:*

$$\textstyle\sum_{\ell \in \mathcal{U}^k} Z[\ell] \leq 2e^{-k^{1/3}/48} \qquad \forall k \leq 1/64b_Y^3$$

Proof. Using the definition of Z, we can recast the sum as a probability:

$$\sum_{\ell \in \mathcal{U}^k} Z[\ell] \leq \sum_{\ell \in \mathcal{U}^k} 2 \cdot \max\left\{Z^0[\ell], Z^1[\ell]\right\} \leq 2 \sum_{q=k/2+k^{2/3}}^{k} \binom{k}{q} \left(\frac{1}{2} + b_Y\right)^q \left(\frac{1}{2} - b_Y\right)^{k-q}$$

Now, the last quantity can be interpreted as the probability of having at least $k/2 + k^{2/3}$ successes in k independent trials where the success probability is $1/2 + b_Y$. Therefore, we may leverage a standard Chernoff bound as follows:

$$\sum_{\ell \in \mathcal{U}^k} Z[\ell] \leq 2 \Pr\left[\sum_{i \in [k]} \text{Bernoulli}(1/2 + b_Y) \geq (1+\delta)\mu\right] \quad \text{where } \mu = \frac{k}{2} + kb_Y \text{ and } \delta = \frac{k/2 + k^{2/3}}{\mu} - 1$$

Note that under the hypothesis that $k \leq 1/64b_Y^3$, we have that $\delta \geq 0$ and $\delta^2\mu/3 \geq k^{1/3}/48$, thus:

$$\sum_{\ell \in \mathcal{U}^k} Z[\ell] \leq 2 \Pr\left[\sum_{i \in [k]} \text{Bernoulli}(1/2 + b_Y) \geq (1+\delta)\mu\right] \leq e^{-\delta^2\mu/3} \leq e^{-k^{1/3}/48}$$ □

Acknowledgements

We thank anonymous FOCS reviewers for their comments.

References

[ABK16] Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th Symposium on Theory of Computing (STOC)*, pages 863–876. ACM, 2016. doi:10.1145/2897518.2897644.

[AGJ+18] 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 Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 93, pages 10:1–10:13, 2018. doi:10.4230/LIPIcs.FSTTCS.2017.10.

[AKK16] Andris Ambainis, Martins Kokainis, and Robin Kothari. Nearly optimal separations between communication (or query) complexity and partitions. In *Proceedings of the 31st Computational Complexity Conference (CCC)*, pages 4:1–4:14. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.CCC.2016.4.

[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)*. IEEE, nov 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)*. IEEE, nov 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. Complexity and Logic. 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 24h 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.

[BK16] Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 55, pages 60:1–60:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.60.

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

[DM21] Yogesh Dahiya and Meena Mahajan. On (simple) decision tree rank. In *Proceedings of the 41st oundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 213, pages 15:1–15:16. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.15.

[GJPW18] Mika Göös, T. S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication versus partition number. *ACM Transactions on Computation Theory*, 10(1), 2018. doi:10.1145/3170711.

[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)*, volume 132, pages 64:1–64:13, 2019. doi:10.4230/LIPIcs.ICALP.2019.64.

[GM21] Mika Göös and Gilbert Maystre. A majority lemma for randomised query complexity. In *Proceedings of the 36th Computational Complexity Conference (CCC)*, volume 200, pages 18:1–18:15. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.CCC.2021.18.

[GSS16] Justin Gilmer, Michael Saks, and Srikanth Srinivasan. Composition limits and separating examples for some boolean function complexity measures. *Combinatorica*, 36(3):265–311, 2016. doi:10.1007/s00493-014-3189-x.

[GTW21] Uma Girish, Avishay Tal, and Kewen Wu. Fourier Growth of Parity Decision Trees. In *Proceedings of the 36th Computational Complexity Conference (CCC)*, volume 200, pages 39:1–39:36. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.CCC.2021.39.

[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.

[Li21] Yaqiao Li. Conflict complexity is lower bounded by block sensitivity. *Theoretical Computer Science*, 856:169–172, feb 2021. doi:10.1016/j.tcs.2020.12.038.

[LMR$^+$11] Troy Lee, Rajat Mittal, Ben Reichardt, Robert Spalek, 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.

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

[She12] Alexander A. Sherstov. The communication complexity of gap hamming distance. *Theory of Computing*, 8(8):197–208, 2012. URL: http://www.theoryofcomputing.org/articles/v008a008, doi:10.4086/toc.2012.v008a008.

[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, 2013. doi:10.1145/2422436.2422485.

[Ver98] Nikolai Vereshchagin. Randomized boolean decision trees: Several remarks. *Theoretical Computer Science*, 207(2):329–342, nov 1998. doi:10.1016/s0304-3975(98)00071-1.

[Yao77] Andrew Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, Oct 1977. doi:10.1109/SFCS.1977.24.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il