A New Minimax Theorem for Randomized Algorithms

Shalev Ben-David
University of Waterloo
shalev.b@uwaterloo.ca

Eric Blais
University of Waterloo
eric.blais@uwaterloo.ca

September 22, 2020

Abstract

The celebrated minimax principle of Yao (1977) says that for any Boolean-valued function f with finite domain, there is a distribution μ over the domain of f such that computing f to error ϵ against inputs from μ is just as hard as computing f to error ϵ on worst-case inputs. Notably, however, the distribution μ depends on the target error level ϵ: the hard distribution which is tight for bounded error might be trivial to solve to small bias, and the hard distribution which is tight for a small bias level might be far from tight for bounded error levels.

In this work, we introduce a new type of minimax theorem which can provide a hard distribution μ that works for all bias levels at once. We show that this works for randomized query complexity, randomized communication complexity, some randomized circuit models, quantum query and communication complexities, approximate polynomial degree, and approximate log-rank. We also prove an improved version of Impagliazzo's hardcore lemma.

Our proofs rely on two innovations over the classical approach of using Von Neumann's minimax theorem or linear programming duality. First, we use Sion's minimax theorem to prove a minimax theorem for ratios of bilinear functions representing the cost and score of algorithms.

Second, we introduce a new way to analyze low-bias randomized algorithms by viewing them as "forecasting algorithms" evaluated by a certain proper scoring rule. The expected score of the forecasting version of a randomized algorithm appears to be a more fine-grained way of analyzing the bias of the algorithm. We show that such expected scores have many elegant mathematical properties: for example, they can be amplified linearly instead of quadratically. We anticipate forecasting algorithms will find use in future work in which a fine-grained analysis of small-bias algorithms is required.

Contents

1 Introduction

Yao's minimax principle [Yao77] is a central tool in the analysis of randomized algorithms in many different models of computation. In its most commonly-used form, it states that for every Boolean-valued function f with a finite domain, if $\mathcal{R}^{(c)}$ denotes the set of randomized algorithms with worst-case cost at most c and Δ denotes the set of distributions over the domain of f, then

$$\min_{R \in \mathcal{R}^{(c)}} \max_{\mu \in \Delta} \Pr[R(x) \neq f(x)] = \max_{\mu \in \Delta} \min_{R \in \mathcal{R}^{(c)}} \Pr[R(x) \neq f(x)]$$

with both probabilities being over the choice of x drawn from μ and the internal randomness of R. This identity implies that there exists a distribution μ for which any algorithm that computes f with bounded error over inputs drawn from μ must have cost at least $\mathrm{R}(f)$, the cost of computing f to worst-case bounded error. But it does not say anything else about μ itself. Notably,

I. The minimax principle does not guarantee that the resulting distribution μ must be balanced on the sets $f^{-1}(0)$ and $f^{-1}(1)$.

II. More generally, it does not rule out the possibility that f is very easy to compute by randomized algorithms that are only required to output the correct value with probability at least $\frac{1+\gamma}{2}$ for some small bias measure $\gamma > 0$ over inputs drawn from the distribution μ.

A separate application of the minimax principle can be used to show that there is a distribution μ' for which all randomized algorithms computing f with bias γ over μ' have cost at least $\mathrm{R}_{\frac{1-\gamma}{2}}(f)$ (the cost of computing f to worst-case error $(1-\gamma)/2$), but then there is no guarantee that randomized algorithms with bounded error over μ' must have cost anywhere close to $\mathrm{R}(f)$.

Intuitively, it seems reasonable to expect that for every function f, there is a distribution μ for f that addresses issues I and II: a distribution that is balanced on $f^{-1}(0)$ and $f^{-1}(1)$, and which is at least slightly hard even to solve to a small bias level γ.

Question 1.1 (Informal)**.** *Is there a distribution μ which certifies the hardness of f for all bias levels $\gamma > 0$ at the same time?*

More formally, observe that the cost of computing f to worst-case bias γ cannot be smaller than $\gamma^2 \mathrm{R}(f)$. This is because randomized algorithms can be *amplified*: by repeating an algorithm $O(1/\gamma^2)$ times and outputting the majority vote of the runs, we can increase its bias from γ^2 to $\Omega(1)$. Therefore, a natural refinement of Question 1.1 is as follows.

Question 1.2 (Refinement of Question 1.1)**.** *Is there a distribution μ such that for all bias levels $\gamma > 0$, any algorithm computing f to bias γ against μ must have cost at least $\Omega(\gamma^2 \mathrm{R}(f))$?*

Question 1.2 is the primary focus of this work. We answer it affirmatively in a variety of computational models (we can handle most models in which amplification and Yao's minimax principle both apply). We note that the distribution satisfying the conditions of Question 1.2 is hard for bounded error in Yao's sense, since each algorithm solving f to bounded error against μ must have cost at least $\Omega(\mathrm{R}(f))$. In addition to this, such μ must also be perfectly balanced between 0- and 1-inputs of f (by considering the limit as $\gamma \to 0$), and must remain somewhat hard to solve even to small bias levels.

The study of Question 1.2 has led us to consider randomized forecasting algorithms which output *probabilistic confidence predictions* about the value of $f(x)$, instead of a Boolean guess for $f(x)$. When evaluated using a certain proper scoring rule, the best possible score of a forecasting

algorithm is intimately related to the best possible bias of a randomized algorithm; in fact, the score appears to be a more fine-grained way of measuring the bias. Scores of forecasting algorithms appear to be the "right" way of measuring the success of randomized algorithms, as such scores satisfy elegant mathematical properties. The following question, which we answer affirmatively, turns out to be a strengthening of Question 1.2.

Question 1.3. *Is there a distribution μ such that for all $\eta > 0$, any forecasting algorithm which achieves expected score at least η against μ must have cost at least $\Omega(\eta \, \mathrm{R}(f))$?*

1.1 Motivation from joint computation

The answers to Question 1.2 and Question 1.3 have a direct impact on the study of composition theorems and joint computation problems in randomized computational models: a natural approach for such problems involves first applying a minimax theorem and then establishing the required inequalities in the deterministic distributional setting. However, as observed by Shaltiel [Sha03] this approach runs into trouble if the hard distribution is easy to solve to small bias. Specifically, Shaltiel considered distributions μ which are hard to solve most of the time, but which give a completely trivial input with small probability γ. Then computing n independent copies from μ is a little easier than n times the cost of computing f, because on average, γn of the copies are trivial; the cost of computing n independent inputs from μ is at most $(1-\gamma)n$ times the cost of solving f.

Things get even worse when the inputs have a promised correlation, as can happen when proving composition theorems. For a concrete example, consider the partial function $\textsc{Trivial}_n$, which is defined on domain $\{0^n, 1^n\}$ and maps $0^n \to 0$ and $1^n \to 1$. Suppose we want to prove a composition lower bound with $\textsc{Trivial}_n$ on the outside: that is, we want to show that for every function f, computing $\textsc{Trivial}_n$ composed with n copies of f requires $\Omega(\mathrm{R}(f))$ cost. In other words, we want to lower bound the cost of an algorithm which outputs 0 when given n 0-inputs to f, outputs 1 when given n 1-inputs to f, and outputs arbitrarily when given some other type of input.

Now, if we try to lower bound this using the hard distribution from Yao's minimax principle, then the distribution might give a trivial input with small probability γ, as Shaltiel observed; but then so long as $n = \Omega(1/\gamma)$, one of the inputs to f will be trivial with high probability, and we can solve this "all-0s vs all-1s" problem simply by searching for the trivial copy – potentially much faster than the worst-case cost of computing a single copy of f!

The hard distributions we give in this work solve this issue by being hard for all bias levels. In our companion manuscript [BB20], we use one of the query versions of our minimax theorem (Theorem 4.6) to prove a new composition theorem for randomized query complexity.

1.2 Main tools

Minimax theorem for cost/score ratios. The first main result is a new minimax theorem for the ratio of the cost and score of randomized algorithms. A special case of the theorem with a simple formulation is as follows.

Theorem 1.4. *[Special case of Theorem 2.18] Let $\mathcal{R}$ be a set of randomized algorithms that can be expressed as a convex subset of a real topological vector space. Let S be a nonempty finite set, and let Δ be the set of all probability distributions over S, viewed as a subset of $\mathbb{R}^{|S|}$. Let $\mathrm{cost}\colon \mathcal{R} \times \Delta \to (0, \infty)$ and $\mathrm{score}\colon \mathcal{R} \times \Delta \to [-\infty, \infty)$ be continuous bilinear functions. Then using the convention $r/0 = \infty$ for all $r \in (0, \infty)$ and the notation $r^+ = \max\{r, 0\}$ for all $r \in [-\infty, \infty]$, we have*

$$\inf_{R \in \mathcal{R}} \max_{x \in S} \frac{\mathrm{cost}(R, x)}{\mathrm{score}(R, x)^+} = \max_{\mu \in \Delta} \inf_{R \in \mathcal{R}} \frac{\mathrm{cost}(R, \mu)}{\mathrm{score}(R, \mu)^+}.$$

Further, all of the above maximums are attained.

The general version of the minimax theorem in Theorem 2.18 shows that the same identity holds even when the cost and score functions are semicontinuous and saddle (but not necessarily linear) under some mild additional restrictions. Furthermore, a variant of the theorem also holds when we consider convex and compact subsets of distributions over the finite set S instead of the set Δ of all distributions over that set.

Minimax theorems for ratios of semicontinuous and saddle functions as in Theorem 2.18 do not seem to have appeared in the literature previously in the precise form we need, but as we show in Section 2, they can be obtained by extending Sion's minimax theorem [Sio58] with standard arguments. We believe that the main contribution of Theorem 2.18 is in its interpretation for randomized algorithms. Various extensions and variations of Yao's minimax theorem have been considered in the computer science literature previously [Yao77; Imp95; Ver98; Bra15; BGK+18; BB19], but all of them appear to consider the cost of an algorithm (with the minimax theorem applied to algorithms with a fixed worst-case score), the score of an algorithm (with the cost being fixed), or a linear combination of the two. None of those variants suffice to answer the questions raised at the beginning of the introduction or to establish the results in the following subsections; what was needed in those cases was a minimax theorem for the *ratio* of the cost/score of randomized algorithms, and we suspect that this ratio minimax theorem will find further applications in computer science in the future as well.

Forecasting algorithms and linear amplification. To convert the statements obtained from Theorem 2.18 regarding the cost/score ratios of randomized algorithms under some distribution μ into more familiar lower bounds on the cost of randomized algorithms that achieve some bias on μ, we need a *linear amplification* theorem. Ideally, we would like to argue that if there exists a randomized algorithm R with bias γ on μ, then by combining $O(1/\gamma)$ instances of R we can obtain a randomized algorithm R' with $\mathrm{cost}(R',\mu) = O\left(\frac{1}{\gamma}\cdot \mathrm{cost}(R,\mu)\right) = O\left(\frac{\mathrm{cost}(R,\mu)}{\mathrm{bias}_f(R,\mu)}\right)$ and constant bias. Unfortunately, such a linear amplification property does not hold for most models of randomized algorithms, where amplification from bias γ to bounded error requires combining $O(1/\gamma^2)$ instances of the original algorithm. To obtain a linear amplification result, we must turn our attention away from bias and error and consider other score functions instead.[1]

To describe our score function, we first generalize our computational model from randomized algorithms that output 0 or 1 to *forecasting algorithms*, which are randomized algorithms that output a *confidence value* in $[0,1]$ for the value $f(x)$ of the function f on their given input x. A "low" confidence prediction is a value close to $\frac{1}{2}$ whereas a "high" confidence prediction would be a value very close to 0 or to 1. There are many natural ways to assign a score to a confidence value for $f(x)$. The study of such scoring rules and their properties has a rich history in the statistics and decision theory communities (see for instance [BSS05; GR07] and references therein); we discuss some fundamental scoring rules and give relations between them in Section 3. Of particular importance to our main purpose is the scoring rule $\mathrm{hs}\colon [0,1] \to [-\infty, 1]$ defined by

$$\mathrm{hs}_f(p) = \begin{cases} 1 - \sqrt{\frac{1-p}{p}} & \text{when } f(x) = 1 \\ 1 - \sqrt{\frac{p}{1-p}} & \text{when } f(x) = 0. \end{cases}$$

[1] The astute reader may have noticed that we obtain linear amplification if we simply set the score to be the squared bias of the randomized algorithm. That is true, but this approach does not work in conjunction with the ratio minimax theorem since this score function no longer satisfies the appropriate saddle property requirements of that theorem; this is why we instead consider forecasting algorithms as described below.

Define the score of a forecasting algorithm R on an input x in the domain of f to be $\mathrm{score}_{\mathrm{hs},f}(R,x) = \mathbb{E}[\mathrm{hs}_f(R(x))]$, the expectation of the hs score of the output of R over the internal randomness of R. Then linear amplification does hold for this score function.

Lemma 1.5. *For any Boolean-valued function f, any forecasting algorithm R, and any $k \geq 1$, there is a forecasting algorithm R' that combines the outputs of k instances of R and satisfies*

$$\mathrm{score}_{\mathrm{hs},f}(R',x) \geq 1 - (1 - \mathrm{score}_{\mathrm{hs},f}(R,x))^k$$

for every x in the domain of f. In particular, when $k = \max_x \frac{2}{\mathrm{score}_{\mathrm{hs},f}(R,x)}$ then for each $x \in \mathrm{Dom}(f)$, $\mathrm{score}_{\mathrm{hs},f}(R',x) \geq 1 - e^{-2} > 0.85$.

To the best of our knowledge, Lemma 1.5 has not previously appeared in the literature. This lemma is sensitive to the precise definition of hs_f; other scoring rules do not appear to satisfy this amplification property, which is crucial for the proof of our main results. Additionally, the scoring rule hs_f is special because there is a close connection between hs score of forecasting algorithms and the bias of randomized algorithms.

Lemma 1.6. *For any Boolean-valued function f, any distribution μ on $\mathrm{Dom}(f)$, and any parameter $\gamma > 0$,*

- *If there exists a randomized algorithm R with $\mathrm{bias}_f(R,\mu) = 1 - 2\Pr[R(x) \neq f(x)] \geq \gamma$, then there is a forecasting algorithm R' with $\mathrm{score}_{\mathrm{hs},f}(R',\mu) \geq 1 - \sqrt{1-\gamma^2} \geq \gamma^2/2$, and*
- *If there exists a forecasting algorithm R with $\mathrm{score}_{\mathrm{hs},f}(R,\mu) \geq \gamma$ then there is a randomized algorithm R' with $\mathrm{bias}_f(R',\mu) \geq \gamma$.*

Moreover, in both cases R' can be explicitly constructed from R by modifying its output.

Lemma 1.5 and Lemma 1.6 can be used to reprove the fact that $O(1/\gamma^2)$ instances of a bias-γ randomized algorithms can be combined to obtain a bounded-error algorithm; combining those lemmas (or, more precisely, specific instantiations of these lemmas that account for the explicit constructions of the relevant algorithms and their costs) with the minimax theorem also leads to new results as described in the next section.

1.3 Main results

Hard distributions for bounded error and small bias. The minimax theorem for cost/score ratios and linear amplification of forecasting algorithms can be combined to show that for many measures of randomized complexity, for every Boolean-valued function f with finite domain there exists a single distribution μ on which it is hard to compute f with bounded error *and* with (any) small bias. For example, letting $\mathrm{RDT}(f)$ denote the minimum (worst-case) query complexity of a randomized algorithm computing f (or equivalently the minimum worst-case depth of a decision tree computing f) with error at most $\frac{1}{3}$ on every input in $\mathrm{Dom}(f)$ and $\mathrm{RDT}^{\mu}_{\dot{\gamma}}$ denote the minimum query complexity of a randomized algorithm that has error probability at most $\dot{\gamma} := \frac{1-\gamma}{2}$ when inputs are drawn from μ, we obtain the following result.

Theorem 1.7. *For any non-constant partial function $f\colon \{0,1\}^n \to \{0,1\}$, there exists a distribution μ on $\mathrm{Dom}(f)$ such that for every $\gamma \in [0,1]$,*

$$\mathrm{RDT}^{\mu}_{\dot{\gamma}}(f) = \Omega\big(\gamma^2\,\mathrm{RDT}(f)\big).$$

We establish analogous theorems for multiple other computational models as well:

Randomized communication complexity	$\mathrm{RCC}^{\mu}_{\dot{\gamma}}(f) = \Omega\big(\gamma^2\,\mathrm{RCC}(f)\big)$	Corollary 4.8
Quantum query complexity	$\mathrm{QDT}^{\mu}_{\dot{\gamma}}(f) = \gamma \cdot \tilde{\Omega}\big(\mathrm{QDT}(f)\big)$	Theorem 5.2
Quantum communication complexity	$\mathrm{QCC}^{\mu}_{\dot{\gamma}}(f) = \gamma \cdot \tilde{\Omega}\big(\mathrm{QCC}(f)\big)$	Theorem 5.9
Polynomial degree	$\deg^{\mu}_{\dot{\gamma}}(f) = \gamma \cdot \tilde{\Omega}(\mathrm{adeg}(f))$	Theorem 6.5
Log-rank complexity	$\log \mathrm{rank}^{\mu}_{\dot{\gamma}}(f) = \gamma \cdot \tilde{\Omega}(\log \mathrm{rank}_{1/3}(f))$	Theorem 6.8
Circuit complexity	$\mathrm{Rcirc}^{\mu}_{\dot{\gamma}}(f) = \gamma^2 \cdot \tilde{\Omega}\big(\mathrm{Rcirc}(f)\big)$	Theorem 7.8
Log-depth circuit complexity	$\mathrm{RNC1}^{\mu}_{\dot{\gamma}}(f) = \gamma^2 \cdot \tilde{\Omega}\big(\mathrm{RNC1}(f)\big)$	Theorem 7.9
Threshold circuit complexity	$\mathrm{RTC0}^{\mu}_{\dot{\gamma}}(f) = \gamma^2 \cdot \tilde{\Omega}\big(\mathrm{RTC0}(f)\big)$	Theorem 7.10

(Note that as in Theorem 1.7, the novel aspect of all these results is that they guarantee that for each of the stated inequalities, there exists a *single* distribution μ that satisfies the inequality for *every* value of γ simultaneously.)

Hard distributions for forecasting algorithms. The theorems listed above settle Question 1.2 in the affirmative for the specified models. For the models with quadratic dependence on γ (i.e. randomized query complexity, randomized communication complexity, and the various randomized circuit models), we also get hard distributions which lower bound the expected score of a forecasting algorithm, settling Question 1.3 affirmatively.

Distinguishing power of randomized algorithms and protocols. In the communication complexity setting, we can also analyze how well a randomized communication protocol computes a function $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ via its communication transcripts. Let $\mathrm{tran}(R, \mu_0)$ denote the distribution on communication transcripts of the randomized protocol R on inputs drawn from μ. Then one way to measure how well R is able to distinguish 0- and 1-inputs of f is to measure the Hellinger distance between the distributions $\mathrm{tran}(R, \mu_0)$ and $\mathrm{tran}(R, \mu_1)$ of transcripts of R on some distributions μ_0 over $f^{-1}(0)$ and μ_1 over $f^{-1}(1)$. We can use the minimax and linear amplification theorems to give a strong upper bound on this Hellinger distance as a measure of the cost of the protocol.

Theorem 1.8. *For any non-constant partial function $f\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ over finite sets $\mathcal{X}$ and $\mathcal{Y}$, there is a pair of distributions μ_0 on $f^{-1}(0)$ and μ_1 on $f^{-1}(1)$ such that for any randomized communication protocol R, the squared Hellinger distance between the distribution of its transcripts on μ_0 and μ_1 is bounded above by*

$$\mathrm{h}^2\left(\mathrm{tran}(R,\mu_0), \mathrm{tran}(R,\mu_1)\right) = O\left(\frac{\min\{\mathrm{cost}(R,\mu_0), \mathrm{cost}(R,\mu_1)\}}{\mathrm{RCC}(f)}\right).$$

Here $\mathrm{cost}(R,\mu)$ denotes the expected amount of communication the protocol R transmits when given inputs from μ.

Theorem 4.6 establishes an analogous result for query complexity. In our companion paper [BB20], that theorem is one of the ingredients that enables us to establish a new composition theory for query complexity.

Hardcore lemma. Impagliazzo's Hardcore Lemma [Imp95] states that for every $\epsilon, \delta > 0$, if every circuit C of size at most s computes f with error at least δ on the uniform distribution, then there is a δ-regular distribution $\mu = \mu(\delta, \epsilon)$ for which every circuit that computes f with bias at least ϵ on the distribution μ must have size $\Omega(\epsilon^2 s)$. Informally, the lemma shows that if a function f is mildly hard on average, it is because it is "very" hard to compute on a fairly large subset of its inputs. But, interestingly, this version of the hardcore lemma leaves open the possibility that the hard core might be different for various levels ϵ of hardness. Using our main theorems, we can show that this is not the case.

Theorem 1.9. *There exists a universal constant $c > 0$ such that for any $\delta > 0$ and function $f : \{0,1\}^n \to \{0,1\}$, if every circuit C of size at most s satisfies $\Pr[C(x) = f(x)] \leq 1 - \delta$ when the probability is taken over the uniform distribution of x in $\{0,1\}^n$, then there is a distribution μ with min-entropy δ such that for every $\epsilon > 0$, any circuit C' of size at most $c \cdot \epsilon^2 / \log(1/\delta) \cdot s$ has success probability bounded by*

$$\Pr[C'(x) = f(x)] \leq \frac{1+\epsilon}{2}.$$

The proof of Theorem 1.9 follows closely the original argument of Nisan in [Imp95] that established the hardcore lemma via a minimax theorem. Since that original work, many extensions and different proofs of the hardcore lemma have been established (e.g., [Imp95; KS03; BHK09; TTV09]), but to the best of our knowledge Theorem 1.9 represents the first version of the lemma which gives a single distribution μ which is hard for all values of $\epsilon > 0$ simultaneously.

1.4 Recent independent work

In independent work concurrent with this one, Bassilakis, Drucker, Göös, Hu, Ma, and Tan [BDG+20] showed the existence of a certain hard distribution for randomized query complexity. They showed every Boolean function f has hard distributions μ_0 and μ_1 (on 0- and 1-inputs respectively) such that given query access to k independent samples from μ_b, it is still necessary to use $\Omega(\mathrm{R}(f))$ queries to the bits of the samples in order to decide the value of $b \in \{0,1\}$ to bounded error.

The guarantee on the hard distribution provided by [BDG+20] is formally stronger than the one we provide in Theorem 4.6 (though in our companion manuscript [BB20], we prove a new composition theorem for randomized query complexity, and use it to conclude that the guarantee of [BDG+20] turns out to be equivalent to the guarantee of Theorem 4.6 in our current work). The tools used by [BDG+20] are also completely different: they use arguments specific to query complexity that construct the hard distribution more explicitly, but their arguments do not generalize to other models such as communication complexity or circuit complexity.

1.5 Organization and overview of the remaining sections

Section 2 is devoted to proving the main minimax theorem for the cost/score ratio of randomized algorithms. The main result of that section is Theorem 2.18; the rest of the section is devoted to introducing the mathematical notions and preliminaries required to obtain a proof of that theorem from Sion's minimax theorem.

Section 3 introduces the basic definitions and some basic scoring rules for forecasting algorithms. The section establishes some of the core properties of scoring functions, including notably connections between the best score achievable by forecasting algorithms on distributions over inputs and various distance measures on those distributions. The final portions of this section then establish the main linear amplification theorem in general form in Lemma 1.5 and the general form of the conversion between randomized and forecasting algorithms in Lemma 3.15.

Section 4 focuses on the query and communication complexity settings. Conversions between randomized and forecasting algorithms in the query complexity setting are straightforward, but there is one significant challenge in applying the linear amplification theorem to obtain the results in Theorem 1.7 and Theorem 4.6: the cost and score of a randomized algorithm R on an input x can both depend on x itself. This is a problem because to obtain a constant score (and after the final conversion, a bounded-error randomized algorithm), we want to amplify R with a number k of copies that depends on the score of R on x—but since we don't know x we don't know what $\text{score}(R, x)$ is either. We get around this problem with odometer arguments: by empirically estimating the expected number of queries R makes on x, we can obtain effective bounds on the number k of copies of R that we need to obtain successful amplification.

As we show in the section, the communication complexity results Corollary 4.8 and Theorem 1.8 follow immediately from their query complexity analogues.

Section 5 establishes the results in the quantum query and communication complexity settings. Unlike in the classical setting, amplification that is linear in the bias of an algorithm *does* hold in the quantum query complexity setting. However, the proof of Theorem 5.2 requires that the set of algorithms must be representable as a convex subset of a real topological space, and that the cost of an algorithm is a convex function on this set. It is not immediately clear how quantum query algorithms can satisfy this condition, because in the usual definition, the cost of a mixture of two quantum algorithms would be the *maximum* of the costs of the algorithms rather than the average. To overcome this issue, we instead establish Theorem 5.2 via consideration of what we call *probabilistic* quantum algorithms, which correspond to probability distributions over quantum algorithms and do easily satisfy the appropriate convexity requirements. Probabilistic quantum algorithms are harder to amplify than regular quantum algorithms (due to their lack of coherence), but we show that a linear amplification theorem still holds.

Another important difference between the quantum and the classical setting is that the communication complexity result, Theorem 5.9, is not implied by the analogous query complexity result. Nonetheless, the same argument used for quantum query algorithms also holds for quantum communication protocols as well. We complete the proof of Theorem 5.9 by first providing an abstraction of the query complexity argument in Theorem 5.8 and then showing how communication protocols satisfy the conditions of this abstract theorem.

Section 6 considers the approximate polynomial degree and the logrank complexity of functions. As with quantum query complexity, approximate polynomial degree satisfies an amplification theorem that is linear in the bias, meaning that we do not need to use forecasting algorithms or scoring rules. However, also as with quantum query complexity, polynomials and their cost do not satisfy the right convexity requirements, as the degree of a mixture of two polynomials is not the average of their degrees. We overcome this by considering probabilistic polynomials. Proving an amplification theorem for probabilistic polynomials turns out to be somewhat tricky, and requires tools from approximation theory such as Jackson's theorem.

Approximate logrank inherits all of the problems of approximate polynomial degree, and adds a few more. To handle approximate logrank, we switch over to the nearly-equivalent model of the logarithm of the approximate gamma 2 norm, and then use the previous trick of considering the *probabilistic* approximate gamma 2 norm. To prove an amplification theorem for probabilistic gamma 2 norm we apply the same tools as for probabilistic polynomials.

Section 7 establishes the circuit complexity results. There are two main hurdles in establishing Theorem 7.8. The first is that the notion of randomized circuits is not as trivially extendable to forecasting circuits as in other computational models. We show that this conversion can be done efficiently when we discretize the set of confidence values that can be returned by forecasting circuits, and that this discretization does not affect the guaranteed relations between score and bias. The second is that the overhead required to combine the output of multiple instances of a randomized circuit during linear amplification is not trivial. This second hurdle can be overcome with the use of efficient circuit constructions for elementary arithmetic operations and the iterated addition problem.

The proof of the universal hardcore lemma in Theorem 1.9 is obtained via a slight generalization of the ratio minimax theorem. This variant of the minimax theorem is stated in Lemma 7.12 and the rest of the proof of Theorem 1.9 is presented in Section 7.3.

1.6 Further remarks and open problems

We make a few remarks regarding other possible generalizations of Yao's original minimax theorem. First, one may wonder why we provide a hard distribution μ satisfying $\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \Omega(\gamma^2\,\mathrm{R}(f))$ for all γ, rather than the stronger statement $\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \Omega(\mathrm{R}_{\dot{\gamma}}(f))$ for all γ. In other words, we've stated our lower bounds in terms of the bounded-error randomized cost $\mathrm{R}(f)$, which required amplification; why not directly compare the average-case complexity to bias γ, denoted $\mathrm{R}^{\mu}_{\dot{\gamma}}(f)$, to the worst-case complexity to bias γ, denoted $\mathrm{R}_{\dot{\gamma}}(f)$?

The reason is that this stronger version of the minimax is actually false: that is, there need not be a distribution μ for which $\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \Omega(\mathrm{R}_{\dot{\gamma}}(f))$ for all γ (even though for every given γ, such a distribution μ that depends on γ does exist, by Yao's minimax theorem). For a counterexample, consider the query complexity model. Let f be the Boolean function on $n+m+1$ bits, where if the first bit is 0 the function f evaluates to the parity of the next m bits, whereas if the first bit is 1 the function f evaluates to the majority of the last n bits. Say we take $n = m^2$. Then, since parity is hard to compute even to small bias, we have $\mathrm{R}_{\dot{\gamma}}(f) \geq m$ for all γ. We also have $\mathrm{R}_{1/3}(f) = \Omega(m^2)$, since majority on m^2 bits requires $\Omega(m^2)$ queries. Now, consider any distribution μ over the domain of f. If μ places nonzero probability mass on inputs with first bit 1, then μ can necessarily be solved to some sufficiently small bias using at most 2 queries (one query to the first bit of the input, and one to a random position in the input to majority). In this case, we would have $\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = O(1)$ and $\mathrm{R}_{\dot{\gamma}}(f) = \Omega(\sqrt{n})$ for this sufficiently small γ. Alternatively, if μ places zero probability mass on inputs with first bit 1, then solving f against μ is solving parity on $m = O(\sqrt{n})$ bits; hence $\mathrm{R}^{\mu}_{1/3}(f) = O(\sqrt{n})$, even though $\mathrm{R}_{1/3}(f) = \Omega(n)$. Similar counterexamples can be constructed in other computational models.

Another possible generalization of Yao's minimax is to a distribution μ for which $\mathrm{R}^{\mu}(f)$ is large even when the both the error of the algorithm and the expected cost are measured against μ. That is, in a normal application of Yao's minimax, we either consider randomized algorithms which only ever make at most T queries (against any input) and measure their expected error against μ, or else we consider randomized algorithms which only ever make error at most ϵ (against any input) and measure their expected cost against μ. One may wonder if it is possible for one distribution to certify the hardness of f in both ways at once, with both the cost and the error measured in expectation against μ.

The answer turns out to be yes, as first observed by Vereshchagin for query complexity [Ver98]. Vereshchagin stated his theorem for bounded error, but in the case of small bias γ, his techniques appear to give a distribution μ (which depends on γ) such that $\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \Omega(\gamma\,\mathrm{R}_{\dot{\gamma}}(f))$ even where the

left-hand side is defined as the *expected* query complexity against μ to bias at least γ (also against μ). This is in contrast to Yao-style minimax theorems, which are stronger in that they lack the γ factor on the right hand side, but weaker in that the left-hand side has either the cost or the error being worst-case (rather than both being average-case against μ).

Our results in this work are "Vereshchagin-like" in that they hold even when $\mathrm{R}^{\mu}_{\dot{\gamma}}(f)$ has both the cost and the bias defined in expectation against μ. We prove such results for randomized query complexity and randomized communication complexity, showing a single μ satisfies $\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \Omega(\gamma^2 \mathrm{R}(f))$ for all $\gamma > 0$, even when both the error and the cost in the definition of $\mathrm{R}^{\mu}_{\dot{\gamma}}(f)$ are average-case against μ. (For models such as quantum query complexity or circuit complexity, the expected cost of an algorithm does not have an obvious interpretation, since the algorithms generally have the same cost for all inputs; therefore, for those models we do not give a theorem in which the cost is measured in expectation against μ.)

Note that our minimax theorem is not directly comparable to Vereshagin, because we state our lower bounds in an "amplified" form – that is, the lower bounds are with respect to $\mathrm{R}(f)$ rather than $\mathrm{R}_{\dot{\gamma}}(f)$. As previously mentioned, this is necessary when proving that a single distribution works for all γ, and our theorems appear to be tight in that setting. Moreover, Vereshchagin's theorem is tight in its setting: the factor of γ is necessary, because average-case query complexity can be smaller than worst-case query complexity (for example, consider the parity function on n bits, which has $\mathrm{R}_{\dot{\gamma}}(f) = n$ for all γ; if we design a randomized algorithm which queries all the bits with probability γ and queries no bits with probability $1-\gamma$, it will use only γn expected queries, and it will solve f to bias γ).[2]

A remaining open problem is as follows: can Vereshchagin's theorem be modified to show

$$\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \Omega(\overline{\mathrm{R}}_{\dot{\gamma}}(f)), \tag{1}$$

where both cost and bias on the left are measured in expectation against μ, and where $\overline{\mathrm{R}}_{\dot{\gamma}}(f)$ denotes the worst-case (over the inputs of f) expected (over the internal randomness of the algorithm) query complexity of f to bias γ? Note that in the bounded-error setting, $\overline{\mathrm{R}}(f) = \Theta(\mathrm{R}(f))$, so for bounded γ this result follows from both Vereshchagin's theorem and from our work here. For small γ, we leave this question as an intriguing open problem.

We also note that we cannot hope that a single distribution μ satisfies (1) for all γ, because one can construct a counterexample via a modification of our earlier function: we let f be defined on $1+m+n$ bits, where if $x_1 = 0$ the function evaluates to the parity of the next m bits, and if $x_1 = 1$ the function evaluates to the majority of the last n bits, as before; this time we will have $n = m^{4/3}$. We also add a promise: we require that the input always has Hamming weight either at most $n/2 - \sqrt{n}$ or at least $n/2 + \sqrt{n}$ on the last n bits, turning the majority part of the function into a $\sqrt{n}$-gap majority function. Now, to compute f to worst-case bias γ requires at least γm expected queries on inputs x with $x_1 = 0$, and requires at least $\gamma^2 n$ expected queries on inputs with $x_1 = 1$, so at least $\Omega(\max\{\gamma m, \gamma^2 n\})$ expected queries in the worst case. This is $\Omega(n^{1/4})$ when $\gamma = n^{-1/2}$ and $\Omega(n)$ when γ is constant. Now fix a distribution μ, let p be the probability that μ assigns to inputs with $x_1 = 1$. If $p \leq 1/2$, then we can compute f to constant bias simply by querying the first bit, guessing randomly if $x_1 = 1$, and querying m bits to compute f exactly when $x_1 = 0$; this uses $O(n^{3/4})$ queries to achieve constant bias, instead of the $\Omega(n)$ which were required in the worst case. On the other hand, if $p \geq 1/2$, then we can compute f against μ by querying the first bit and nothing else when $x_1 = 0$ (guessing the answer randomly), and otherwise making one additional query to estimate the gap majority function to bias $1/\sqrt{n}$. This uses 2 queries and achieves bias $n^{-1/2}$ against μ, instead of the $\Omega(n^{1/4})$ queries required in the worst case.

[2] We thank an anonymous reviewer for this example.

2 Minimax theorem for the ratio of saddle functions

Minimax theorems take the form

$$\inf_{x\in X}\sup_{y\in Y}\alpha(x,y) = \sup_{y\in Y}\inf_{x\in X}\alpha(x,y).$$

For any function α, the left-hand side above is always at least the right hand side, but equality only holds under certain conditions; when equality does hold, we call it a minimax theorem.

Broadly speaking, the following conditions are required to ensure that a minimax theorem holds. First, X and Y must be convex sets (and they must be subsets of some real vector spaces). Second, α must be *saddle* – or at least quasisaddle – meaning that it is convex as a function of x and concave as a function of y (or at least quasiconvex and quasiconcave). Third, α must satisfy some continuity conditions. And finally, one of X or Y must be compact (importantly, it's not necessary for both to be compact).

In this section, we show that under certain conditions, minimax theorems also hold for *ratios* of positive saddle functions. Such a ratio of saddle functions is not necessarily saddle, but the important insight is that it is still quasisaddle.

2.1 Background definitions

In order to formally state the conditions in which minimax theorems hold, we will need a few definitions. We assume the reader is familiar with vector spaces and topological spaces, including standard terminology such as compact sets and neighborhoods.

Definition 2.1 (Real topological vector space)**.** *A* real topological vector space *is a tuple $(V,+,\cdot,\tau)$, where V is a set, $+$ is a function $V\times V\to V$, $\cdot$ is a function $V\times\mathbb{R}\to V$, and $\tau\subseteq 2^V$, such that*

- *$(V,+,\cdot)$ is a vector space over $\mathbb{R}$,*
- *(V,τ) is a topological space,*
- *$+$ is continuous under the topology τ, and*
- *$\cdot$ is continuous under the topology τ and the standard topology of $\mathbb{R}$.*

We note that any normed real vector space is a real topological space, as the norm induces a topology. We will primarily focus on the real topological vector spaces $\mathbb{R}^n$ for $n\in\mathbb{N}$, which have a standard topology.

Definition 2.2 (Extended reals)**.** *The* extended reals *is the set $\overline{\mathbb{R}} \coloneqq \mathbb{R}\cup\{-\infty,\infty\}$. We use the extended interval notation $(r,\infty] \coloneqq (r,\infty)\cup\{\infty\}$ for $r\in\mathbb{R}$, and similarly for $[-\infty,r)$ and $[-\infty,\infty]$. We associate with $\overline{\mathbb{R}}$ the following topology. A set $S\subseteq\overline{\mathbb{R}}$ is a neighborhood of $x\in\mathbb{R}$ if it contains an open interval $(x-\epsilon,x+\epsilon)$ for some $\epsilon\in(0,\infty)$, it is a neighborhood of ∞ if it contains the interval $(r,\infty]$ for some $r\in\mathbb{R}$, and it is a neighborhood of $-\infty$ if it contains the interval $[-\infty,r)$ for some $r\in\mathbb{R}$.*

We define addition, subtraction, multiplication, and division of extended reals in the intuitive way, with $\infty-\infty$, $0\cdot\infty$, ∞/∞, and $x/0$ for $x\in\overline{\mathbb{R}}$ all undefined. Note also that the extended reals are ordered (for each $x,y\in\overline{\mathbb{R}}$, we have either $x=y$, $x<y$, or $x>y$).

Note that while we define the extended reals and will often talk about extended-real-valued functions, our vector spaces will always be over the reals, not over the extended reals. In particular, the extended reals are not a field.

Definition 2.3 (Convexity of sets)**.** *We say a subset X of a real vector space V is* convex *if*

$$\forall x, y \in X,\ \forall \lambda \in (0,1) \quad \lambda x + (1-\lambda)y \in X.$$

Definition 2.4 (Convex hull)**.** *Let V be a real vector space and let $X \subseteq V$. The* convex hull *of X, denoted* $\mathrm{Conv}(X)$*, is the intersection of all convex subsets of V that contain X as a subset.*

Note that it is easy to verify that an arbitrary intersection of convex sets is convex, which means that the convex hull of any set is always convex.

Definition 2.5 ((quasi)convexity and (quasi)concavity of functions)**.** *Let V be a real vector space, let $X \subseteq V$ be convex, and let $\phi : X \to \overline{\mathbb{R}}$. We say that ϕ is* convex *if for all $x, y \in X$ and $\lambda \in (0,1)$, we have $\phi(\lambda x + (1-\lambda)y) \leq \lambda\phi(x) + (1-\lambda)\phi(y)$. We say ϕ is* quasiconvex *if for all $x, y \in X$ and $\lambda \in (0,1)$, we have $\phi(\lambda x + (1-\lambda)y) \leq \max\{\phi(x), \phi(y)\}$. We say that ϕ is* concave *if $-\phi$ is convex, and we say ϕ is* quasiconcave *if $-\phi$ is quasiconvex. If ϕ is both convex and concave, we say it is* linear*.*

Note that if ∞ and $-\infty$ are both in the range of ϕ, then $\lambda\phi(x) + (1-\lambda)\phi(y)$ may be $\infty - \infty$, which is undefined; in this case we say ϕ is neither convex nor concave. A function with both ∞ and $-\infty$ in its range may still be quasiconcave or quasiconvex.

Definition 2.6 (Saddle and quasisaddle)**.** *Let V_1 and V_2 be real vector spaces, let $X \subseteq V_1$ and $Y \subseteq V_2$, and let $\alpha : X \times Y \to \overline{\mathbb{R}}$. We say that α is* saddle *if for all $x \in X$ the function $\alpha(x, \cdot)$ is concave and for all $y \in Y$ the function $\alpha(\cdot, y)$ is convex. We say that α is* quasisaddle *if for all $x \in X$ the function $\alpha(x, \cdot)$ is quasiconcave and for all $y \in Y$ the function $\alpha(\cdot, y)$ is quasiconvex.*

Definition 2.7 (Semicontinuity)**.** *Let X be a topological space and let $\phi : X \to \overline{\mathbb{R}}$. We say that ϕ is* upper semicontinuous *at $x \in X$ if for all $y \in (\phi(x), \infty]$ there exists some neighborhood U of x on which the value of $\phi(x')$ for $x' \in U$ is less than y. We say that ϕ is* lower semicontinuous *at x if $-\phi$ is upper semicontinuous at x.*

Let Y be another topological space and let $\alpha : X \times Y \to \overline{\mathbb{R}}$ be a function. We say that α is semicontinuous *if for all $x \in X$ the function $\alpha(x, \cdot)$ is upper semicontinuous over all of Y, and for all $y \in Y$ the function $\alpha(\cdot, y)$ is lower semicontinuous over all of X.*

We note the following two useful lemmas about upper and lower semicontinuous functions. These lemmas are standard, but for completeness we reprove them in Appendix A.

Lemma 2.8 (An upper semicontinuous function on a compact set attains its max)**.** *Let X be a nonempty compact topological space, and let $\phi : X \to \overline{\mathbb{R}}$ be a function. Then if ϕ is upper semicontinuous, it attains its maximum, meaning there is some $x \in X$ such that for all $x' \in X$, $\phi(x') \leq \phi(x)$. Similarly, if ϕ is lower semicontinuous, it attains its minimum.*

Lemma 2.9 (A pointwise infimum of upper semicontinuous functions is upper semicontinuous)**.** *Let X be a topological space, let I be a set, and let $\{\phi_i\}_{i \in I}$ be a collection of functions $\phi_i : X \to \overline{\mathbb{R}}$. Then if each ϕ_i is upper semicontinuous, the function $\phi(x) = \inf_{i \in I} \phi_i(x)$ is also upper semicontinuous. Similarly, if each ϕ_i is lower semicontinuous, the pointwise supremum is lower semicontinuous.*

From these lemmas, it follows that if $\alpha : X \times Y \to \overline{\mathbb{R}}$ is semicontinuous, the expressions

$$\inf_{x \in X} \sup_{y \in Y} \alpha(x, y)$$

$$\sup_{y\in Y}\inf_{x\in X}\alpha(x,y)$$

have all the infimums attained if X is nonempty and compact, and all the supremums attained if Y is nonempty and compact. Hence on compact sets, inf-sup theorems become min-max theorems.

The following lemma will also come in useful. We also prove it in Appendix A.

Lemma 2.10 (Quasiconvex functions on convex hulls)**.** *Let V be a real vector space, let $X \subseteq V$, and let $\phi\colon \operatorname{Conv}(X) \to \overline{\mathbb{R}}$ be a function. If ϕ is quasiconvex, then*

$$\sup_{x\in\operatorname{Conv}(X)}\phi(x) = \sup_{x\in X}\phi(x).$$

Similarly, if ϕ is quasiconcave, then

$$\inf_{x\in\operatorname{Conv}(X)}\phi(x) = \inf_{x\in X}\phi(x).$$

2.2 Minimax theorems

We are now ready to state Sion's minimax theorem. Actually, we will need a version of Sion's minimax for extended-real-valued functions, while Sion [Sio58] originally only dealt with real-valued functions; luckily, proving this extension is not hard given Sion's original theorem, and we do so in Appendix A.

Theorem 2.11 (Sion's minimax for extended reals)**.** *Let V_1 and V_2 be real topological vector spaces, and let $X \subseteq V_1$ and $Y \subseteq V_2$ be convex. Let $\alpha : X \times Y \to \overline{\mathbb{R}}$ be semicontinuous and quasisaddle. If either X or Y is compact, then*

$$\inf_{x\in X}\sup_{y\in Y}\alpha(x,y) = \sup_{y\in Y}\inf_{x\in X}\alpha(x,y).$$

Next, we use Sion's minimax theorem to show a minimax theorem for the ratio of positive saddle functions. To do so, we will need the following lemma.

Lemma 2.12. *Let $a, b, c, d \in (0,\infty)$, and let $\lambda \in (0,1)$. Then*

$$\min\left\{\frac{a}{b},\frac{c}{d}\right\} \le \frac{\lambda a + (1-\lambda)c}{\lambda b + (1-\lambda)d} \le \max\left\{\frac{a}{b},\frac{c}{d}\right\}.$$

This still holds if any of a, b, c, d are 0, or if a or c are ∞, so long as we interpret $x/0 = \infty$ for $x \in [0,\infty]$.

Proof. When $a, c \in [0,\infty)$ and $b, d \in (0,\infty)$, it's easy to check that

$$\frac{\lambda a + (1-\lambda)c}{\lambda b + (1-\lambda)d} = \frac{a}{b}\cdot\frac{1}{1+z} + \frac{c}{d}\cdot\frac{z}{1+z},$$

where $z = (1-\lambda)d/\lambda b$. Since $z > 0$, this is a convex combination of a/b and c/d, from which the desired result follows. When $a = \infty$ or $c = \infty$, both the middle expression and the max expression equal ∞, and the result trivially holds. The same thing happens when $b = d = 0$. Finally, when $a, c \in [0,\infty)$ and exactly one of b and d is 0, the max expression is again infinity, and the inequality on the left and side can be easily verified. □

The simple lemma above is enough to imply that a convex function divided by a concave function is quasiconvex, and that a concave function divided by a convex function is quasiconcave.

Lemma 2.13. *Let V be a real topological vector space, and let $X \subseteq V$ be convex. Let $\phi\colon X \to [0,\infty]$ and $\psi\colon X \to [0,\infty)$ be functions, and define $\rho\colon X \to [0,\infty]$ by $\rho(x) \coloneqq \phi(x)/\psi(x)$, with $r/0$ interpreted as ∞ for $r \in [0,\infty]$. Then*

1. *If ϕ is convex and ψ is concave, ρ is quasiconvex.*
2. *If ϕ is concave and ψ is convex, ρ is quasiconcave.*
3. *If ϕ is upper semicontinuous and ψ is lower semicontinuous, ρ is upper semicontinuous.*
4. *If ϕ is lower semicontinuous and ψ is upper semicontinuous, and if ϕ is strictly positive on X, then ρ is lower semicontinuous.*

Proof. We start with (1). Fix $x, y \in X$ and $\lambda \in (0,1)$. Then

$$\begin{aligned}\rho(\lambda x + (1-\lambda)y) &= \frac{\phi(\lambda x + (1-\lambda)y)}{\psi(\lambda x + (1-\lambda)y)} \\ &\leq \frac{\lambda\phi(x) + (1-\lambda)\phi(y)}{\lambda\psi(x) + (1-\lambda)\psi(y)} \\ &\leq \max\left\{\frac{\phi(x)}{\psi(x)}, \frac{\phi(y)}{\psi(x)}\right\} \\ &= \max\{\rho(x), \rho(y)\},\end{aligned}$$

so ρ is quasiconvex, as desired. Here we used the convexity of ϕ and concavity of ψ in the first inequality, and Lemma 2.12 in the second inequality. (2) works similarly:

$$\begin{aligned}\rho(\lambda x + (1-\lambda)y) &= \frac{\phi(\lambda x + (1-\lambda)y)}{\psi(\lambda x + (1-\lambda)y)} \\ &\geq \frac{\lambda\phi(x) + (1-\lambda)\phi(y)}{\lambda\psi(x) + (1-\lambda)\psi(y)} \\ &\geq \min\left\{\frac{\phi(x)}{\psi(x)}, \frac{\phi(y)}{\psi(x)}\right\} \\ &= \min\{\rho(x), \rho(y)\}.\end{aligned}$$

Next, we prove (3). Fix $x \in X$; our goal is to show ρ is upper semicontinuous at x. If $\rho(x) = \infty$, then any function ρ is upper semicontinuous at x by definition, so assume $\rho(x) < \infty$. In particular, this means that $\phi(x) < \infty$ and that $\psi(x) > 0$. Now, fix $y > \rho(x) = \phi(x)/\psi(x)$. By the upper semicontinuity of ϕ, find a neighborhood U_1 of x on which $\phi(\cdot)$ is at most $\phi(x) + \epsilon$ (with $\epsilon > 0$ to be chosen later). By the lower semicontinuity of ψ, find a neighborhood U_2 of x on which $\psi(\cdot)$ is at least $\psi(x) - \epsilon$. Setting $U \coloneqq U_1 \cap U_2$, we see that on U we have $\rho(\cdot) \leq (\phi(x) + \epsilon)/(\psi(x) - \epsilon)$, assuming we pick $\epsilon < \psi(x)$. We now simply pick ϵ small enough that this expression is less than y, giving us a neighborhood U of x on which $\rho(\cdot)$ is less than y, as desired.

Finally, we prove (4). As before, we fix $x \in X$. Our goal is to show $\rho(x)$ is lower semicontinuous in at x. Let $y < \rho(x)$. We seek a neighborhood U of x on which $\rho(\cdot) > y$. To start with, the upper semicontinuity of ψ ensures there is a neighborhood U_1 of x on which $\psi(\cdot) < \psi(x) + \epsilon$, with $\epsilon > 0$ arbitrarily small. Now, if $\phi(x) = \infty$, then $\rho(x) = \infty$. In this case, the lower semicontinuity of ϕ ensures there is a neighborhood U_2 on which $\phi(\cdot)$ is at least z, with $z \in \mathbb{R}$ is arbitrarily large. Then in $U_1 \cap U_2$, the value of $\rho(\cdot)$ is also arbitrarily large, and can be made to exceed $y \in \mathbb{R}$ given appropriate choices of z and ϵ. Alternatively, if $\phi(x) < \infty$, then there is a neighborhood U_2 on

which $\phi(\cdot) > \phi(x) - \epsilon$. In this case, on $U_1 \cap U_2$ we have $\rho(\cdot) > (\phi(x) - \epsilon)/(\psi(x) + \epsilon)$. By picking ϵ sufficiently small, we can again get a neighborhood $U_1 \cap U_2$ of x on which $\rho(\cdot) > y$, meaning that ρ is lower semicontinuous. □

We now state the minimax theorem for the ratio of two positive saddle functions. In the statement below, it may help to think of $\mathcal{R}$ as a set of randomized algorithms, and to think of Δ as the set of all probability distributions over a finite input set. Further, think of $\text{cost}(R, \mu)$ as measuring the cost of the algorithm R when run on μ (for some models, this will depend only on R and not on μ), and think of $\text{score}(R, \mu)$ as quantifying the success or bias that the algorithm R achieves against input distribution μ.

Theorem 2.14 (Minimax theorem for the positive ratio of saddle functions)**.** *Let V_1 and V_2 be real topological vector spaces. Let $\mathcal{R} \subseteq V_1$ be convex, and let $\Delta \subseteq V_2$ be nonempty, convex, and compact. Let the function $\text{cost}\colon \mathcal{R} \times \Delta \to (0, \infty]$ be semicontinuous and saddle, and let the function $\text{score}\colon \mathcal{R} \times \Delta \to [0, \infty)$ be such that its negation, $-\text{score}$, is semicontinuous and saddle. Then using $x/0 = \infty$ for $x \in (0, \infty]$, we have*

$$\inf_{R \in \mathcal{R}} \max_{\mu \in \Delta} \frac{\text{cost}(R, \mu)}{\text{score}(R, \mu)} = \max_{\mu \in \Delta} \inf_{R \in \mathcal{R}} \frac{\text{cost}(R, \mu)}{\text{score}(R, \mu)},$$

and the maximums are attained.

Proof. Let $\alpha\colon \mathcal{R} \times \Delta \to (0, \infty]$ be defined by $\alpha(R, \mu) := \text{cost}(R, \mu)/\text{score}(R, \mu)$, with $x/0$ interpreted as ∞ for $x \in (0, \infty]$. For any fixed $\mu \in \Delta$, the function $\alpha(\cdot, \mu)$ is quasiconvex and lower semicontinuous by Lemma 2.13. Similarly, for any fixed $R \in \mathcal{R}$, the function $\alpha(R, \cdot)$ is concave and upper semicontinuous by Lemma 2.13. Hence α is semicontinuous and quasisaddle, and the desired minimax theorem follows from Theorem 2.11. Furthermore, since Δ is nonempty and compact, the supremums are attained as maximums by Lemma 2.9 and Lemma 2.8. □

Finally, we will need two extensions of this theorem. First, we will want to allow the denominator to be a function of the form $\text{score}(R, \mu)^+$, where the $+$ superscript denotes the maximum of $\text{score}(R, \mu)$ with 0, and where we only know about saddle properties of $\text{score}(R, \mu)$, not of $\text{score}(R, \mu)^+$. To do this, we need to show such a maximum with 0 preserves the properties we care about. We have the following lemma, which we prove in Appendix A.

Lemma 2.15. *Let V be a real topological vector space, and let $X \subseteq V$ be convex. For a function $\psi\colon X \to \overline{\mathbb{R}}$, let ψ^+ denote the function $\psi^+(x) = \max\{\psi(x), 0\}$. Then this operation on ψ preserves convexity, quasiconvexity, quasiconcavity, upper semicontinuity, and lower semicontinuity, but not concavity.*

This lemma is useful, but doesn't quite give us everything we need, because the operation ψ^+ does not preserve concavity. We will need the following additional lemma, which says that Lemma 2.13 also works when dividing by ψ^+, despite its lack of concavity.

Lemma 2.16. *Let V be a real topological vector space, and let $X \subseteq V$ be convex. Let $\phi\colon X \to [0, \infty]$ and $\psi\colon X \to [-\infty, \infty)$ be functions, and define $\rho\colon X \to [0, \infty]$ by $\rho(x) := \phi(x)/\psi(x)^+$, with $r/0$ interpreted as ∞ for $r \in [0, \infty]$. Then if ϕ is convex and ψ is concave, ρ is quasiconvex.*

Proof. Fix $x, y \in X$ and $\lambda \in (0, 1)$. If $\psi(x) > 0$ and $\psi(y) > 0$, we have $\rho(\lambda x + (1 - \lambda)y) \leq \max\{\rho(x), \rho(y)\}$ using the same argument as in Lemma 2.13. On the other hand, if $\psi(x) \leq 0$ or $\psi(y) \leq 0$, then we have $\max\{\rho(x), \rho(y)\} = \infty$, and the inequality $\rho(\lambda x + (1-\lambda)y) \leq \max\{\rho(x), \rho(y)\}$ trivially holds. □

The second extension we will need in our final minimax theorem is to the case where the numerator is allowed to be 0. Unfortunately, as we can see from the statement of Lemma 2.13, the ratio does not preserve lower semicontinuity in this setting. We will need to impose some additional conditions on the cost and score functions, particularly with regard to their behavior around 0.

Definition 2.17. *We say that* $\text{cost}\colon \mathcal{R}\times\Delta \to [0,\infty]$ *and* $\text{score}\colon \mathcal{R}\times\Delta \to [-\infty,\infty)$ *are* well-behaved *if the following conditions hold:*

1. *(Finite cost and score can be achieved.) For each* $\mu \in \Delta$*, there is some* $R \in \mathcal{R}$ *such that* $\text{cost}(R,\mu) > 0$*,* $\text{cost}(R,\mu) < \infty$*, and* $\text{score}(R,\mu) > 0$*.*
2. *(A zero-cost algorithm has zero cost regardless of the input.) For each* $R \in \mathcal{R}$*, either* $\text{cost}(R,\mu) = 0$ *for all* $\mu \in \Delta$*, or else* $\text{cost}(R,\mu) > 0$ *for all* $\mu \in \Delta$*.*
3. *(Mixing a zero-cost algorithm with a nonzero-cost algorithm gives a nonzero-cost algorithm.) For each* $\mu \in \Delta$*, if* $R, R' \in \mathcal{R}$ *are such that* $\text{cost}(R,\mu) = 0$ *and* $\text{cost}(R',\mu) > 0$*, then* $\text{cost}(\lambda R + (1-\lambda)R',\mu) > 0$ *for all* $\lambda \in (0,1)$*.*

Finally, we are ready for our main workhorse minimax theorem.

Theorem 2.18. *Let* V *be a real topological vector space, and let* $\mathcal{R} \subseteq V$ *be convex. Let* S *be a nonempty finite set, and let* Δ *be the set of all probability distributions over* S*, viewed as a subset of* $\mathbb{R}^{|S|}$*. Let* $\text{cost}\colon \mathcal{R}\times\Delta \to [0,\infty]$ *be semicontinuous and saddle, and let* $\text{score}\colon \mathcal{R}\times\Delta \to [-\infty,\infty)$ *be such that its negation,* $-\text{score}$*, is semicontinuous and saddle. Suppose* cost *and* score *are well-behaved. Then using the convention* $r/0 = \infty$ *for all* $r \in [0,\infty]$*, we have*

$$\inf_{R\in\mathcal{R}} \max_{\mu\in\Delta} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+} = \max_{\mu\in\Delta} \inf_{R\in\mathcal{R}} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+}.$$

Moreover, if $\text{cost}(R,\cdot)$ *and* $\text{score}(R,\cdot)$ *are both linear in* μ *for each* $R \in \mathcal{R}$*, then*

$$\inf_{R\in\mathcal{R}} \max_{x\in S} \frac{\text{cost}(R,x)}{\text{score}(R,x)^+} = \max_{\mu\in\Delta} \inf_{R\in\mathcal{R}} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+}.$$

Further, all of the above maximums are attained.

Proof. First, note that if $S = \{x_1, x_2, \ldots, x_{|S|}\}$, then we can view Δ as the convex hull of the set $\{e_1, e_2, \ldots, e_{|S|}\} \subseteq \mathbb{R}^{|S|}$, where the e_i are the unit vectors $e_i = (0,0,\ldots,0,1,0,0,\ldots,0)$ with the 1 at position i. Hence Δ is convex. It is also closed and bounded, making it compact. We identify e_i with x_i, so that $\Delta = \text{Conv}(S)$.

Note that since each $R \in \mathcal{R}$ has either cost 0 for all μ or cost greater than 0 for all μ, we can define the set $\mathcal{R}' \subseteq \mathcal{R}$ of R with nonzero cost. Now, on $\mathcal{R}'$, the function $\alpha(R,\mu) = \text{cost}(R,\mu)/\,\text{score}(R,\mu)^+$ is semicontinuous and quasisaddle by Lemma 2.13 together with Lemma 2.16 and Lemma 2.15. Additionally, Δ is nonempty, convex, and compact. Thus by Theorem 2.11, we know that

$$\inf_{R\in\mathcal{R}'} \max_{\mu\in\Delta} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+} = \max_{\mu\in\Delta} \inf_{R\in\mathcal{R}'} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+},$$

with the maximums attained.

What we want to show is this statement with the infimums over $\mathcal{R}$ instead of $\mathcal{R}'$. The inf-sup is always at least the sup-inf for every function, so we need only show that the sup-inf is at least the inf-sup. Moreover, since expanding the domain can only decrease the infimum, we know that

$$\max_{\mu\in\Delta} \inf_{R\in\mathcal{R}'} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+} = \inf_{R\in\mathcal{R}'} \max_{\mu\in\Delta} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+} \geq \inf_{R\in\mathcal{R}} \max_{\mu\in\Delta} \frac{\text{cost}(R,\mu)}{\text{score}(R,\mu)^+},$$

where the rightmost maximum is attained by virtue of the fact that we know it is attained when $R \in \mathcal{R}'$, and if $R \in \mathcal{R} \setminus \mathcal{R}'$, then $\operatorname{cost}(R,\mu)/\operatorname{score}(R,\mu)^+$ is either 0 or ∞ for all μ. Thus we only need to show that the max-inf over $\mathcal{R}$ is at least the max-inf over $\mathcal{R}'$, and that the former maximum is attained.

To see this, let $\mu \in \Delta$ be the maximizing μ for the expression

$$\max_{\mu\in\Delta} \inf_{R\in\mathcal{R}'} \frac{\operatorname{cost}(R,\mu)}{\operatorname{score}(R,\mu)^+}.$$

Suppose by contradiction that there was some $\hat{R} \in \mathcal{R} \setminus \mathcal{R}'$ such that

$$\frac{\operatorname{cost}(\hat{R},\mu)}{\operatorname{score}(\hat{R},\mu)^+} < \inf_{R\in\mathcal{R}'} \frac{\operatorname{cost}(R,\mu)}{\operatorname{score}(R,\mu)^+}.$$

Since $\hat{R} \in \mathcal{R} \setminus \mathcal{R}'$, we must have $\operatorname{cost}(\hat{R},\mu) = 0$. Since $0/\operatorname{score}(\hat{R},\mu)^+$ is less than something, and since we're interpreting $0/0 = \infty$, we must have $\operatorname{score}(\hat{R},\mu) > 0$, so that $0/\operatorname{score}(\hat{R},\mu)^+ = 0$. We wish to show that $\inf_{R\in\mathcal{R}'} \operatorname{cost}(R,\mu)/\operatorname{score}(R,\mu)^+ = 0$. To this end, pick $\epsilon > 0$. We will find $R \in \mathcal{R}'$ such that $\operatorname{cost}(R,\mu)/\operatorname{score}(R,\mu)^+ < \epsilon$. The idea is to pick some $R' \in \mathcal{R}'$ such that $\operatorname{cost}(R',\mu) < \infty$ and $\operatorname{score}(R',\mu) > 0$, as guaranteed by the well-behaved condition on cost and score. Then set $R \coloneqq \lambda R' + (1-\lambda)\hat{R}$, with $\lambda > 0$ extremely small. Now, the well-behaved property of cost says that $\operatorname{cost}(R,\mu) > 0$, so $R \in \mathcal{R}'$. By convexity, we also have $\operatorname{cost}(R,\mu) = \operatorname{cost}(\lambda R' + (1-\lambda)\hat{R},\mu) \le \lambda \operatorname{cost}(R',\mu) + (1-\lambda)\operatorname{cost}(\hat{R},\mu) = \lambda \operatorname{cost}(R',\mu)$, and by the concavity of $\operatorname{score}(\cdot,\mu)$, we have $\operatorname{score}(R,\mu) = \operatorname{score}(\lambda R' + (1-\lambda)\hat{R},\mu) \ge \lambda \operatorname{score}(R',\mu) + (1-\lambda)\operatorname{score}(\hat{R},\mu) \ge (1/2)\operatorname{score}(\hat{R},\mu)$, assuming $\lambda \le 1/2$.

This means that $\operatorname{score}(R,\mu)$ and $\operatorname{score}(\hat{R},\mu)$ are both positive, and $\operatorname{cost}(R,\mu)/\operatorname{score}(R,\mu) \le 2\lambda \operatorname{cost}(\hat{R},\mu)/\operatorname{score}(\hat{R},\mu)$. Since $\operatorname{cost}(\hat{R},\mu) < \infty$, setting $\lambda > 0$ to be small causes the ratio $\operatorname{cost}(R,\mu)/\operatorname{score}(R,\mu)^+$ to be arbitrarily close to 0, as desired. It follows that there exists $\mu \in \Delta$ such that

$$\inf_{R\in\mathcal{R}} \frac{\operatorname{cost}(R,\mu)}{\operatorname{score}(R,\mu)} \ge \inf_{R\in\mathcal{R}} \max_{\mu'\in\Delta} \frac{\operatorname{cost}(R,\mu')}{\operatorname{score}(R,\mu')},$$

and since the inf-max is always at least the max-inf, there does not exist a μ for which the left-hand infimum is any larger; thus we get the desired result and the maximum is attained.

Finally, suppose that $\operatorname{cost}(R,\cdot)$ and $\operatorname{score}(R,\cdot)$ are linear for each $R \in \mathcal{R}$. In that case, $\operatorname{cost}(R,\cdot)$ is convex and $\operatorname{score}(R,\cdot)$ is concave, which means that $\operatorname{cost}(R,\cdot)/\operatorname{score}(R,\cdot)^+$ is quasiconvex on Δ by Lemma 2.16. Then Lemma 2.10 implies that the maximum over $\mu \in \operatorname{Conv}(S)$ is attained at a point in S. Moreover, if $R \in \mathcal{R} \setminus \mathcal{R}'$, then the maximum over $\mu \in \Delta$ evaluates to either 0 or ∞. If it is 0, then it is clearly also attained in S. If it is ∞, it means some $\mu \in \Delta$ has $\operatorname{score}(R,\mu) \le 0$; the concavity of $\operatorname{score}(R,\cdot)$ then gives us some $x \in S$ such that $\operatorname{score}(R,x) \le \operatorname{score}(R,\mu)$, meaning there is a point $x \in S$ on which $\operatorname{score}(R,x)^+ = 0$ and $\operatorname{cost}(R,x)/\operatorname{score}(R,x)^+ = \infty$, as desired. □

Theorem 2.18 is the main tool we will use to prove minimax theorems for algorithmic models. We will usually apply it in a setting where $\mathcal{R}$ is a set of algorithms, S is a finite input set, Δ is a set of distributions over the inputs, $\operatorname{cost}(R,\mu)$ is a cost measure for the performance of an algorithm against a distribution, and $\operatorname{score}(R,\mu)$ is some kind of success measure. We will sometimes choose $\operatorname{score}(R,\mu) = \operatorname{bias}_f(R,\mu)$, where $\operatorname{bias}_f(R,\mu)$ is the bias R achieves against distribution μ in computing f.

We will generally combine Theorem 2.18 with an amplification theorem; such a theorem will turn the left hand side $\inf_R \max_x \operatorname{cost}(R,x)/\operatorname{score}(R,x)$ into something more familiar, such as $\inf_R \max_x \operatorname{cost}(R,x)$ where the infimum is restricted to algorithms R which achieve at least constant

bias (i.e. bounded error) on each input. With such an amplification theorem, the minimax result will guarantee the existence of a hard distribution μ against which $\mathrm{cost}(R, \mu)/\mathrm{score}(R, \mu)$ is large for all R; this means μ is hard to solve even to small bias.

While the above strategy works for models that can be amplified linearly in the bias (going from bias γ to constant bias using $O(1/\gamma)$ overhead), such as quantum query complexity, for randomized algorithms the situation is more complicated. For randomized algorithms, we may instinctively want to use something like $\mathrm{score}(R, \mu) = \mathrm{bias}_f(R, \mu)^2$, but this does not work as it does not satisfy the right saddle properties. Instead, we introduce a new way of evaluating the success of randomized algorithms, called *scoring rules*. Evaluation via scoring rules ends up being the "correct" way to measure the success of a randomized algorithm, and has more elegant properties than simply the bias. It is also highly intuitive: to evaluate the success of an algorithm, we require it to give a confidence prediction for whether the output is 0 or 1, and then we score the prediction using a scoring rule which incentivizes honesty (that is, a scoring rule that causes a Bayesian agent who wishes to maximize her expected score to output her true subjective probability).

3 Forecasting algorithms

In this section we introduce the notion of *forecasting algorithms*, which output not just a $\{0, 1\}$ guess at the function value but also a confidence parameter $q \in [0, 1]$ for that prediction. These algorithms will be scored using a *scoring rule*, which rewards them 1 point for a correct prediction made with perfect confidence, and 0 points for a confidence of 1/2. As we will see, normal algorithms can be converted into forecasting algorithms and vice versa, and the expected score of the forecasting version can often be related to the bias of the algorithm in its regular (discrete outputs) form.

3.1 Scoring rules

Definition 3.1 (Scoring rule)**.** *A* scoring rule *is a function $s : [0, 1] \to [-\infty, 1]$ such that $s(1) = 1$, $s(1/2) = 0$, and $s(\cdot)$ is increasing over $[0, 1]$. We say a scoring rule is* proper *if for each $p \in (0, 1)$, the expression $ps(q) + (1 - p)s(1 - q)$ is uniquely maximized at $q = p$.*

Generally, if a forecasting algorithm outputs $q \in [0, 1]$, we will interpret it as assigning confidence q to the output 1 and confidence $1 - q$ to the output 0; we give it score $s(q)$ if the right answer was 1, and score $s(1 - q)$ if the right answer was 0. A proper scoring rule is therefore a scoring rule that incentivizes the algorithm to output $q = p$ in the case where the right answer is sampled from Bernoulli(p). In other words, a proper scoring rule is one that incentivizes a Bayesian agent to output her true subjective probability for the outcome being 1.

Definition 3.2. *We define the following scoring rules.*

1. $\mathrm{hs}(q) \coloneqq 1 - \sqrt{\frac{1-q}{q}}$
2. $\mathrm{Brier}(q) \coloneqq 1 - 4(1 - q)^2$
3. $\mathrm{bias}(q) \coloneqq 1 - 2(1 - q)$
4. $\mathrm{ls}(q) \coloneqq 1 - \log(1/q)$.

We note that $\mathrm{Brier}(\cdot)$ and $\mathrm{ls}(\cdot)$ are known as the Brier scoring rule and logarithmic scoring rule, respectively, and are well-known in the literature. The Brier scoring rule is useful because it is a proper scoring rule which is bounded (that is, $s(q) \in [-3, 1]$ for all $q \in [0, 1]$, instead of $s(\cdot)$ diverging

to $-\infty$ at 0). The logarithmic scoring rule has an information-theoretic interpretation, with the algorithm essentially starting at score 1 and losing an amount of score depending on its "surprisal" at the correct outcome.

The scoring rule bias$(\cdot)$ is not proper, but as we will see, it is closely related to the bias an algorithm will make. Finally, the scoring rule hs$(\cdot)$ will be the most useful of the bunch for our purposes. Despite not having any intuitive interpretation and not being bounded, it is an incredibly convenient scoring rule due to the fact that it can be *amplified*, as we will see. hs$(\cdot)$ has been previously studied (for example in [BSS05], where it is called the "boosting loss" due to its relationship with boosting), but we believe its amplification property has not been previously known (we prove this amplification property later on in Lemma 3.10; this ends up being a key ingredient of our minimax theorems).

Lemma 3.3. hs*, Brier, and* ls *are proper scoring rules.* bias *is a scoring rule which is not proper.*

This lemma can be proven using elementary calculus, and we do so in Appendix B.

3.2 Distance measures

Fascinatingly, the above scoring rules all correspond to well-known distance measures between probability distributions. To describe the correspondence, we first start by defining the following distance measures.

Definition 3.4. *For probability distributions ν_0 and ν_1 over a finite domain P, define*

$$\Delta(\nu_0, \nu_1) \coloneqq \frac{1}{2} \sum_{x \in P} |\nu_0[x] - \nu_1[x]| \qquad \textit{(Total variation)}$$

$$\mathrm{h}^2(\nu_0, \nu_1) \coloneqq \frac{1}{2} \sum_{x \in P} (\sqrt{\nu_0[x]} - \sqrt{\nu_1[x]})^2 \qquad \textit{(Hellinger)}$$

$$\mathrm{S}^2(\nu_0, \nu_1) \coloneqq \frac{1}{2} \sum_{x \in P} \frac{(\nu_0[x] - \nu_1[x])^2}{\nu_0[x] + \nu_1[x]} \qquad \textit{(Symmetrized } \chi^2\textit{)}$$

$$\mathrm{JS}(\nu_0, \nu_1) \coloneqq \frac{1}{2} \sum_{x \in P} \nu_0[x] \log \frac{2\nu_0[x]}{\nu_0[x] + \nu_1[x]} + \nu_1[x] \log \frac{2\nu_1[x]}{\nu_0[x] + \nu_1[x]} \qquad \textit{(Jensen-Shannon).}$$

The above measures give the distance between two probability distributions. We will sometimes want to have an asymmetric distance that is weighted towards one of the two distributions; while these asymmetric distances look strange at first, they show up naturally in the study of scoring rules. We extend the above distance measures as follows.

Definition 3.5. *Given probability distributions ν_0 and ν_1 over a finite domain P, as well as a weight $w \in [0, 1]$, set $\nu = (1 - w)\nu_0 + w\nu_1$. Let R be the random variable over $x \in P$ defined by $R(x) \coloneqq |(1 - w)\nu_0[x] - w\nu_1[x]|/\nu[x]$ for all $x \in P$. Then define*

$$\Delta(\nu_0, \nu_1, w) \coloneqq \mathop{\mathbb{E}}_{x \leftarrow \nu}[R]$$

$$\mathrm{h}^2(\nu_0, \nu_1, w) \coloneqq \mathop{\mathbb{E}}_{x \leftarrow \nu}[1 - \sqrt{1 - R^2}]$$

$$\mathrm{S}^2(\nu_0, \nu_1, w) \coloneqq \mathop{\mathbb{E}}_{x \leftarrow \nu}[R^2]$$

$$\mathrm{JS}(\nu_0, \nu_1, w) \coloneqq \mathop{\mathbb{E}}_{x \leftarrow \nu}\left[1 - H\left(\frac{1 + R}{2}\right)\right],$$

where $H(\alpha) \coloneqq \alpha \log 1/\alpha + (1 - \alpha) \log 1/(1 - \alpha)$ is the binary entropy function.

It's not hard to see that when $w = 1/2$, the expressions in Definition 3.5 equal the ones in Definition 3.4. Perhaps surprisingly, the distance measures h^2, S^2, and JS are all related to each other by a constant factor.

Lemma 3.6 (Relations between distance measures)**.** *When applied to fixed ν_0, ν_1, and w, the distance measures satisfy*

$$\frac{\mathrm{S}^2}{2} \le 1 - \sqrt{1 - \mathrm{S}^2} \le \mathrm{h}^2 \le \mathrm{JS} \le \mathrm{S}^2$$

as well as

$$\Delta^2 \le \mathrm{S}^2 \le \Delta.$$

We also have $\mathrm{JS} \le \mathrm{h}^2 / \ln 2$ *and* $\mathrm{S}^2 \le (\ln 4)\,\mathrm{JS}$.

While these relationships are certainly known in the literature, it is hard to chase down good citations (though see [Tøp00; MCAL17] for parts of this result); in any case, we prove Lemma 3.6 in Appendix B.

3.3 The highest achievable expected score is a distance measure

Consider the following problem: suppose distributions ν_0 and ν_1 are known (for example, perhaps they are the distributions of the transcript of a fixed randomized algorithm when run on a known 0-distribution and a known 1-distribution, respectively). Further, suppose a Bernoulli(w) process generates a bit $b \in \{0, 1\}$, and then a sample $x \leftarrow \mu_b$ is provided. We assume the parameter w is known. What is the best algorithm for predicting b given x, assuming you wish to maximize the expected score according to one of the scoring rules $\mathrm{hs}(\cdot), \mathrm{Brier}(\cdot), \mathrm{ls}(\cdot), \mathrm{bias}(\cdot)$? It turns out that best attainable expected score is exactly the distance between ν_0 and ν_1 according to the distance measures $\mathrm{h}^2, \mathrm{S}^2, \mathrm{JS}, \Delta$, respectively. To prove this, we introduce the following definitions.

Definition 3.7. *For a scoring rule $s : [0, 1] \to [-\infty, 1]$, we define $s_1(p) \coloneqq s(p)$ and $s_0(p) \coloneqq s(1-p)$. This way, if a forecasting algorithm outputs p and the real outcome is b, the score of this prediction will be $s_b(p)$.*

Definition 3.8 (Expected score notation)**.** *Let S be a finite set, and let $\phi : S \to [0, 1]$ be a function representing predictions. Let ν be a distribution over S, let $P(x)$ be a Boolean-valued random variable for each $x \in S$ representing the correct outcome, and let $s : [0, 1] \to [-\infty, 1]$ be a scoring rule. The* expected score *of ϕ, denoted* $\mathrm{score}_s(\phi, \nu, P)$*, is defined as*

$$\mathrm{score}_s(\phi, \nu, P) \coloneqq \mathbb{E}_{x \leftarrow \nu} \mathbb{E}_{b \leftarrow P(x)} [s_b(\phi(x))].$$

In these expectations, if a value of ∞ or $-\infty$ occurs with probability 0, we set $0 \cdot \infty \coloneqq 0$.

We can also extend the score notation to the case where $\phi(x)$ outputs a probability distribution over $[0, 1]$ instead of always outputting a deterministic prediction given the observation x. We won't worry about this case for now.

Equipped with these definitions, we are now ready to prove the correspondence between scoring rules and distance measures. This correspondence appears to be known in the literature (indeed, variants of it seem to have been rediscovered many times); see [RW11] for an overview. However, the form we need here is somewhat different from the usual form in the literature, which usually discusses divergences instead of distances. We therefore include the proof for completeness.

Lemma 3.9. *Let ν_0 and ν_1 be probability distributions over a finite set S, and let $w \in [0,1]$. Let $M_s(\nu_0, \nu_1, w)$ be the maximum possible score of for predicting $b \leftarrow \text{Bernoulli}(w)$ given $x \leftarrow \nu_b$, where ν_0, ν_1, and w known. That is, $M_s(\nu_0, \nu_1, w)$ is the maximum over choice of $\phi : S \to [0,1]$ of the expression $\text{score}_s(\phi, \nu, P)$, where $\nu = (1-w)\nu_0 + w\nu_1$ and $P(x)$ is the posterior probability distribution of b given prior $\text{Bernoulli}(w)$ and observation $x \leftarrow \nu_b$. Then*

$$\begin{aligned} M_{\text{bias}}(\nu_0, \nu_1, w) &= \Delta(\nu_0, \nu_1, w) \\ M_{\text{hs}}(\nu_0, \nu_1, w) &= \text{h}^2(\nu_0, \nu_1, w) \\ M_{\text{Brier}}(\nu_0, \nu_1, w) &= \text{S}^2(\nu_0, \nu_1, w) \\ M_{\text{ls}}(\nu_0, \nu_1, w) &= \text{JS}(\nu_0, \nu_1, w). \end{aligned}$$

Proof. Consider a fixed $x \in D$. The contribution of x to the expected score of ϕ (with respect to scoring rule s) is simply $(1-w)\nu_0[x]s_0(\phi(x)) + w\nu_1[x]s_1(\phi(x)) = (1-w)\nu_0[x]s(1-\phi(x)) + w\nu_1[x]s(\phi(x))$. The total expected score of ϕ is therefore the sum over $x \in D$ of the above expression. The function ϕ which maximizes the expected score is simply the one where $\phi(x) = q$, where q maximizes the expression $(1-w)\nu_0[x]s(1-q)+w\nu_1[x]s(q)$. Now, the expression we wish to maximize has the form $\nu[x] \cdot ((1-p)s(1-q)+ps(q))$, where $p = w\nu_1[x]/\nu[x]$. Hence, if s is proper, the unique maximum occurs at $q = p = w\nu_1[x]/\nu[x]$. This means that for the maximizing ϕ, the contribution of each x to the expected score is $(1-w)\nu_0[x]s((1-w)\nu_0[x]/\nu[x]) + w\nu_1[x]s(w\nu_1[x]/\nu[0])$, assuming s is proper.

For $s \in \{\text{hs}, \text{ls}, \text{Brier}\}$, the scoring rule s is indeed proper, meaning that we have a closed expression for the maximum possible expected score. Setting $R[x] \coloneqq |w\nu_1[x] - (1-w)\nu_0[x]|/\nu[x]$, it's not hard to check that for hs, the contribution of each x is $\nu[x](1 - \sqrt{1 - R[x]^2})$, for ls, the contribution of each x is $\nu[x](1 - H((1+R[x])/2))$, and for Brier, the contribution of each x is $\nu[x]R[x]^2$, as desired.

It remains to deal with $s = \text{bias}$. The contribution of each x is the maximum possible value of $(1-w)\nu_0[x]\,\text{bias}(1-q) + w\nu_1[x]\,\text{bias}(q)$ for $q \in [0,1]$. Since $\text{bias}(q) = 2q-1$, it's not hard to see that the maximizing value of q is $q = 0$ when $(1-w)\nu_0[x] > w\nu_1[x]$, $q = 1$ when $w\nu_1[x] > (1-w)\nu_0[x]$, and when $(1-w)\nu_0[x] = w\nu_1[x]$, the contribution of x to the score is 0 regardless of the value of q. The contribution of x to the maximum score is therefore $\nu[x]R[x]$, as desired. □

We note that in the statement of Lemma 3.9, we are implicitly assuming that the predictive algorithms are deterministic: that given x, one is only allowed to output a deterministic prediction $\phi(x) \in [0,1]$ instead of a random choice of prediction. However, it is not hard to see that randomized algorithms won't help in this setting, since we are maximizing the expected score, which is a linear function of the probabilities inside the randomized choice. That is to say, if the randomized algorithm chooses (on input x) to output a with probability p and b with probability $1-p$, then the final score of this algorithm will be a linear function of p, and hence the optimal choice of p will be either 0 or 1. Hence Lemma 3.9 also characterizes the best possible score of a randomized prediction algorithm with respect to those four scoring rules.

3.4 Linear amplification of hs score

From here on out, we consider only the $\text{hs}(\cdot)$ scoring rule (and occasionally $\text{bias}(\cdot)$, which will correspond to the bias of a randomized algorithm). We will sometimes omit the subscript in the expression $\text{score}_s(\phi, \nu, P)$ when $s = \text{hs}$.

We now proceed to show a few nice properties of the hs scoring rule. First among them is the amplification property. We believe this property (which is crucial for our purposes) has not previously appeared in the literature.

Lemma 3.10 (Amplification of hs)**.** *Let S be a finite set, and let $\phi : S \to [0,1]$ represent a prediction function. Then for each $k \in \mathbb{N}$, there is a function $\phi^{(k)} : S^k \to [0,1]$ such that for any distribution ν over S, we have*

$$\mathrm{score}_{\mathrm{hs}}(\phi^{(k)}, \nu^{\otimes k}, 0) \geq 1 - (1 - \mathrm{score}_{\mathrm{hs}}(\phi, \nu, 0))^k$$
$$\mathrm{score}_{\mathrm{hs}}(\phi^{(k)}, \nu^{\otimes k}, 1) \geq 1 - (1 - \mathrm{score}_{\mathrm{hs}}(\phi, \nu, 1))^k.$$

Furthermore, equality holds except when $\mathrm{score}_{\mathrm{hs}}(\phi, \nu, 0) = \mathrm{score}_{\mathrm{hs}}(\phi, \nu, 1) = -\infty$. Here 0 and 1 are interpreted as the constant functions $0(x) = 0$ and $1(x) = 1$.

Informally, this lemma is saying the following. Consider a randomized forecasting algorithm R, which takes input x and outputs a confidence $q \in [0,1]$ representing its belief that $f(x) = 1$. Evaluate this algorithm according to its *worst-case expected score* with respect to the $\mathrm{hs}(\cdot)$ scoring rule. That is to say, for each input $x \in f^{-1}(1)$, consider the expectation $\mathbb{E}[\mathrm{hs}(R(x))]$ of the expected score R gets when run on x, and for each $x \in f^{-1}(0)$, consider the analogous expectation $\mathbb{E}[\mathrm{hs}(1 - R(x))]$. Then take the minimum η of all these expected scores, minimizing over any $x \in \mathrm{Dom}(f)$. This is the worst-case expected score of R. The lemma then says that we can run R on x several times, say k times independently, and combine the confidence outputs $q_1, q_2, \ldots, q_k$ in such a way that the new algorithm has worst-case expected score equal to $1 - (1 - \eta)^k$.

Proof. We define $\phi^{(k)}(x_1 \ldots x_k)$ as follows. First, if it holds that some pair (x_i, x_j) in the input satisfies $\phi(x_i) = 0$ and $\phi(x_j) = 1$, we define $\phi^{(k)}(x_1 \ldots x_k) \coloneqq 1/2$. Otherwise, we set $\phi^{(k)}(x_1 \ldots x_k) \coloneqq \left(1 + \prod_{i=1}^k \frac{1-\phi(x_i)}{\phi(x_i)}\right)^{-1}$, where we interpret $1/0 = \infty$ if it occurs (we need not interpret $\infty \cdot 0$ since that will only occur if $\phi(x_i) = 0$ and $\phi(x_j) = 1$ for some i and j). Note that if $\phi(x) = 0$ and $\phi(x') = 1$ for $x, x' \in S$ that have nonzero weight in ν, then we have $\mathrm{score}_{\mathrm{hs}}(\phi, \nu, 0) = \mathrm{score}_{\mathrm{hs}}(\phi, \nu, 1) = -\infty$, so the desired inequalities trivially hold. Otherwise, for $b \in \{0,1\}$ we write

$$\begin{aligned}
\mathrm{score}_{\mathrm{hs}}(\phi^{(k)}, \nu^{\otimes k}, b) &= \mathbb{E}_{x_1 \ldots x_k \leftarrow \nu^{\otimes k}} \left[1 - \sqrt{\left(\frac{\phi^{(k)}(x_1 \ldots x_k)}{1 - \phi^{(k)}(x_1 \ldots x_k)}\right)^{(-1)^b}}\right] \\
&= 1 - \mathbb{E}_{x_1 \ldots x_k \leftarrow \nu^{\otimes k}} \left[\sqrt{\prod_{i=1}^k \left(\frac{\phi(x_i)}{1 - \phi(x_i)}\right)^{(-1)^b}}\right] \\
&= 1 - \prod_{i=1}^k \mathbb{E}_{x_i \leftarrow \nu} \left[\sqrt{\left(\frac{\phi(x_i)}{1 - \phi(x_i)}\right)^{(-1)^b}}\right] \\
&= 1 - (\mathbb{E}_{x \leftarrow \nu}[1 - \mathrm{hs}_b(\phi(x))])^k \\
&= 1 - (1 - \mathrm{score}_{\mathrm{hs}}(\phi, \nu, b))^k.
\end{aligned}$$

Note that equality holds except in the case where $\mathrm{score}_{\mathrm{hs}}(\phi, \nu, 0) = \mathrm{score}_{\mathrm{hs}}(\phi, \nu, 1) = -\infty$. □

The following lemma will be convenient when using this amplification theorem. We prove it in Appendix B.

Lemma 3.11. *If $x \in [0,1]$ and $k \in [1, \infty)$, we have*

$$\frac{1}{2} \min\{kx, 1\} \leq 1 - (1 - x)^k \leq \min\{kx, 1\}.$$

3.5 Bias and hs score

Another nice property of hs is that it is at most bias.

Lemma 3.12. *For all $q \in [0,1]$, we have* $\mathrm{hs}(q) \le \mathrm{bias}(q)$.

Proof. Recall that $\mathrm{hs}(q) = 1 - \sqrt{(1-q)/q}$ and $\mathrm{bias}(q) = 1 - 2(1-q)$. The desired inequality clearly holds at $q = 0$ and $q = 1$. For $q \in (0,1)$, it suffices to show that $4(1-q)^2 \le (1-q)/q$, or equivalently $4q(1-q) \le 1 \Leftrightarrow 1 - 4q + 4q^2 \ge 0 \Leftrightarrow (1-2q)^2 \ge 0$, which also clearly holds. □

Finally, the last main property of hs that we exploit is that hs scores and biases are quadratically related. To explain what we mean, start with the following definition of a general algorithm, where we take care not to put any restriction on the structure of the algorithm but want it to take inputs and return outputs while incurring some cost.

Definition 3.13. *Let S be a finite set, and let Δ be the set of probability distributions over S. A* general algorithm*, which we denote by R, is a pair of functions. The first function is from Δ to $[0,\infty]$, and we denote it by $\mathrm{cost}(R,\cdot)\colon \Delta \to \infty$, so that $\mathrm{cost}(R,\mu)$ returns a value in $[0,\infty]$ for $\mu \in \Delta$. The second function takes inputs from S and returns a random variable supported on $\{0,1\}$, and we denote it by $\mathrm{output}(R,\cdot)$, so that $\mathrm{output}(R,x)$ is a random variable on $\{0,1\}$ for each $x \in S$.*

The bias *of a general algorithm R on input $x \in S$ with respect to function $f\colon S \to \{0,1\}$ is* $\mathrm{bias}_f(R,x) \coloneqq 1 - 2\Pr[\mathrm{output}(R,x) \ne f(x)]$.

We note that if $\mathrm{output}(R,x)$ has distribution $\mathrm{Bernoulli}(q)$, then $\mathrm{bias}_f(R,x) = \mathrm{bias}_{f(x)}(q)$, where the function $\mathrm{bias}_{f(x)}(q)$ is defined according to Definition 3.2 and Definition 3.7.

Just like we defined general algorithms, we also define forecasting algorithms, which output confidences in $[0,1]$ instead of values in $\{0,1\}$.

Definition 3.14. *Let S be a finite set and let Δ be the set of all probability distributions over S. A* forecasting algorithm*, which we also denote by R, is a pair of functions. The first function is $\mathrm{cost}(R,\cdot)\colon \Delta \to \infty$, just like a general algorithm. The second function takes inputs from S and returns a random variable supported on $[0,1]$, and we denote it by $\mathrm{pred}(R,\cdot)$, so that $\mathrm{pred}(R,x)$ is a random variable on $[0,1]$ for each $x \in S$.*

The score *of a forecasting algorithm R on input $x \in S$ with respect to function $f\colon S \to \{0,1\}$ and scoring rule s is $\mathrm{score}_{s,f}(R,x) \coloneqq \mathbb{E}[s_{f(x)}(\mathrm{pred}(R,x))]$. When the function f is clear by the context, for notational simplicity we often omit it and write $\mathrm{score}_s(R,x)$. Additionally, when $s = \mathrm{hs}$, we sometimes omit it and write simply $\mathrm{score}(R,x)$.*

The following lemma is key. It says that we can convert any algorithm which achieves bias γ into a forecasting algorithm which achieves expected score at least $\gamma^2/2$ under the hs scoring rule; further, this conversion only manipulates the output of the algorithm, meaning it can be applied without changing the cost. That is, to turn R into a forecasting algorithm, we only need to run R, get an output 0 or 1, and then erase the output and write $(1-\gamma)/2$ or $(1+\gamma)/2$, respectively.

Moreover, it is possible to convert backward as well! To turn a forecasting algorithm R into a normal randomized algorithm, run R, take the output $q \in [0,1]$, erase it and write down a sample from $\mathrm{Bernoulli}(q)$ instead. If the original forecasting algorithm achieved expected score η, the new algorithm will achieve bias at least η. In particular, this lemma tells us that the best expected score and the best bias that an algorithm can make (under any cost restriction) are always quadratically related.

Lemma 3.15 (Conversion between regular and forecasting algorithms)**.** *A general algorithm R achieving worst-case bias $\gamma > 0$ for a function f can be converted into a forecasting algorithm R' with worst-case score at least $1 - \sqrt{1-\gamma^2} \geq \gamma^2/2$ for f. This conversion is pointwise: it depends only on changing a sample from the random variable* $\mathrm{output}(R, x)$ *after receiving it, as well as on the value of the worst-case bias γ.*

Conversely, a forecasting algorithm R with worst-case score η can be converted into a general algorithm R' with worst-case bias at least η. This conversion is pointwise: it depends only on changing a sample from $\mathrm{pred}(R, x)$ *after receiving it (and not even on the value of η).*

Proof. Start with a general algorithm R with worst-case bias $\gamma > 0$. On input x, run R to receive a sample $b \in \{0, 1\}$ from $\mathrm{output}(R, x)$. Then output $\mathrm{pred}(R', x) = (1-\gamma)/2$ if $b = 0$ and output $\mathrm{pred}(R', x) = (1+\gamma)/2$ if $b = 1$. It is clear that this R' was constructed in a pointwise fashion out of R, depending only on a sample from $\mathrm{output}(R, x)$. Now, fix $x \in S$, and let $p \in [0, 1]$ be the probability that $\mathrm{output}(R, x)$ gives the right answer. Since R has worst-case bias γ, it has bias at least γ on x, so $p \geq (1+\gamma)/2$. The expected score of R' on x is then

$$\begin{aligned}
\mathrm{score}(R', x) &= p\,\mathrm{hs}((1+\gamma)/2) + (1-p)\,\mathrm{hs}((1-\gamma)/2) \\
&= p - p\sqrt{\frac{1-\gamma}{1+\gamma}} + (1-p) - (1-p)\sqrt{\frac{1+\gamma}{1-\gamma}} \\
&= 1 - \sqrt{\frac{1+\gamma}{1-\gamma}} + p\left(\sqrt{\frac{1+\gamma}{1-\gamma}} - \sqrt{\frac{1-\gamma}{1+\gamma}}\right) \\
&\geq 1 - \sqrt{\frac{1+\gamma}{1-\gamma}} + \frac{1+\gamma}{2}\left(\sqrt{\frac{1+\gamma}{1-\gamma}} - \sqrt{\frac{1-\gamma}{1+\gamma}}\right) \\
&= 1 - \left(1 - \frac{1+\gamma}{2}\right)\sqrt{\frac{1+\gamma}{1-\gamma}} - \frac{1}{2}\sqrt{1-\gamma^2} \\
&= 1 - \sqrt{1-\gamma^2}.
\end{aligned}$$

For the other direction, let R be a forecasting algorithm with worst-case score $\eta > 0$. On input x, run R to receive a sample $q \in [0, 1]$ from $\mathrm{pred}(R, x)$. Then output 1 with probability q and 0 with probability $1 - q$, i.e. $\mathrm{output}(R', x) \sim \mathrm{Bernoulli}(q)$. It is clear that this R' is constructed in a pointwise fashion out of R (without even a dependence on η). Now, fix $x \in S$. We know that $\eta \leq \mathrm{score}(R, x) = \mathbb{E}[\mathrm{hs}_{f(x)}(\mathrm{pred}(R, x))]$. Now, we note that $\mathrm{hs}_{f(x)}(p) \leq \mathrm{bias}_{f(x)}(p)$ by Lemma 3.12. Thus we get $\eta \leq \mathbb{E}[\mathrm{bias}_{f(x)}(\mathrm{pred}(R, x))] = \mathrm{bias}(R', x)$, as desired. □

To demonstrate the power of these lemmas, observe that they imply a well-known amplification theorem for randomized algorithms, as we show in the lemma below. Note that this lemma does not refer to scoring rules or forecasting algorithms at all; those only appear as proof techniques.

Lemma 3.16 (informal)**.** *A randomized algorithm with bias γ can be amplified to bias $1/2$ by repeating it $2/\gamma^2$ times.*

Proof. Start with an algorithm making bias γ. Using Lemma 3.15, get a forecasting algorithm with expected score at least $1 - \sqrt{1-\gamma^2}$. Using Lemma 3.10, repeating the algorithm k times increases the expected score on each input x to at least $1-(1-\gamma^2)^{k/2}$. Using Lemma 3.15, we get an algorithm with worst-case bias at least $1 - (1-\gamma^2)^{k/2}$. Using Lemma 3.11, this is at least $\min\{k\gamma^2/4, 1/2\}$. Picking $k \geq 2/\gamma^2$, we get an algorithm with worst-case bias at least $1/2$ using only k repetitions of the original algorithm, as desired. □

4 Randomized query and communication complexity

To prove a strong minimax theorem for randomized query complexity, we start by formally defining forecasting algorithms in the query complexity setting. We will need these forecasting algorithms as a tool, despite our final statement not referring to them.

Definition 4.1. *A* deterministic forecasting decision tree *(on $n \in \mathbb{N}$ bits, with finite alphabet Σ) is a rooted tree on n bits whose internal vertices are labeled by $[n]$, where each internal vertex has $|\Sigma|$ children labeled by Σ, and where the leaves are labeled by $[0,1]$.*

A randomized forecasting decision tree *(on $n \in \mathbb{N}$ bits, with finite alphabet Σ) is a probability distribution over finitely many deterministic forecasting decision trees.*

We interpret a randomized forecasting decision tree as a forecasting algorithm in the intuitive way, where $\text{cost}(R, x)$ is the expected height of R on x (the expected height of the leaf of x in a deterministic forecasting tree sampled from the distribution R), and where $\text{pred}(R, x)$ is the random variable which samples from the leaf label when a random deterministic tree from R is run on x. Note that since we restrict to distributions with finite support, we do not need to invoke measure theory or integrals in interpreting these probabilities and expectations, even though there are uncountably many deterministic forecasting decision trees.

We extend $\text{cost}(R, \cdot)$ to the set Δ of probability distributions over S by writing $\text{cost}(R, \mu) = \mathbb{E}_{x \leftarrow \mu}[\text{cost}(R, x)]$, and similarly for $\text{score}(R, \mu) = \mathbb{E}_{x \leftarrow \mu}[\text{score}(R, x)]$. We now show a minimax theorem for the ratio of cost to score$^+$ for forecasting randomized algorithms. This minimax theorem will form the base of our final result: we will convert the left-hand side to $\mathrm{R}(f)$, and convert the right hand side to some desirable properties of a hard distribution μ.

Theorem 4.2. *Let $n \in \mathbb{N}$, let Σ be a finite alphabet, let $S \subseteq \Sigma^n$, and let $f \colon S \to \{0,1\}$. Let $\mathcal{R}$ be the set of all randomized forecasting decision trees on n bits with alphabet Σ. Let Δ be the set of probability distributions over S. Then*

$$\inf_{R \in \mathcal{R}} \max_{x \in S} \frac{\text{cost}(R, x)}{\text{score}(R, x)^+} = \max_{\mu \in \Delta} \inf_{R \in \mathcal{R}} \frac{\text{cost}(R, \mu)}{\text{score}(R, \mu)^+},$$

and the maximums are attained.

Proof. We use Theorem 2.18. All we need to do is verify that the conditions of the theorem hold. Our first task will be to deal with the strange set $\mathcal{R}$; we wish to turn it into a convex subset of a real topological vector space. To do so, we define the vector $v_R \in \mathbb{R}^{2|S|}$ for each $R \in \mathcal{R}$ by $v_R[x, 1] = \text{cost}(R, x)$ and $v_R[x, 2] = \text{score}(R, x)$, and consider the set $V = \{\, v_R : R \in \mathcal{R} \,\}$. For a vector $v \in V$, we define $\text{cost}(v, x) = v[x, 1]$ and $\text{score}(v, x) = v[x, 2]$, and we extend these definitions to $\text{cost}(v, \mu)$ and $\text{score}(v, \mu)$ by taking expectations over μ. Then it is clear that optimizing some function of $\text{cost}(R, \mu)$ and $\text{score}(R, \mu)$ over $\mathcal{R}$ is the same as optimizing the corresponding function of $\text{cost}(v, \mu)$ and $\text{score}(v, \mu)$ over V. Hence it suffices to show that

$$\inf_{v \in V} \max_{x \in S} \frac{\text{cost}(v, x)}{\text{score}(v, x)^+} = \max_{\mu \in \Delta} \inf_{v \in V} \frac{\text{cost}(v, \mu)}{\text{score}(v, \mu)^+},$$

with the maximums attained.

To do so, we first note that $V \subseteq \mathrm{R}^{2|S|}$ is convex. This is because if $v_1, v_2 \in V$ and $\lambda \in (0, 1)$, we know there are algorithms $R_1, R_2 \in \mathcal{R}$ such that $v_1 = v_{R_1}$ and $v_2 = v_{R_2}$, and then the algorithm $\lambda R_1 + (1 - \lambda) R_2$ (which mixes the distributions R_1 and R_2 over deterministic forecasting decision trees) is a valid member of $\mathcal{R}$. Then we have $v_{\lambda R_1 + (1-\lambda) R_2}[x, 1] = \text{cost}(\lambda R_1 + (1 - \lambda) R_2, x) =$

$\lambda \operatorname{cost}(R_1, x) + (1-\lambda)\operatorname{cost}(R_2, x) = \lambda v_{R_1}[x, 1] + (1-\lambda)v_{R_2}[x, 2]$, and similarly $v_{\lambda R_1 + (1-\lambda)R_2}[x, 2] = \lambda v_{R_1}[x, 2] + (1-\lambda)v_{R_2}[x, 2]$, so $v_{\lambda R_1 + (1-\lambda)R_2} = \lambda v_{R_1} + (1-\lambda)v_{R_2}$.

Next, we note that $\operatorname{cost}(v, \cdot)$ and $\operatorname{score}(v, \cdot)$ are linear functions of μ; this is because they are defined as expectations over μ. Further, observe that $\operatorname{cost}(\cdot, \mu)$ and $\operatorname{score}(\cdot, \mu)$ are linear in v. It is also clear that $\operatorname{cost}(v, \mu)$ and $\operatorname{score}(v, \mu)$ are continuous in both v and μ.

It remains to check that cost and score are well-behaved. First, note that there is always an algorithm which queries all the bits and outputs the right answer $f(x)$ with perfect confidence. Such an algorithm R has $\operatorname{cost}(v_R, \mu) = n$ and $\operatorname{score}(v_R, \mu) = 1$ for all μ, so finite costs and scores are attainable. Next, note that if R is such that $\operatorname{cost}(v_R, \mu) = 0$ for any μ, then R must make no queries when run on μ. This means R makes no queries when run on any input, so $\operatorname{cost}(v_R, \mu') = 0$ for all $\mu' \in \Delta$. Finally, note that $\operatorname{cost}(\cdot, \mu)$ is linear for each μ, so if $\operatorname{cost}(v, \mu) = 0$ and $\operatorname{cost}(v', \mu) > 0$, we necessarily have $\operatorname{cost}(\lambda v + (1-\lambda)v', \mu) > 0$ for $\lambda \in (0, 1)$. Hence all the conditions of Theorem 2.18 are satisfied, and the desired result follows. □

Our next task is to relate the left-hand side of the equation in the last theorem to $\mathrm{R}(f)$.

Theorem 4.3. *Using the notation of Theorem 4.2, we have*

$$\inf_{R \in \mathcal{R}} \max_{x \in S} \frac{\operatorname{cost}(R, x)}{\operatorname{score}(R, x)^+} \geq \frac{\mathrm{R}(f)}{240}.$$

To prove this theorem, the idea is to take R from the left-hand side, amplify the score of R up to a constant (using the fact that score amplifies linearly), and then convert the constant score to constant bias (and hence constant error), getting an upper bound on $\mathrm{R}(f)$. This is slightly tricky, because the amount we need to amplify by may depend on the input x; for some x, both $\operatorname{cost}(R, x)$ and $\operatorname{score}(R, x)$ may be small, while for other x they are both large. Unfortunately, we do not have access to $\operatorname{score}(R, x)$ when we receive input x. Instead, in order to amplify by approximately the correct amount, we estimate $\operatorname{cost}(R, x)$ (by repeatedly running R on x and observing the number of queries), and we use this cost estimate to decide the amount of amplification needed.

Proof. Let Y^* be the optimal value of the left-hand side, and let R be an algorithm such that $\max_{x \in S} \operatorname{cost}(R, x)/\operatorname{score}(R, x)^+ = Y$, where Y is arbitrarily close to Y^* (and $Y \geq Y^*$). Then in particular, $\operatorname{score}(R, x) > 0$ for all $x \in S$, and for each $x \in S$ we have $\operatorname{cost}(R, x)/\operatorname{score}(R, x) \leq Y$. Let R' be a modification of R where we cut off each decision tree in the support of R after $2Y$ queries, and return $1/2$ in case of a cutoff (ensuring we get a score of 0 for that branch). Note that by Markov's inequality, the probability of encountering a cutoff branch on input x to R' is at most $\operatorname{cost}(R, x)/2Y \leq Y \operatorname{score}(R, x)/2Y = \operatorname{score}(R, x)/2$. Since each non-cut-off leaf can contribute at most 1 to the score (as the maximum of $\mathrm{hs}(\cdot)$ is 1), and since the score at a cutoff is 0, the decrease in score when going from R to R' is at most the probability of encountering a cutoff. It follows that $\operatorname{score}(R', x) \geq \operatorname{score}(R, x) - \operatorname{score}(R, x)/2 = \operatorname{score}(R, x)/2$ for all $x \in S$.

Next, we describe a randomized forecasting algorithm R''. The algorithm R'' runs R' on x until the number of queries made reaches $10Y$. Let L be the number of runs of R' on x it takes to reach $10Y$ queries. Then R'' runs R' on x an additional L times, and uses those new runs to amplify the score, achieving score $1 - (1 - \operatorname{score}(R', x))^L$. We wish to prove this score is at least a constant and that the total number of queries is only $O(Y)$.

First, we bound the expectation of L, the random variable for the number of runs of R' on x it takes to reach $10Y$ queries. Let X_i be i.i.d. random variables each representing the number of queries in a single run of R' on x (so each X_i is supported on $\{0, 1, \ldots, 2Y\}$). Consider the total number of queries made until the cutoff is reached; this is $\sum_{i=1}^{L} X_i$. Let I_i be the Boolean random

variable which is 0 if $L < i$ and 1 if $L \geq i$. Then $\sum_{i=1}^{L} X_i = \sum_{i=1}^{\infty} X_i I_i$. Note that the value of $\sum_{i=1}^{L} X_i$ is always at most $10Y + 2Y$, because after the threshold $10Y$ is reached, less than one full run of R' on x will happen (using at most $2Y$ queries). Hence[3]

$$\begin{aligned}
12Y > \mathbb{E}\left[\sum_{i=1}^{L} X_i\right] &= \mathbb{E}\left[\sum_{i=1}^{\infty} X_i I_i\right] = \sum_{i=1}^{\infty} \mathbb{E}\left[X_i I_i\right] \\
&= \sum_{i=1}^{\infty} \Pr[I_i = 0]\mathbb{E}[X_i I_i | I_i = 0] + \Pr[I_i = 1]\mathbb{E}[X_i I_i | I_i = 1] \\
&= \sum_{i=1}^{\infty} \Pr[L \geq i]\mathbb{E}[X_i] \\
&= \operatorname{cost}(R', x)\mathbb{E}[L].
\end{aligned}$$

It follows that $\mathbb{E}[L] < 12Y/\operatorname{cost}(R', x)$. This means the total expected number of queries R'' makes is at most $12Y$ for getting the estimate L, plus $\operatorname{cost}(R', x) \cdot \mathbb{E}[L] < 12Y$ for amplifying the score, for a total of fewer than $24Y$ expected queries.

To bound the expected score, we start by ensuring L is not too small except with small probability. Note that for a constant T, we have $\Pr[L \leq T] = \Pr[\sum_{i=1}^{T} X_i \geq bY]$. The sum $\sum_{i=1}^{T} X_i$ has expected value $T \operatorname{cost}(R', x)$ and has variance T times the variance of one X_i. Since X_i is non-negative and bounded above by $2Y$, its variance is bounded above by $\operatorname{Var}[X_i] \leq \mathbb{E}[X_i^2] \leq 2Y\mathbb{E}[X_i] = 2Y \operatorname{cost}(R', x)$. Hence, the variance of the sum is at most $2TY \operatorname{cost}(R', x)$. We use Chebyshev's inequality, writing

$$\begin{aligned}
\Pr[L \leq T] &= \Pr\left[\sum_{i=1}^{T} X_i \geq 10\,\mathrm{R}(f)\right] \\
&= \Pr\left[\sum_{i=1}^{T} X_i - T\operatorname{cost}(R', x) \geq 10Y - T\operatorname{cost}(R', x)\right] \\
&\leq \frac{2TY\operatorname{cost}(R', x)}{(10Y - T\operatorname{cost}(R', x))^2},
\end{aligned}$$

which holds assuming $T \leq 10Y/\operatorname{cost}(R', x)$. In particular, if $T = 2Y/\operatorname{cost}(R', x)$, then $\Pr[L \leq T] \leq 1/16$.

Now, note that conditioned on $L = \ell$, the expected score in the second round of R'' is at least $1 - (1 - \operatorname{score}(R', x))^{\ell}$. This is increasing in ℓ; hence, conditioned on $L > T$, the expected score of R'' on x is greater than $1 - (1 - \operatorname{score}(R', x))^{T}$. Conditioned on $L \leq T$, we still have the expected score be at least 0, since it is at least 0 for every fixed ℓ. Hence the final expected score of R'' on x is greater than $(1 - (1 - \operatorname{score}(R', x))^{T})(1 - \Pr[L \leq T]) \geq 1 - (1 - \operatorname{score}(R', x))^{T} - \Pr[L \leq T]$.

[3]The equality $\mathbb{E}\left[\sum_{i=1}^{L} X_i\right] = \mathbb{E}[X_1]\mathbb{E}[L]$, which we rederive here, is known as Wald's equation.

Picking $T = 2Y/\operatorname{cost}(R', x)$, we get

$$\begin{aligned}\operatorname{score}(R'', x) &> 1 - (1 - \operatorname{score}(R', x))^{2Y/\operatorname{cost}(R', x)} - 1/16 \\ &\geq \frac{1}{2} \min\left\{1, 2Y \frac{\operatorname{score}(R', x)}{\operatorname{cost}(R', x)}\right\} - 1/16 \\ &\geq \frac{1}{2} \min\left\{1, \frac{\operatorname{score}(R, x) Y}{\operatorname{cost}(R, x)}\right\} - 1/16 \\ &\geq \frac{1}{2} - \frac{1}{16} = \frac{7}{16}.\end{aligned}$$

This algorithm R'' makes fewer than $24Y$ expected queries. We cut if off after $240Y$ queries, outputting prediction $1/2$ (getting score 0) in case of a cutoff; this gives an algorithm R''' whose worst-case number of queries is $240Y$, and whose expected score on each $x \in D$ is at least $7/16 - 1/10 \geq 1/3$. Using Lemma 3.15, we can view R''' as a randomized algorithm computing $f(x)$ with worst-case bias at least $1/3$, and hence worst-case error at most $1/3$. This means that $\mathrm{R}(f) \leq 240Y$. Since we can pick Y arbitrarily close to Y^*, we also get that $\mathrm{R}(f)$ is at most the infimum of $240Y$ over feasible choices of Y, which is $240Y^*$, and the desired result follows. □

Our next task is to show that the max-inf side of Theorem 4.2 gives us a distribution μ against which it is hard to tell apart 0-inputs from 1-inputs, in terms of the achievable squared-Hellinger distance between the distributions of the transcript on the 0- and 1-inputs. The following lemma will come in useful. We prove it in Appendix B.

Lemma 4.4 (Hellinger distance of disjoint mixtures)**.** *Let μ be a distribution over a finite support A, and for each $a \in A$, let ν_0^a and ν_1^a be two distributions over a finite support S_a. Let ν_0^μ and ν_1^μ denote the mixture distributions where $a \leftarrow \mu$ is sampled, and then a sample is produced from ν_0^a or ν_1^a respectively. Assume the sets S_a are disjoint for all $a \in A$. Then*

$$\mathrm{h}^2(\nu_0^\mu, \nu_1^\mu) = \mathbb{E}_{a \leftarrow \mu}[\mathrm{h}^2(\nu_0^a, \nu_1^a)].$$

Theorem 4.5. *Let $n \in \mathbb{N}$, let Σ be a finite alphabet, let $S \subseteq \Sigma^n$, and let $f \colon S \to \{0, 1\}$ be a non-constant function. Then there exist distributions μ_0 on $f^{-1}(0)$ and μ_1 on $f^{-1}(1)$ such that for all randomized query algorithms R,*

$$\frac{\operatorname{cost}(R, \mu)}{\mathrm{h}^2(\operatorname{tran}(R, \mu_0), \operatorname{tran}(R, \mu_1))} \geq \frac{\mathrm{R}(f)}{240}.$$

Here $\mu = (\mu_0 + \mu_1)/2$, and we interpret $r/0 = \infty$ for $r \in [0, \infty)$.

Proof. Using Theorem 4.2 and Theorem 4.3, we get a distribution μ on S such that for all randomized forecasting algorithms R, we have $\operatorname{cost}(R, \mu)/\operatorname{score}(R, \mu)^+ \geq \mathrm{R}(f)/240$. Note that it must be the case that an algorithm R which makes no queries must have $\operatorname{score}(R, \mu) \leq 0$; this is because we have $\mathrm{R}(f) \geq 1$ (since f is non-constant), and if there was an algorithm with cost 0 achieving positive score, we'd have $\operatorname{cost}(R, \mu)/\operatorname{score}(R, \mu)^+ = 0$, giving a contradiction. Therefore, it must be the case that μ places equal weight on 0 and 1 inputs, because otherwise a 0-cost algorithm could indeed predict $f(x)$ with positive bias (and hence positive score by Lemma 3.15) against μ. We set μ_0 to be the conditional distribution of μ on the 0-inputs of f, and set μ_1 to be the conditional distribution of μ on the 1-inputs of f.

Next, we simplify the expression

$$\inf_R \frac{\operatorname{cost}(R, \mu)}{\mathrm{h}^2(\operatorname{tran}(R, \mu_0), \operatorname{tran}(R, \mu_1))}.$$

Note that both the numenator and the denominator do not depend on the leaf labels, only on the queries of the randomized decision trees. We can therefore view the set of all randomized query algorithms R as the convex hull of the set of all deterministic decision trees with no leaf labels. Now, note that $\mathrm{cost}(R,\mu)$ and $\mathrm{h}^2(\mathrm{tran}(R,\mu_0),\mathrm{tran}(R,\mu_1))$ are both linear functions of (the probability vector of) R; for the latter, this is due to Lemma 4.4. Then by Lemma 2.12, the ratio is quasiconcave in R, and by Lemma 2.10, the infimum of this ratio over randomized query algorithms R is equal to the minimum over deterministic query algorithms A. Therefore, it suffices to show that for each deterministic query algorithm A making a non-zero number of queries, we have $\mathrm{cost}(A,\mu)/\mathrm{h}^2(\mathrm{tran}(A,\mu_0),\mathrm{tran}(A,\mu_1)) \geq \mathrm{R}(f)/240$.

Fix such A. We assume its leaves are not labeled. By Lemma 3.9, we can label the leaves of A such that $\mathrm{score}(A,\mu) = \mathrm{h}^2(\mathrm{tran}(A,\mu_0),\mathrm{tran}(A,\mu_1))$. This labeling does not affect the cost. Then

$$\frac{\mathrm{cost}(A,\mu)}{\mathrm{h}^2(\mathrm{tran}(A,\mu_0),\mathrm{tran}(A,\mu_1))} = \frac{\mathrm{cost}(A,\mu)}{\mathrm{score}(A,\mu)^+} \geq \frac{\mathrm{R}(f)}{240},$$

as desired. □

Finally, we strengthen this to a lower bound for the *minimum* of $\mathrm{cost}(R,\mu_0)$ and $\mathrm{cost}(R,\mu_1)$, instead of for their average $\mathrm{cost}(R,\mu)$.

Theorem 4.6. *Let $n \in \mathbb{N}$, let Σ be a finite alphabet, let $S \subseteq \Sigma^n$, and let $f\colon S \to \{0,1\}$ be a non-constant function. Then there exist distributions μ_0 on $f^{-1}(0)$ and μ_1 on $f^{-1}(1)$ such that for all randomized query algorithms R,*

$$\frac{\min\{\mathrm{cost}(R,\mu_0),\mathrm{cost}(R,\mu_1)\}}{\mathrm{h}^2(\mathrm{tran}(R,\mu_0),\mathrm{tran}(R,\mu_1))} \geq \frac{\mathrm{R}(f)}{3000},$$

where we interpret $r/0 = \infty$ for $r \in [0,\infty)$.

Proof. We use μ_0 and μ_1 from Theorem 4.5. Note that

$$\inf_R \frac{\min\{\mathrm{cost}(R,\mu_0),\mathrm{cost}(R,\mu_1)\}}{\mathrm{h}^2(\mathrm{tran}(R,\mu_0),\mathrm{tran}(R,\mu_1))} = \inf_{R,b\in\{0,1\}} \frac{\mathrm{cost}(R,\mu_b)}{\mathrm{h}^2(\mathrm{tran}(R,\mu_0),\mathrm{tran}(R,\mu_1))}$$

$$= \min_{b\in\{0,1\}} \inf_R \frac{\mathrm{cost}(R,\mu_b)}{\mathrm{h}^2(\mathrm{tran}(R,\mu_0),\mathrm{tran}(R,\mu_1))}.$$

By the same argument as in the proof of Theorem 4.5, this last infimum over R is equal to the infimum over deterministic unlabeled decision trees D with height at least 1.

Let D be such an algorithm. By Theorem 4.5, it suffices to show that

$$\frac{\min\{\mathrm{cost}(D,\mu_0),\mathrm{cost}(D,\mu_1)\}}{\mathrm{h}^2(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))} \geq (1/c)\min_{D'} \frac{\mathrm{cost}(D',\mu)}{\mathrm{h}^2(\mathrm{tran}(D',\mu_0),\mathrm{tran}(D',\mu_1))},$$

where $\mu = (\mu_0+\mu_1)/2$. By Lemma 3.9, we can label the leaves of D so that we have the property $\mathrm{h}^2(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1)) = \mathrm{score}(D,\mu)$, and similarly for D'. The desired inequality is trivial when $\mathrm{score}(D,\mu) = 0$ (since the ratio is then ∞), so suppose $\mathrm{score}(D,\mu) > 0$. We wish to show

$$\frac{\min\{\mathrm{cost}(D,\mu_0),\mathrm{cost}(D,\mu_1)\}}{\mathrm{score}(D,\mu)} \geq (1/c)\min_{D'} \frac{\mathrm{cost}(D',\mu)}{\mathrm{score}(D',\mu)}.$$

In other words, we wish to show that there exists a deterministic forecasting algorithm D' such that $\mathrm{cost}(D',\mu)/\mathrm{score}(D',\mu) \leq c\,\mathrm{cost}(D,\mu_b)/\mathrm{score}(D,\mu)$, regardless of whether $b=0$ or $b=1$.

We construct D' such that $\mathrm{cost}(D',\mu)/\operatorname{score}(D',\mu) \le c\,\mathrm{cost}(D,\mu_b)/\operatorname{score}(D,\mu)$. The idea is to start with D, and then cut off the branches that are much more likely under μ_{1-b} than under μ_b. That is, for a vertex v of D, let $\mu_0[v]$ denote the probability that v is reached when D is run on an input from μ_b, and define $\mu_{1-b}[v]$ similarly. Recall that the leaves of D are labeled according to the strategy that achieves $\operatorname{score}(D,\mu) = \mathrm{h}^2(\mathrm{tran}(D,\mu_0),\mathrm{tran}(D,\mu_1))$, which, by Lemma 3.9, is such that at a leaf v, the algorithm D outputs $\mu_1[v]/2\mu[v]$.

Pick a constant $a \in (1/2,1)$, and let D' be the algorithm which cuts off D the first time it enters a vertex for which $\mu_{1-b}[v]/2\mu[v] \ge a$, and outputs a (if $b=0$) or $1-a$ (if $b=1$) instead of continuing to run D. Let V be the set of all vertices which cause such a cutoff; note that no vertex in V is a descendant of another vertex in V. For $v \in V$, let μ^v be the distribution μ conditioned on reaching v, and similarly define μ_0^v and μ_1^v. Let μ^* be the distribution μ conditioned on reaching none of the vertices in V, and similarly define μ_0^* and μ_1^*. Since we are dealing with a deterministic decision tree, all the distributions μ_0^v and μ_1^v have disjoint supports for all the different $v \in V$, and they're also disjoint from μ_0^* and μ_1^*; indeed, μ is a disjoint mixture of all different distributions. It follows that $\operatorname{score}(D,\mu)$ is a mixture of terms $\operatorname{score}(D,\mu^v)$ and of $\operatorname{score}(D,\mu^*)$. The score $\operatorname{score}(D',\mu)$ of the algorithm D' is also such a mixture.

Now, note that $\operatorname{score}(D,\mu^v) \le 1$, and that $\operatorname{score}(D',\mu^v) = \mathbb{E}_{x\leftarrow\mu^v}[\mathrm{hs}_{f(x)}(a)]$ if $b=0$ and $\operatorname{score}(D',\mu^v) = \mathbb{E}_{x\leftarrow\mu^v}[\mathrm{hs}_{f(x)}(1-a)]$ if $b=1$. This means

$$\operatorname{score}(D',\mu^v) = \frac{\mu_b[v]}{2\mu[v]}\,\mathrm{hs}(1-a) + \frac{\mu_{1-b}[v]}{2\mu[v]}\,\mathrm{hs}(a) = (1-p)\,\mathrm{hs}(1-a) + p\,\mathrm{hs}(a)$$
$$= 1 - (1-p)\sqrt{a/(1-a)} - p\sqrt{(1-a)/a},$$

where $p = \mu_{1-b}[v]/2\mu[v] \ge a$. Since $a > 1/2$, this is increasing in p, so we have $\operatorname{score}(D',\mu^v) \ge 1 - 2\sqrt{a(1-a)}$, and hence $\operatorname{score}(D',\mu^v) \ge (1-2\sqrt{a(1-a)})\operatorname{score}(D,\mu^v)$. It also holds that $\operatorname{score}(D',\mu^*) = \operatorname{score}(D,\mu^*) \ge (1-2\sqrt{a(1-a)})\operatorname{score}(D,\mu^*)$. Since $\operatorname{score}(D,\mu)$ and $\operatorname{score}(D',\mu)$ are matching mixtures of $\operatorname{score}(D,\mu^v)$ and $\operatorname{score}(D',\mu^v)$ respectively, it follows that $\operatorname{score}(D',\mu) \ge (1-2\sqrt{a(1-a)}\operatorname{score}(D,\mu)$.

We now analyze the cost of D'. Note that $\mathrm{cost}(D',\mu) = (1/2)\,\mathrm{cost}(D',\mu_b) + (1/2)\,\mathrm{cost}(D',\mu_{1-b})$; we clearly have $\mathrm{cost}(D',\mu_b) \le \mathrm{cost}(D,\mu_b)$, so it suffices to upper bound $\mathrm{cost}(D',\mu_{1-b})$. This is the expected height of a leaf D' reaches when run on μ_{1-b}, which is a mixture of $\mathrm{cost}(D',\mu_{1-b}^*)$ and $\mathrm{cost}(D',\mu_{1-b}^v)$. Now, note that a leaf u reached by $\mathrm{cost}(D',\mu_{1-b}^*)$ must have $\mu_{1-b}[u]/2\mu[u] < a$, or $\mu_b[u] < (1-a)/a \cdot \mu_{1-b}[u]$. It follows that

$$\mathrm{cost}(D',\mu_{1-b}^*) \le (1-a)/a \cdot \mathrm{cost}(D',\mu_b^*) = (1-a)/a \cdot \mathrm{cost}(D,\mu_b^*).$$

Similarly, for each $v \in V$, the parent u of v satisfies $\mu_{1-b}[u]/2\mu[u] < a$, meaning that $\mu_b[u] > (1-a)/a \cdot \mu_{1-b}[u]$; note that since this parent u of v is not a leaf, conditioned on reaching u the height of the path will always be at least the height of v (one more than the height of u); since $\mathrm{cost}(B,\mu_{1-b}^v)$ is exactly the height of v, we necessarily have

$$\mathrm{cost}(D,\mu_b^v) \ge \mathrm{cost}(D',\mu_b^v) \ge (1-a)/a \cdot \mathrm{cost}(D',\mu_{1-b}^v).$$

We conclude that $\mathrm{cost}(D',\mu_{1-b}) \le \frac{a}{1-a}\,\mathrm{cost}(D,\mu_b)$, and hence

$$\mathrm{cost}(D',\mu) \le \left(\frac{1}{2} + \frac{a}{2(1-a)}\right)\mathrm{cost}(D,\mu_b) = \frac{\mathrm{cost}(D,\mu_b)}{2(1-a)}.$$

We therefore have

$$\frac{\mathrm{cost}(D',\mu)}{\operatorname{score}(D',\mu)} \le \frac{1}{2(1-a)(1-2\sqrt{a(1-a)})}\,\frac{\mathrm{cost}(D,\mu_b)}{\operatorname{score}(D,\mu)}.$$

Finally, optimizing a, we pick $a = (2+\sqrt{2})/4$ to get

$$\frac{\operatorname{cost}(D',\mu)}{\operatorname{score}(D',\mu)} \leq (6+4\sqrt{2})\frac{\operatorname{cost}(D,\mu_b)}{\operatorname{score}(D,\mu)},$$

from which the desired result follows. □

Corollary 4.7. *Let $n \in \mathbb{N}$, let Σ be a finite alphabet, let $S \subseteq \Sigma^n$, and let $f\colon S \to \{0,1\}$ be a function. Then there exists a distribution μ on S such that for all $\gamma \in [0,1]$,*

$$\overline{\mathrm{R}}^{\mu}_{\dot{\gamma}}(f) \geq \frac{\gamma^2\,\mathrm{R}(f)}{500}.$$

Here $\dot{\gamma} = (1-\gamma)/2$ and $\overline{\mathrm{R}}^{\mu}_{\epsilon}(f)$ denotes the average cost (against μ) of a randomized algorithm achieving error at most ϵ (against μ) for solving f.

Proof. If f is constant, then $\mathrm{R}(f) = 0$ and the desired bound trivially follows. Therefore, assume f is not constant. We use the distribution μ from Theorem 4.5. Let R be a randomized algorithm which achieves bias γ against μ. Then using Lemma 3.15, we can convert R into a forecasting algorithm R' which achieves expected score $1-\sqrt{1-\gamma^2} \geq \gamma^2/2$ against μ, and has the same distribution over query trees (that is, only the leaves changed). Now, by the property of μ, we know that

$$\frac{\operatorname{cost}(R',\mu)}{\operatorname{score}(R',\mu)} \geq \frac{\mathrm{R}(f)}{240},$$

where we used Lemma 3.9 to get a result for score instead of Hellinger distance in the denominator, and where we used the fact that R achieves non-zero bias against μ (despite μ being balanced between 0- and 1-inputs) to conclude that R does not make 0 queries. Using $\operatorname{score}(R',\mu) \geq \gamma^2/2$ and $\operatorname{cost}(R,\mu) = \operatorname{cost}(R',\mu)$, we get $2\operatorname{cost}(R,\mu)/\gamma^2 \geq \mathrm{R}(f)/240$, or $\operatorname{cost}(R,\mu) \geq \gamma^2\,\mathrm{R}(f)/480$, as desired. □

4.1 Communication complexity

Theorem 1.8 (Restated)**.** For any non-constant partial function $F\colon \mathcal{X}\times\mathcal{Y} \to \{0,1\}$ over finite sets $\mathcal{X}$ and $\mathcal{Y}$, there is a pair of distributions μ_0 on $F^{-1}(0)$ and μ_1 on $F^{-1}(1)$ such that for any public-randomness communication protocol Π, the squared Hellinger distance between the distribution of its transcripts on μ_0 and μ_1 is bounded above by

$$\mathrm{h}^2\left(\operatorname{tran}(\Pi,\mu_0), \operatorname{tran}(\Pi,\mu_1)\right) = O\left(\frac{\min\{\operatorname{cost}(\Pi,\mu_0), \operatorname{cost}(\Pi,\mu_1)\}}{\mathrm{RCC}(F)}\right).$$

Proof. This theorem follows directly from Theorem 4.6 once we realize that a communication function can be interpreted as a query function. That is, we take F and convert it into a query function f as follows. The input to f will contain one bit for each possible function of $\mathcal{X}$ (that Alice might send to Bob), and one bit for each possible function of $\mathcal{Y}$ (that Bob might send to Alice), for a total input length of $n = 2^{|\mathcal{X}|} + 2^{|\mathcal{Y}|}$. The inputs to f will be the strings in $\{0,1\}^n$ which are generated by a pair $(x,y) \in S$, that is, the strings $z \in \{0,1\}^n$ for which there exists a pair $(x,y) \in S$ such that z_k is the result of applying the k-th possible function to x (if $k \leq 2^{|\mathcal{X}|}$) or the $(k-2^{|\mathcal{X}|})$-th possible function to y (if $k > 2^{|\mathcal{X}|}$). Then f is a Boolean function of domain of size $|S|$, with each string in its domain corresponding to a string in S.

We note that $\mathrm{RDT}(f) = \mathrm{RCC}(F)$. This is clear from the definition of $\mathrm{RCC}(F)$: the public-coin randomness essentially means that Alice and Bob agree on a randomized decision tree in advance,

including on who speaks when (as a function of the transcript), which is equivalent to agreeing in a decision tree for f in advance. The transcript of f on an input is precisely the transcript of F on the corresponding input, with the catch that in query complexity we defined the transcript to include the deterministic decision tree by the protocol; hence, the query version of a transcript of f actually corresponds to (R, Π) for F, where R is the public randomness and Π is the usual communication complexity transcript. The desired result then follows immediately from applying Theorem 4.6 to f. □

Corollary 4.8. *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets, let $S \subseteq \mathcal{X} \times \mathcal{Y}$, and let $F\colon S \to \{0,1\}$ be a function. Then there exists a distribution μ on S such that for all $\gamma \in [0,1]$,*

$$\overline{\mathrm{RCC}}^{\mu}_{\gamma}(F) = \Omega(\gamma^2 \,\mathrm{RCC}(F)).$$

5 Quantum query and communication complexity

In contrast to the classical case, it is well-known that quantum algorithms can be amplified *linearly* in $1/\gamma$, where γ is the bias. Formally, we have the following theorem.

Theorem 5.1 (Amplitude estimation)**.** *Suppose we have access to a unitary U (representing a quantum algorithm) which maps $|0\rangle$ to $|\psi\rangle$, as well as access to a projective measurement Π, and we wish to estimate $p := \||\Pi|\psi\rangle\|_2^2$ (representing the probability the quantum algorithm accepts). Fix $\epsilon, \delta \in (0, 1/2)$. Then using at most $(100/\epsilon) \cdot \ln(1/\delta)$ controlled applications of U or $U^\dagger$ and at most that many applications of $I - 2\Pi$, we can output $\tilde{p} \in [0,1]$ such that $|\tilde{p} - p| \leq \epsilon$ with probability at least $1 - \delta$.*

This theorem follows from [BHMT02], as well as from the arguably simpler techniques in [AR20]. (In fact, these authors show something slightly stronger: amplitude estimation can be done with overhead $O(\sqrt{\epsilon + p} \cdot (1/\epsilon) \cdot \log 1/\delta)$. We refer the interested reader to Appendix C to see how this follows from [BHMT02].)

Given that quantum algorithms can be amplified linearly in the bias, it would seem that the desired minimax theorem follows easily from Theorem 2.18: simply apply a minimax to $\mathrm{cost}(Q,\mu)/\,\mathrm{bias}_f(Q,\mu)^+$, where Q is a quantum algorithm and μ is a distribution over the inputs. Then use the linear amplification result to argue that $\min_Q \max_\mu \mathrm{cost}(Q,\mu)/\,\mathrm{bias}_f(Q,\mu)^+$ is $\Theta(\mathrm{Q}(f))$. Sounds simple! (This works better than for randomized algorithms, because $\mathrm{bias}_f(\cdot,\cdot)$ is saddle while $\mathrm{bias}_f(\cdot,\cdot)^2$ is not.)

Unfortunately, there is an annoying hole in this argument: the function $\mathrm{cost}(Q,\mu)$ is not convex in Q. While it is not immediately clear what a convex combination of two quantum algorithms Q_1 and Q_2 should be, most intuitive definitions will have the convex combination use a number of unitaries that is equal to the *maximum* of the number used in Q_1 and Q_2, rather than the average.

To get around this, we switch the computational model from quantum algorithms to probability distributions over quantum algorithms. These probabilistic quantum algorithms have outputs and biases defined in the intuitive way, but their cost is defined as the *expected* cost of the underlying quantum algorithms, rather than the maximum cost. This ensures the function $\mathrm{cost}(\cdot,\cdot)$ will be saddle, and Theorem 2.18 can be applied. The trick then becomes showing that these probabilistic quantum algorithms can still be amplified linearly. This turns out to be true, up to logarithmic factors. Once amplified, constant-error probabilistic quantum algorithms can be converted into ordinary quantum algorithms, giving us a minimax theorem that can be applied to ordinary quantum algorithms as well.

5.1 Quantum query complexity

Our goal in this section will be to prove the following theorem.

Theorem 5.2. *For any Boolean-valued function f, there exists a distribution μ over $\mathrm{Dom}(f)$ such that for any $\gamma \in [0,1]$, we have $\mathrm{Q}^{\mu}_{\hat{\gamma}}(f) \geq \gamma \cdot \tilde{\Omega}(\mathrm{Q}(f))$. Here $\mathrm{Q}^{\mu}_{\hat{\gamma}}(f)$ denotes the minimum number of queries required by a quantum algorithm which achieves bias γ against μ for computing f. The constants in the $\tilde{\Omega}$ notation are universal.*

In fact, we will prove a stronger (and tighter) version in terms of *probabilistic* quantum algorithms. These are simply probability distributions over quantum algorithms of possibly different query costs; we define the cost of a probabilistic quantum algorithm as the expected cost of a quantum algorithm sampled from the probability distribution.

Definition 5.3. *A* probabilistic quantum algorithm *is a probability distribution P over quantum algorithms. For an input string x, we let $P(x)$ be the random variable that outputs a sample from $Q(x)$ where Q is a quantum algorithm sampled from P. The cost of P, denoted $|P|$, is the expected cost of a quantum algorithm sampled from P. The error of P on input x to a Boolean function f is defined as $\Pr_{Q \sim P}[Q(x) \neq f(x)]$.*

Definition 5.4. *Let f be a Boolean-valued function with $\mathrm{Dom}(f) \subseteq \Sigma^n$. We define $\mathrm{PQ}_{\hat{\gamma}}(f)$ to be the minimum cost $|P|$ of a probabilistic quantum algorithm P which computes f to worst-case bias γ.*

Theorem 5.5. *For any Boolean function f and any $\gamma \in (0, 1/3)$, we have $\mathrm{PQ}_{\hat{\gamma}}(f) = \widetilde{\Theta}(\gamma\, \mathrm{Q}(f))$. More explicitly,*

$$\mathrm{PQ}_{\hat{\gamma}}(f) = O(\gamma\, \mathrm{Q}(f))$$
$$\mathrm{PQ}_{\hat{\gamma}}(f) = \Omega\left(\frac{\gamma\, \mathrm{Q}(f)}{\log(1/\gamma) \log\log(1/\gamma)}\right).$$

Proof. For the upper bound, let Q be a quantum algorithm computing f to error $1/3$ using $\mathrm{Q}(f)$ queries. Let Q' be the probabilistic quantum algorithm which runs $\mathrm{Q}(f)$ with probability 3γ and otherwise uses no queries and guesses the output at random (with probability $1/2$ for outputting both 0 and 1). The probability of error of Q' is at most $(1/2)(1-3\gamma) + (1/3)(3\gamma) = (1/2)(1-\gamma)$, which means its bias is at least γ on every input. The expected number of queries Q' uses is $3\gamma\, \mathrm{Q}(f)$. Hence we have $\mathrm{PQ}_{\hat{\gamma}}(f) \leq 3\gamma\, \mathrm{Q}(f)$.

For the lower bound, we start with a probabilistic quantum algorithm P which achieves worst-case bias γ and has cost $|p| = \mathrm{PQ}_{\hat{\gamma}}(f)$, and make several modifications to it. First, we remove from the support of P all quantum algorithms which use more than $2|P|/\gamma$ queries, and we replace them with a 0-query quantum algorithm that guesses the answer at random (with $1/2$ probability on outputs 0 and 1). This gives us a probabilistic quantum algorithm P_1 which uses at most $2|P|/\gamma$ queries even in the worst case, and has $|P_1| \leq |P|$ and the worst-case bias of P_1 is at least $\gamma/2$ (since by Markov's inequality, the probability mass over the removed quantum algorithms was at most $\gamma/2$, and they could have had bias at most 1 which turned into bias 0, decreasing the overall bias by at most $\gamma/2$).

Next, we modify P_1 to get a probabilistic algorithm P_2 which always uses a number of queries which is a power of 2. This can be done simply by increasing the number of queries each algorithm in the support of P_1 makes (and ignoring the extra queries). This way, we have $|P_2| \leq 2|P_1| \leq 2|P|$, the largest number of queries P_2 can make is at most $4|P|/\gamma$, and the bias of P_2 is at least $\gamma/2$ on every input.

Further, we modify P_2 to get a probabilistic quantum algorithm P_3 which always uses at least $8|P|$ queries (but still only uses a number of queries which is a power of 2). This can be done by again increasing the number of queries a quantum algorithm in the support of P_2 makes, when necessary. This adds at most an additive $16|P|$ queries (since the smallest power of 2 which is at least $8|P|$ is smaller than $16|P|$). Hence $|P_3| < |P_2| + 16|P| \leq 18|P|$. Note that P_3 achieves bias at least $\gamma/2$ on every input, and that P_3 always uses a number of queries which is a power of 2 in the range $[8|P|, 4|P|/\gamma)$.

Finally, we modify P_3 to get P_4 which collapses together all quantum algorithms in the support of P_3 that use the same number of queries. That is, instead of placing support on two different quantum algorithms which both use (say) 32 queries, P_4 will place support on a single quantum algorithm which implements the mixture of both. This does not affect the number of queries or the bias of the algorithm. Hence we have $|P_4| < 18|P|$, and P_4 achieves bias at least $\gamma/2$ on each input. Further, P_4 has support on fewer than $\log(1/\gamma)$ quantum algorithms.

Next we introduce some notation for talking about P_4. Let $L = \lfloor \log(1/\gamma) \rfloor$ and let 2^k be the smallest power of 2 which is at least $4|P|$. Let the quantum algorithms in the support of P_4 be $Q_1, Q_2, \ldots, Q_L$, with Q_i using 2^{k+i} queries for each i. Let p_i be the probability P_4 assigns to algorithm Q_i. Then $p_i \geq 0$ for all i, and $\sum_{i=1}^{L} p_i = 1$. We also have $\sum_{i=1}^{L} p_i 2^{k+i} = |P_4| < 18|P|$, which means $\sum_{i=1}^{L} p_i 2^i < 5$. On input x, let $\alpha_i(x)$ be the probability that Q_i outputs 1 when run on x, and let $\beta_i(x) \coloneqq 1 - 2\alpha_i(x)$. This way, $(-1)^{f(x)}\beta_i(x)$ is the bias of Q_i when run on x. Then $\sum_{i=1}^{L} p_i \beta_i(x)$ is $(-1)^{f(x)}$ times the bias of P_4 on x, which means that it is negative if $f(x) = 1$, positive if $f(x) = 0$, and satisfies $\left|\sum_{i=1}^{L} p_i \beta_i(x)\right| \geq \gamma/2$.

We now wish to amplify P_4 from bias $\gamma/2$ to constant bias. To do so, it suffices to estimate $\sum_{i=1}^{L} p_i \beta_i(x)$ to additive error less than $\gamma/2$, and output the sign of this estimate. Our query budget for this task will be roughly $|P|/\gamma$. We know the values p_i, and seek to generate estimates $\tilde{\beta}_i(x)$ for $\beta_i(x)$. We will say an estimate $\tilde{\beta}_i(x)$ is *good* if $|\tilde{\beta}_i(x) - \beta_i(x)| \leq 2^i\gamma/10$. This way, if all $\tilde{\beta}_i(x)$ are good, our final estimate for the sum will satisfy

$$\left|\sum_{i=1}^{L} p_i \tilde{\beta}_i(x) - \sum_{i=1}^{L} p_i \beta_i(x)\right| = \left|\sum_{i=1}^{L} p_i (\tilde{\beta}_i(x) - \beta_i(x))\right| \leq \sum_{i=1}^{L} p_i |\tilde{\beta}_i(x) - \beta_i(x)| \leq \sum_{i=1}^{L} p_i 2^i \gamma/10 < \gamma/2,$$

where we used $\sum_i p_i 2^i < 5$.

To generate $\tilde{\beta}_i(x)$, we use Theorem 5.1 on algorithm Q_i with $\epsilon = 2^i\gamma/20$ and $\delta = 1/3L$. Since the query cost of Q_i is 2^{k+i}, this uses at most $2000 \cdot (2^k/\gamma) \cdot \ln(3L)$ queries. Since $2^k < 8|P|$ and $L \leq \log(1/\gamma)$, this costs $O(|P|/\gamma \cdot \log\log(1/\gamma))$. The query cost of generating all L estimates this way is therefore $O(|P|/\gamma \cdot \log(1/\gamma)\log\log(1/\gamma))$. The probability that any one estimate is not good is at most $1/3L$ by our choice of δ, so by the union bound, all are good except with probability $1/3$; hence we've given a quantum algorithm which achieves worst-case bounded error for computing f, and whose query cost is $O(\mathrm{PQ}_{\hat{\gamma}}(f)/\gamma \cdot \log(1/\gamma)\log\log(1/\gamma))$, as desired. □

Using this theorem, we now proceed to prove a strong minimax theorem for $\mathrm{PQ}_{\hat{\gamma}}(f)$, showing that a single hard distribution μ works to lower bound this measure for all values of γ at once.

Theorem 5.6. *Fix a finite alphabet Σ as well as $n \in \mathbb{N}$. Let f be a Boolean-valued function with $\mathrm{Dom}(f) \subseteq \Sigma^n$. Then there exists a distribution μ over $\mathrm{Dom}(f)$ such that for any $\gamma \in [0,1]$, we have*

$$\mathrm{PQ}^{\mu}_{\hat{\gamma}}(f) \geq \gamma \cdot \tilde{\Omega}(\mathrm{Q}(f)),$$

where the constants in the $\tilde{\Omega}$ notation are universal.

As usual, the notation $\mathrm{PQ}^{\mu}_{\hat{\gamma}}(f)$ denotes the expected cost of a probabilistic quantum algorithm which is required to achieve bias at least γ against μ (rather than in the worst case); that is, the algorithm and the bias level γ are both allowed to depend on the distribution μ. Note that since $\mathrm{PQ}(f)$ is always smaller than $\mathrm{Q}(f)$ for any given bias level, this implies Theorem 5.2.

Proof. Fix Σ, n, and f. Let $\mathcal{R}$ be the set of all probabilistic quantum algorithms for computing f. For each $P \in \mathcal{R}$ and each distribution μ over $\mathrm{Dom}(f)$, define $\mathrm{cost}(P, \mu) \coloneqq |P|$ and define $\mathrm{score}(P, \mu)$ to be the bias P makes against distribution μ for computing f (this will be in the range $[-1, 1]$). We will use Theorem 2.18. It is clear that $\mathcal{R}$ is convex, and that $\mathrm{Dom}(f)$ is a nonempty finite set. Let Δ denote the set of all probability distributions over $\mathrm{Dom}(f)$. Then cost and score are continuous functions $\mathcal{R} \times \Delta \to \mathbb{R}$, with $\mathrm{cost}(\cdot, \cdot)$ always non-negative, and both functions are linear in both variables. These functions are well-behaved, since finite cost and score can be achieved (some quantum algorithm computes f with positive bias), the cost is independent of the input, and mixing a zero-cost algorithm with a nonzero-cost algorithm gives a nonzero-cost algorithm. Hence Theorem 2.18 gives us

$$\inf_{P \in \mathcal{R}} \max_{x \in \mathrm{Dom}(f)} \frac{|P|}{\mathrm{score}(P, x)^+} = \max_{\mu \in \Delta} \inf_{P \in \mathcal{R}} \frac{|P|}{\mathrm{score}(P, \mu)^+},$$

where we use the convention $r/0 = \infty$ for all $r \in \overline{\mathbb{R}}$.

We simplify the left-hand side. For a probabilistic quantum algorithm P, use $\mathrm{bias}_f(P)$ to denote its worst-case bias, that is, $\mathrm{bias}_f(P) \coloneqq \min_{x \in \mathrm{Dom}(f)} \mathrm{score}(P, x)$. Then the left-hand side is the infimum over P of $|P|/\mathrm{bias}_f(P)^+$. Since a probabilistic algorithm P with $\mathrm{bias}_f(P) \leq 0$ will never be selected in this infimum, the left-hand side is equal to

$$\inf_{\gamma \in (0,1]} \inf_{P \in \mathcal{R}_\gamma} \frac{|P|}{\gamma},$$

where $\mathcal{R}_\gamma$ denotes the set of all probabilistic quantum algorithms which achieve worst-case bias at least γ. The inner infimum is the definition of $(1/\gamma) \cdot \mathrm{PQ}_{\hat{\gamma}}(f)$, so the left-hand side equals $\inf_{\gamma \in (0,1]} \mathrm{PQ}_{\hat{\gamma}}(f)/\gamma$.

Note that this is at most $3\,\mathrm{Q}(f)$ by picking $\gamma = 1/3$ and using $\mathrm{PQ}(f) \leq \mathrm{Q}(f)$. We claim there is no reason to use any $\gamma \in (0, 1/6\,\mathrm{Q}(f))$ in the infimum. The reason is that if P is a probabilistic quantum algorithm achieving worst-case bias at least γ such that $|P|/\gamma < 3\,\mathrm{Q}(f)$, and if $\gamma < 1/6\,\mathrm{Q}(f)$, it means that P has nonzero support on zero-cost quantum algorithms. Without loss of generality, we can assume $P = aP_0 + bP_1 + (1 - a - b)P'$, where P_0 is a zero-cost algorithm that always outputs 0, P_1 is a zero-cost algorithm that always outputs 1, and P' is a probabilistic algorithm with no support on zero-cost algorithms. Let $c = \min\{a, b\}$, and write $P = 2cZ + (1 - 2c)P''$, where Z is the 0-cost algorithm which is an even mixture of P_0 and P_1. Then it is not hard to see that $|P| = (1 - 2c)|P''|$ and $\mathrm{score}(P, \mu) = (1 - 2c)\,\mathrm{score}(P'', \mu)$ for all μ. This means that P'' has the same cost-to-score ratio as P for all distributions μ. Hence we can always use P'' in place of P for the infimum. Further, supposing without loss of generality that $b \geq a$, we have $P'' = (b - a)P_1 + (1 - b + a)P'$. Since f is not constant, let x be an input on which $f(x)$ disagrees with $P_1(x)$ (that is, a 0-input). Then note that if $b - a \geq 1/2$, the algorithm P'' cannot output 0 on x with probability above $1/2$, so $\mathrm{score}(P'', x) \leq 0$ and P'' will not be used in the infimum. On the other hand, if $b - a < 1/2$, we have $|P''| = (1 - b + a)|P'| > (1/2) \cdot 1 = 1/2$, as P' does not place weight on algorithms which make 0 queries. Now, unless P'' achieves worst-case bias at least $1/(6\,\mathrm{Q}(f))$, its ratio of cost to score would be greater than $3\,\mathrm{Q}(f)$, which we already know is achievable.

This means we only need to use $\gamma > 1/(6\,\mathrm{Q}(f))$ in the infimum. Thus the left-hand side equals

$$\inf_{\gamma \in [\frac{1}{6\,\mathrm{Q}(f)}, 1]} \frac{\mathrm{PQ}_{\dot{\gamma}}(f)}{\gamma}.$$

Using Theorem 5.5, this is at least

$$\inf_{\gamma \in [\frac{1}{6\,\mathrm{Q}(f)}, 1]} \frac{\mathrm{Q}(f)}{C \log(1/\gamma) \log\log(1/\gamma)}$$

for some universal constant C. The above is clearly optimized at $\gamma = 1/(6\,\mathrm{Q}(f))$, which means the left-hand side is at least $\Omega\left(\frac{\mathrm{Q}(f)}{\log \mathrm{Q}(f) \log\log \mathrm{Q}(f)}\right)$.

Looking at the right hand side, we see that there exists a distribution μ such that every probabilistic quantum algorithm P satisfies $|P|/\operatorname{score}(P,\mu)^+ \geq \tilde{\Omega}(\mathrm{Q}(f))$, from which the desired statement follows. □

5.2 Abstraction of the query complexity argument

We note that the argument we used to prove the existence of the hard distribution for quantum query complexity only used a few properties of quantum algorithms. Since we will want to apply the same argument to quantum communication, polynomial degree, and logrank, it makes sense to step back and provide an abstraction of this argument to more general models.

In general, we will consider Boolean-valued functions f with a finite input set $\mathrm{Dom}(f)$. We will have a set $\mathcal{A}$ of algorithms that may attempt to compute f. Formally, we will need $\mathcal{A}$ to be a subset of a real vector space. Each $A \in \mathcal{A}$ will have an associated *cost*, denoted $|A|$, with $|\cdot| \colon \mathcal{A} \to [0,\infty)$. We write $\mathcal{A}_T$ to denote the set $\{\, A \in \mathcal{A} : |A| \leq T \,\}$.

For an algorithm $A \in \mathcal{A}$ and an input $x \in \mathrm{Dom}(f)$, we let $\mathrm{bias}_f(A,x)$ denote the bias of algorithm A on input x. For now, the only property we need of the bias is that it is a function $\mathrm{bias}_f \colon \mathcal{A} \times \mathrm{Dom}(f) \to [-1,1]$. The worst-case bias of an algorithm A will be denoted $\mathrm{bias}_f(A) \coloneqq \min_{x \in \mathrm{Dom}(f)} \mathrm{bias}_f(A,x)$. If μ is a distribution over $\mathrm{Dom}(f)$, we will further write $\mathrm{bias}_f(A,\mu) \coloneqq \mathbb{E}_{x\sim\mu}[\mathrm{bias}_f(A,x)]$. Similarly, if P is a probability distribution over $\mathcal{A}$ with finite support, we denote $\mathrm{bias}_f(P,\mu) \coloneqq \mathbb{E}_{A\sim P}\mathbb{E}_{x\sim\mu}[\mathrm{bias}_f(A,x)]$ and $\mathrm{bias}_f(P) \coloneqq \min_{x\in\mathrm{Dom}(f)} \mathrm{bias}_f(P,x)$. We also set $|P| \coloneqq \mathbb{E}_{A\sim P}|A|$. Finally, we define $M(f) \coloneqq \inf_{A\in\mathcal{A}:\mathrm{bias}_f(A)\geq 1/3} |A|$.

So far, this setting is extremely general, capturing many computational models. For the quantum-style strong minimax to work, we will need the following properties to also hold for a given function f.

1. $\mathcal{A}_T$ is convex for each $T \in [0,\infty)$, and $\mathrm{bias}_f(\cdot,x)$ is linear over $A \in \mathcal{A}_T$ for each $x \in \mathrm{Dom}(f)$.
2. There exists some $A \in \mathcal{A}$ such that $\mathrm{bias}_f(A) \geq 1/3$. (Equivalently, $M(f) < \infty$.)
3. All $A \in \mathcal{A}$ with $|A| < 1$ have $|A| = 0$, and $\mathcal{A}_0$ is the convex hull of exactly two algorithms, Z_0 and Z_1. For each $x \in \mathrm{Dom}(f)$, we also have $\mathrm{bias}_f(Z_0,x) = -\mathrm{bias}_f(Z_1,x) = \pm 1$, and if f is not constant, $\mathrm{bias}_f(Z_0,x)$ attains both values 1 and -1 for $x \in \mathrm{Dom}(f)$.
4. Suppose P is a probability distribution over $\mathcal{A}$ that has support $\{A_1, A_2, \ldots, A_k\}$, with probability p_i for A_i, such that (a) $|A_i| \leq 2^i T$ for some $T \in [1/10,\infty)$, (b) $\sum_i 2^i p_i \leq 5$, and (c) $\mathrm{bias}_f(P) \geq 2^{-k-1}$. Then there is some $A \in \mathcal{A}$ with $\mathrm{bias}_f(A) \geq 1/3$ and $|A| \leq 2^k T \cdot \mathrm{poly}(k)$ (with the constants in the poly being universal).

We note that (1) essentially requires the computational model to be randomized (or, in communication complexity, to have public randomness). (2) only says that each function can be computed by some finite-cost algorithm. (3) says that algorithms with cost less than 1 cannot look at the input, and therefore have cost 0 and must either always output 0 or always output 1 (or some convex combination of the two).

The main important point is (4). This point amplifies a certain restricted type of low-bias probability distribution over algorithms into a full-blown constant-bias algorithm, and the cost of amplification is nearly linear in one over the bias.

We now prove that these points together suffice to guarantee the existence of a strongly-hard distribution. To start, we establish the following lemma, which says that if (4) holds – meaning we can amplify the restricted type of probabilistic algorithms – then we can amplify all probabilistic algorithms.

Lemma 5.7. *Suppose f and $\mathcal{A}$ satisfy the above conditions. Let P be any finite-support probability distribution over $\mathcal{A}$ with $\operatorname{bias}_f(P) > 0$. Then*

$$M(f) \leq \frac{|P|}{\operatorname{bias}_f(P)} \cdot \operatorname{polylog}(1/\operatorname{bias}_f(P)).$$

Proof. The proof of this will be directly analogous to the quantum query case. We convert P into the restricted form of (4), being careful to lose only a constant factor in the bias and in the cost. Let $\gamma \coloneqq \operatorname{bias}_f(P) > 0$. We first use Markov's inequality to argue that the total probability mass P places on algorithms A of cost $|A| \geq 2|P|/\gamma$ is at most $\gamma/2$, and hence discarding all such algorithms from the support of P decreases its bias by at most $\gamma/2$ (while not increasing its cost). Next, we group the remaining algorithms in the support of P into $\log(1/\gamma)$ bins: one bin for algorithms of cost 0 to $2T$ (with T equal to something like $4|P|$), and one additional bin for algorithms of cost 2^iT to $2^{i+1}T$ for i between 1 and $\log(1/\gamma)$. Within each bin, we use the convexity of $\mathcal{A}_{2^iT}$ to replace the entire bin with a single algorithm (whose cost is up to the upper boundary of that bin). For the first bin, this increases the cost $|P|$ by up to an additive $O(T)$, while for the other bins, this increases the cost by up to a factor of 2. Altogether, we have only $\log(1/\gamma)$ algorithms remaining in the support, and setting $k = \log(1/\gamma)$ it is not hard to check that the conditions in (4) are satisfied. □

Theorem 5.8. *Suppose f and $\mathcal{A}$ satisfy the above conditions. Then there exists a distribution μ over $\operatorname{Dom}(f)$ such that for any finite-support probability distribution P over $\mathcal{A}$, we have*

$$\operatorname{bias}_f(P, \mu) \leq O(M(f)/|P| \cdot \operatorname{polylog} M(f)).$$

In particular, if $M^\mu_\gamma(f)$ denotes the infimum cost $|A|$ over algorithms $A \in \mathcal{A}$ with $\operatorname{bias}_f(A, \mu) \geq \gamma$, then for all $\gamma \in (0, 1/3)$ we have

$$M^\mu_\gamma(f) \geq \gamma \cdot \tilde{\Omega}(M(f)).$$

Proof. The proof will be exactly the same as in the quantum query setting. In the special case where f is constant, the result trivially follows as $M(f) = 0$, so assume f is not constant.

First, we let $\mathcal{R}$ the set of all finite-support probability distributions over $\mathcal{A}$, and let Δ be the set of probability distributions over $\operatorname{Dom}(f)$. Then we define cost: $\mathcal{R} \times \Delta \to [0, \infty)$ by $\operatorname{cost}(P, \mu) \coloneqq |P|$, and score: $\mathcal{R} \times \Delta \to [-1, 1]$ by $\operatorname{score}(P, \mu) \coloneqq \operatorname{bias}_f(P, \mu)$. Note that cost and score are both continuous and linear in each variable. They are also well-behaved, because $M(f) < \infty$ ensures finite cost and score can be achieved, cost does not depend on μ, and cost is linear in P. Hence Theorem 2.18 gives

$$\inf_{P \in \mathcal{R}} \max_{x \in \operatorname{Dom}(f)} \frac{|P|}{\operatorname{bias}_f(P, x)^+} = \max_{\mu \in \Delta} \inf_{P \in \mathcal{R}} \frac{|P|}{\operatorname{bias}_f(P, \mu)^+}.$$

We examine the left-hand side. It equals $\inf_{P \in \mathcal{R}} \frac{|P|}{\text{bias}_f(P)^+}$. We note that this infimum is at most $3M(f)$ by the definition of $M(f)$. We now claim that there is no need to use any P in the infimum if $\text{bias}_f(P) < 1/(6M(f))$. To show this, it suffices to show that there is no need to use any P in the infimum if $|P| < 1/2$, because we know that $3M(f)$ is attainable using only algorithms in A with cost at least 1.

Now, suppose that $|P| < 1/2$ and $\text{bias}_f(P) > 0$. We can write $P = aZ_0 + bZ_1 + (1-a-b)P'$ where P' has support only on $A \in \mathcal{A}$ with $|A| \geq 1$. Define $P'' \coloneqq (a-c)Z_0 + (b-c)Z_1 + (1-a-b+2c)P'$, where $c = \min\{a, b\}$. Then as we showed in the quantum query case, we have $|P''|/\text{bias}_f(P'') \geq |P|/\text{bias}_f(P)$. Moreover, since f is not constant, there is some input $x \in \text{Dom}(f)$ such that $\text{bias}_f(Z_0, x) = -1$, and some input $y \in \text{Dom}(f)$ such that $\text{bias}_f(Z_1, y) = -1$. Since $\text{bias}_f(P'') > \text{bias}_f(P) > 0$, and since $\text{bias}_f(P') \leq 1$, we must have $(1-a-b+2c) > 1/2$, meaning that $|P''| > 1/2$, as desired.

Hence the left-hand side equals $\inf_{P \in \mathcal{R}'} \frac{|P|}{\text{bias}_f(P)}$, where $\mathcal{R}'$ is the set of all $P \in \mathcal{R}$ with $\text{bias}_f(P) \geq 1/(6M(f))$. Using Lemma 5.7, we know that for each $P \in \mathcal{R}'$, we have $M(f) \leq |P|/\text{bias}_f(P) \cdot \text{polylog}(1/\text{bias}_f(P)) \leq |P|/\text{bias}_f(P) \cdot \text{polylog}\, M(f)$. Hence the left-hand side is at least $\frac{M(f)}{\text{polylog}\, M(f)}$. Finally, examining the right hand side, we see that there is a distribution μ over $\text{Dom}(f)$ such that for all $P \in \mathcal{R}$, we have $|P| \geq \text{bias}_f(P) \cdot M(f)/\text{polylog}\, M(f)$, and the desired result follows. □

5.3 Quantum communication complexity

To prove an analogous minimax for quantum communication complexity, all we need is to show that quantum communication complexity satisfies the four conditions from Section 5.2. It's easy to see that as long as there is public randomness (whether or not there is also shared entanglement), the first three conditions are satisfied. It remains to deal with the fourth condition. Let P be a probability distribution over protocols $\Pi_1, \Pi_2, \ldots, \Pi_k$, which assigns probability p_i to Π_i and satisfies $|\Pi_i| \leq 2^i T$, $\sum_i 2^i p_i \leq 10$, and P achieves bias at least 2^{-k-1} for computing communication function F on any input $(x, y) \in \text{Dom}(F)$. Our goal is to construct a communication protocol which uses $T \cdot \tilde{O}(2^k)$ communication to compute F to bounded error.

As in the quantum query case, all we need to do is create a protocol Π in which Alice and Bob estimate the biases $\Pi_i(x, y)$ of the protocols Π_i when run on their inputs. Each estimate for protocol i needs to be within $2^{-(k-i)}/20$ of the correct bias, and it must satisfy this property with probability at least $1 - 1/3k$ (see the query complexity section for a formal analysis). To achieve this, it suffices for Alice and Bob to use amplitude estimation from Theorem 5.1 to generate an estimate of the probability $\Pi_i(x, y)$ outputs 1. Hence the only remaining difficulty is running amplitude estimation of a communication protocol in the communication complexity setting.

This turns out to be possible in both the shared-entanglement and the non-shared-entanglement settings (though note that we've already assumed shared randomness, so we cannot handle the non-shared-randomness non-shared-entanglement quantum communication complexity model). The idea is to have one of the players, say Alice, take charge. We will assume that Alice is the one who outputs the final answer in Π_i. Then from Alice's point of view, $\Pi_i(x, y)$ can be viewed as a unitary U and a measurement M such that Alice needs Bob's help to apply U, and after applying U to a shared state $|0\rangle_A |0\rangle_B$, Alice can apply the measurement M on her side alone to get the output $\Pi_i(x, y)$. Now, to apply amplitude estimation, Alice only needs the ability to apply controlled U, $U^\dagger$, and $(I - 2M)$ operations. She can do the latter alone. For controlled U and $U^\dagger$ applications, she needs Bob's help, but that's fine: she will just send him a qubit each time alerting him to whether they are about to apply U or $U^\dagger$ to their shared state (Bob will return that qubit afterwards to ensure coherence of Alice's controlled applications of U and $U^\dagger$).

We conclude the following theorem.

Theorem 5.9. *Let $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a (possibly partial) communication function. Then there exists a probability distribution μ over $\mathrm{Dom}(F)$ such that for all $\gamma \in (0, 1/3)$, we have*

$$\mathrm{QCC}^{\mu}_{\gamma}(F) \geq \gamma \cdot \tilde{\Omega}(\mathrm{QCC}(F)).$$

Here $\mathrm{QCC}^{\mu}_{\gamma}(F)$ denotes the minimum amount of communication required by a quantum communication protocol which achieves bias at least γ against μ. This theorem works in both the shared entanglement setting and in the shared-randomness, non-shared entanglement setting.

6 Approximate polynomial degree and logrank

As in the quantum case, polynomials can be amplified linearly in the bias. However, also as in the quantum case, the degree of polynomials is not convex: the degree of the convex combination of p_1 and p_2 is the maximum degree of p_1 and p_2, not the average degree.

The same ideas that worked for quantum query and communication complexities will allow us to get a strong hard distribution for approximate polynomial degree and approximate logrank. The main difference will be how we do the estimation of success probabilities: instead of amplitude estimation, we will need a polynomial variant of this, which turns out to be a little tricky.

6.1 Approximate degree

The approximate degree of a (possibly partial) Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is the minimum degree of an n-variate polynomial p which satisfies $|p(x) - f(x)| \leq \epsilon$ for all $x \in \mathrm{Dom}(f)$, where ϵ is a parameter representing the allowed error. When f is a partial function, there are actually two different notions of polynomial degree: one where p is required to be bounded on the entire Boolean hypercube (that is, $p(x) \in [0,1]$ for all $x \in \{0,1\}^n$, even when $x \notin \mathrm{Dom}(f)$), and one where p is not restricted outside the domain of f. Our results will apply to both versions of polynomial degree, but for conciseness, we restrict our attention to the bounded version.

With polynomials, it is often convenient to switch from talking about functions $f\colon \{0,1\}^n \to \{0,1\}$ to talking about functions $f\colon \{+1,-1\}^n \to \{+1,-1\}$. Note that by doing a simple variable substitution, we can convert between $\{0,1\}$ variables to $\{+1,-1\}$ variables without changing the degree of the polynomial. That is, we can substitute $1 - 2x_i$ in place of the variable x_i inside p to make it take $\{0,1\}$ inputs instead of $\{+1,-1\}$ inputs, and we can substitute $(1 - x_i)/2$ to go the other way. We can similarly change the output of p from being in the range $[0,1]$ to the range $[-1,1]$ and vice versa (the error changes by a factor of 2 when switching between these bases). Another well-known observation is that to approximate a Boolean function f, we only need *multilinear* polynomials, and their degree only needs to be at most n.

To get our hard distribution, we will use Theorem 5.8. We need to check the four conditions, but using polynomials as our "algorithms". More explicitly, the set $\mathcal{A}$ will be the set of all real n-variate multilinear bounded polynomials, viewed in the $\{+1,-1\}$ basis (bounded means that $p(x) \in [-1,1]$ for all $x \in \{+1,-1\}^n$). For a polynomial $p \in \mathcal{A}$, we define $\mathrm{bias}_f(p,x)$ to be $f(x)p(x)$. Then (1) holds, as the set of polynomials of a given degree is convex and $\mathrm{bias}_f(\cdot,x)$ is linear over that set. (2) holds because every Boolean function can be computed exactly by a polynomial of degree n. Next, (3) holds because polynomials of degree less than 1 have degree 0, and since we're dealing with bounded polynomials, these are a convex combination of the two constant polynomials -1 and 1.

It remains to show (4). To this end, let P be a probability distribution over k polynomials $q_1, q_2, \ldots, q_k$, with $\deg(q_i) \leq 2^i T$. Let p_i be the probability P assigns to q_i, and suppose $\sum_{i=1}^{k} 2^i p_i \leq 5$. Finally, suppose that $\text{bias}_f(P) \geq 2^{-k-1}$. Our goal is to find a polynomial q of degree at most $2^k T \cdot \text{poly}(k)$ that computes f to constant error. To do so, we'll need a polynomial version of the amplitude estimation algorithm we did in the quantum case. That is, we'd like to estimate the output that polynomial $q_i(x)$ returns, and do arithmetic computations on it. Crucially, one of the arithmetic computations we'd like to do is comparison, for example, to see if $q_i(x) > 0$. Such a comparison is not a polynomial operation, so we cannot use the polynomial $q_i(x)$ itself. What we'll do instead is to create polynomials that compute the *bits* of the binary expansion of $q_i(x)$, to a certain precision. We will then do arithmetic operations using those bits, and we'll be able to implement those operations using polynomials.

To do so, we'll need some approximation theory. The following theorem, known as Jackson's theorem, will be useful. It traces back to Jackson (1911) [Jac11], but see also [MMR94] (page 750, Theorem 3.1.1) for some discussion and a more thorough list of references.

Theorem 6.1 (Jackson's theorem)**.** *Let $\alpha\colon [-1,1] \to \mathbb{R}$ be a continuous function, and let $n \in \mathbb{N}$. Then there is a real polynomial p of degree n such that for all $x \in [-1,1]$, we have*

$$|p(x) - \alpha(x)| \leq 6 \cdot \sup_{|y-z| \leq 1/n} |\alpha(y) - \alpha(z)|.$$

In particular, if α has Lipschitz constant K, then for each $n \in \mathbb{N}$ there is a polynomial p_n of degree at most n which approximates α to within an additive $6K/n$ at each point in $[-1,1]$. Jackson's theorem can be used to prove the well-known result that polynomials can be amplified with a linear dependence in the bias. For completeness, we reprove this here (see also e.g. [GKKT17]).

Corollary 6.2 (Polynomial amplification (small bias to constant bias))**.** *For each $\gamma \in (0,1)$, there is a real polynomial p of degree at most $13/\gamma$ such that p maps $[-1,1]$ to $[-1,1]$, p maps $[-1,-\gamma]$ to $[-1,-1/3]$, and p maps $[\gamma,1]$ to $[1/3,1]$.*

Proof. Let $\alpha\colon [-1,1] \to \mathbb{R}$ be the function with $\alpha(x) = -2/3$ for $x \in [-1,-\gamma]$, $\alpha(x) = 2/3$ for $x \in [\gamma,1]$, and $\alpha(x) = 2x/3\gamma$ for $x \in (-\gamma,\gamma)$. Then α is continuous and has Lipschitz constant $2/3\gamma$. By Theorem 6.1, for every $n \in \mathbb{N}$, there exists a polynomial p_n of degree at most n which approximates α to additive error $4/\gamma n$. Picking $n = \lceil 12/\gamma \rceil \leq 13/\gamma$, we get a polynomial which approximates α to error $1/3$, which means it has the desired properties. □

We will also need an amplification polynomial that goes from constant bias to small error. We reprove the following well-known lemma here for completeness (it also appears in [BNRW07], and another version appears in [She13]).

Lemma 6.3 (Polynomial amplification (constant error to small error))**.** *For each $\epsilon \in (0, 2/3)$, there is a real polynomial p of degree at most $17\log(1/\epsilon)$ such that p maps $[-1,1]$ to $[-1,1]$, p maps $[-1,-1/3]$ to $[-1,-(1-\epsilon)]$, and p maps $[1/3,1]$ to $[1-\epsilon,1]$.*

Proof. We set

$$q(x) = \sum_{i=0}^{k} \binom{2k+1}{i} \left(\frac{1+x}{2}\right)^i \left(\frac{1-x}{2}\right)^{2k+1-i},$$

and set $p(x) = 1 - 2q(x)$. Note that for $x \in [-1,1]$, the value $q(x)$ is exactly the probability that, when flipping a coin $2k+1$ times, less than half of the coin flips will come out heads, assuming the probability of heads is $(1+x)/2$. Because of this interpretation, we know that q maps $[-1,1]$

to $[0,1]$ and is decreasing in x, so p maps $[-1,1]$ to $[-1,1]$ and is increasing in x. We also have $q(x) = 1 - q(-x)$, which means that $p(-x) = -p(x)$, i.e. p is odd. Given these properties, the lemma will follow if we show that $p(1/3) \geq 1 - \epsilon$, or equivalently, that $q(1/3) \leq \epsilon/2$.

We have

$$q(1/3) = \sum_{i=0}^{k} \binom{2k+1}{i} \left(\frac{2}{3}\right)^i \left(\frac{1}{3}\right)^{2k+1-i} = 3^{-(2k+1)} \sum_{i=0}^{k} \binom{2k+1}{i} 2^i \leq 3^{-(2k+1)} 2^k \sum_{i=0}^{k} \binom{2k+1}{i}$$

$$= 3^{-(2k+1)} 2^k 2^{2k} = (1/3)(8/9)^k.$$

To get this to be smaller than $\epsilon/2$, it suffices to pick k large enough so that $(8/9)^k \leq \epsilon$, or equivalently, $k \geq \frac{1}{\log(9/8)} \log(1/\epsilon)$. Hence we can pick $k = \lceil \frac{1}{\log(9/8)} \log(1/\epsilon) \rceil \leq \frac{1}{\log(9/8)} \log(1/\epsilon) + 1$. The degree of p will be $2k+1 \leq \frac{2}{\log(9/8)} \log(1/\epsilon) + 3$. Note that $\epsilon \leq 2/3$, so $\log(1/\epsilon) \geq \log(3/2)$, and hence $\frac{2}{\log(9/8)} \log(1/\epsilon) + 3 \leq \left(\frac{2}{\log(9/8)} + \frac{3}{\log(3/2)}\right) \log(1/\epsilon) \leq 17 \log(1/\epsilon)$. □

Equipped with these approximation-theoretic tools, we will now tackle (4), showing that probability distributions over polynomials (which achieve a small amount of worst-case bias γ for computing f) can be amplified to polynomials which compute f to constant error, using only a nearly-linear dependence on $1/\gamma$.

Lemma 6.4. *As in (4), let P be a probability distribution over k bounded multilinear polynomials $q_1, q_2, \ldots, q_k$, which assigns them probabilities $p_1, p_2, \ldots, p_k$ respectively. Suppose that $\sum_{i=1}^{k} 2^i p_i \leq 5$, that $\deg(q_i) \leq 2^i T$ for some real number T, and that $f(x) \sum_{i=1}^{k} p_i q_i(x) \geq 2^{-k-1}$ for all $x \in \mathrm{Dom}(f)$. Then there is a bounded multilinear polynomial q which approximates f with bias at least $1/3$ and which satisfies $\deg(q) \leq 2^k T \cdot \mathrm{poly}(k)$.*

Proof. Recall that in the quantum case, we estimated the bias of the i-th algorithm to within $2^{-(k-i)}/20$, with success probability at least $1 - 1/3k$. We will do a polynomial version of this. What does estimating $q_i(x)$ mean, for polynomials? It means we will construct polynomials which approximately compute the bits in the binary expansion of the number $q_i(x)$. We will have one polynomial for the sign, and an additional $k - i + 4$ polynomials for the first $k - i + 4$ digits in the binary expansion of $q_i(x)$.

In order to do so, we compose univariate polynomials with q_i. This way, the task reduces to creating univariate polynomials which output the bits in the binary expansion of their input (assuming they all receive the same input). More explicitly, the correctness condition is as follows. We say the binary expansion of a real number $\beta \in [-1,1]$ is $2^{-\ell}$-*robust* to t bits if the first t bits of the binary expansion of $\beta + \epsilon$ is the same as that of β for all $\epsilon \in [-2^{-\ell}, 2^{-\ell}]$. Then we require univariate polynomials $d_0^\ell, d_1^\ell, \ldots, d_k^\ell$ such that if $\beta \in [-1,1]$ is $2^{-\ell}$-robust to at least t bits, then $d_t^\ell(\beta)$ is within $O(1/k^{10})$ of the t-th bit in the binary expansion of β. The polynomial d_0^ℓ needs to output the sign of β if β is $2^{-\ell}$-robust to at least 0 bits (that is, if the sign of β does not change upon adding or subtracting $2^{-\ell}$). We will also require all these polynomials to be bounded, i.e. they must map $[-1,1]$ to $[-1,1]$.

To implement these polynomials, we use Theorem 6.1. For simplicity, let's represent the bits in the binary expansion using $+1$ and -1 instead of 0 and 1 (converting back is easy). Consider the function α_i which outputs the i-th bit of the binary expansion of its input (or the sign if $i = 0$). This i is a step function: for $i = 0$, $\alpha_0(\beta)$ jumps from -1 to 1 at $\beta = 0$; for $i = 1$, $\alpha_1(\beta)$ similarly jumps from -1 to 1 and back at $\beta = -1/2, 0, 1/2$. More generally, α_i has 2^{i+1} different plateaus of 1 or -1 on its domain $[-1,1]$. Now, since we only care about getting the i-th bit correct if the i-th bit is robust to β changing by $2^{-\ell}$, consider the continuous functions α_i^ℓ which make the jumps from

-1 to 1 continuous by starting from $2^{-\ell}$ before the jump point, ending $2^{-\ell}$ after the jump point, and drawing a continuous line in between (the slope of the line will be $\pm 2^{-\ell}$). This is well-defined as long as ℓ is sufficiently larger than i, say $\ell \geq i+2$.

Note that α_i^ℓ has Lipschitz constant $2^{-\ell}$. This means we can use Theorem 6.1 to estimate α_i^ℓ by a polynomial of degree $O(2^\ell)$ which achieves constant additive error (say, $1/10$). We can scale down these polynomials slightly to ensure they remain bounded in $[-1,1]$. We then plug them into a single variate bounded polynomial of degree $O(\log k)$ that we get from Lemma 6.3, in order to amplify the error down to $O(1/k^{10})$. The result are polynomials d_t^ℓ (for $\ell \geq t+2$) that have degree $O(2^\ell \log k)$ and, on input β which is $2^{-\ell}$-robust to bit at least t, correctly output the t-th bit of β except with additive error $O(1/k^{10})$.

Now, to get an estimate of $q_i(x)$ to $k-i+5$ bits, we set $\ell = k-i+O(\log k)$ and compose $d_t^\ell(q_i(x))$ for $t = 0, 1, 2, \ldots, k-i+5$. Actually, we scale down $q_i(x)$ and add an extra variable y_i representing a noise term for $q_i(x)$; the final estimating polynomials will be the $n+1$ variate polynomials $r_{i,t}(x, y_i) := d_t^\ell((9/10)q_i(x) + y_i)$. Note that the degree of $r_{i,t}$ is $O(2^{k-i+O(\log k)} \log k \cdot \deg(q_i)) = O(2^k T \operatorname{poly}(k))$.

Next, consider the function which takes binary representations (to $k+5$ bits each) of numbers $\lambda_i \in [-1,1]$, and outputs the sign of $\sum_{i=1}^k p_i \lambda_i$, where p_i are known non-negative constants which sum to 1. This is a Boolean function of $O(k^2)$ variables, so it can be computed exactly by a multilinear polynomial of degree $O(k^2)$. Call this polynomial s. Next, plug in the polynomials $r_{i,t}$ into the inputs of s, so that s calculates the sign of the sum $\sum_{i=1}^k p_i \tilde{\beta}_i$ where each $\tilde{\beta}_i$ is the estimate of $(9/10)q_i(x) + y_i$ that is computed by the polynomials d_t^{k-i+10}. Call this composed polynomial $u(x, y)$.

Observe that $u(x, y)$ is a polynomial in $n+k$ variables (n variables from x and k variables y_i), and has degree $O(2^k T \operatorname{poly}(k))$. This polynomial attempts to compute the sign of $(9/10)\sum_{i=1}^k p_i q_i(x) + \sum_{i=1}^k p_i y_i$. Since we know that $\sum_{i=1}^k p_i q_i(x) \cdot f(x) \geq 2^{-k-1}$, this sign computed by $u(x, y)$ will equal $f(x)$ so long as $\left|\sum_{i=1}^k p_i y_i\right| \leq 2^{-k-2}$. Recall that $\sum_{i=1}^k 2^i p_i \leq 5$. Hence to guarantee that $\left|\sum_{i=1}^k p_i y_i\right| \leq 2^{-k-2}$, it suffices to choose each y_i such that $|y_i| \leq 2^{-(k-i+5)}$. Now, let's call $q_i(x) + y_i$ good if it is $2^{-(k-i+O(\log k))}$-robust to $k-i+5$ bits. If all $q_i(x) + y_i$ are good for all i, then $r_{i,t}$ correctly compute the bits to additive error $O(1/k^{10})$, then a multilinear polynomial of degree $O(k^2)$ in $O(k^2)$ variables will still correctly compute its output to small error, certainly $O(1/k)$. Hence if all $q_i(x) + y_i$ are good for all i and if $y_i \leq 2^{-k-2}/k$ for all i, $u(x, y)$ outputs $f(x)$ to error $O(1/k)$.

To ensure that $q_i(x) + y_i$ are good, we pick y_i at random. That is, we have an allowed range $[-2^{-(k-i+5)}, 2^{-(k-i+5)}]$ for y_i; we fit $\operatorname{poly}(k)$ evenly spaced points into this range, so that the gap between the points is $2^{-(k-i+O(\log k))}$. Note that for all but a constant number of choices of y_i among these $\operatorname{poly}(k)$ options, the resulting number $q_i(x) + y_i$ will be $2^{-(k-i+O(\log k))}$-robust to $k-i+5$ bits. Hence by randomly selecting y_i, the probability that $q_i(x) + y_i$ is not good is at most $O(1/\operatorname{poly}(k))$. By the union bound, this choice means that all $q_i(x) + y_i$ are good except with constant probability. Hence $u(x, y)$ computes $f(x)$ to $O(1/k)$ error with high probability when y is chosen at random according to the above procedure.

Finally, we let $q(x)$ be the average of the polynomials $u(x, y)$ for all possible choices of y in the above procedure. Since $u(x, y)$ outputs a number very close to $f(x)$ when y is good, and since it is always bounded in $[-1, 1]$, and since y is good with high probability, we conclude that $q(x)$ computes $f(x)$ to bounded error. It is also bounded outside the promise of f. The degree of $q(x)$ was $O(2^k T \operatorname{poly}(k))$. We note that $q(x)$ as we constructed it here can actually be viewed as a polynomial ρ in k variables composed with the polynomials $q_1, q_2, \ldots, q_k$. □

The above amplification theorem allows us to conclude the following theorem.

Theorem 6.5. *let $f\colon \{+1,-1\}^n \to \{+1,-1\}$ be a (possibly partial) Boolean function. Then there is a vector $\psi \in [-1,1]^{\mathrm{Dom}(f)}$ such that $\|\psi\|_1 = 1$, $\langle \psi, f\rangle = 1$, and for any polynomial p which is bounded (i.e. $|p(x)| \le 1$ for $x \in \{+1,-1\}^n$), we have*

$$\langle \psi, p\rangle \le \frac{\deg(p)}{\tilde{\Omega}\left(\mathrm{adeg}(f)\right)}.$$

Here $\mathrm{adeg}(f)$ denotes the minimum degree of a bounded polynomial p which computes f to bounded error. The constants in the $\tilde{\Omega}$ notation are universal.

Proof. This follows immediately by taking ψ to be defined by $\psi(x) = f(x)\mu[x]$, where μ is the hard distribution we get from Theorem 5.8. □

6.2 Approximate logrank and gamma 2 norm

Instead of tackling approximate logrank directly, we use approximate γ_2 norm. This measure deserves some introduction. First, we note that the γ_2 norm is a well-known norm of a matrix. One way to define it is to say that $\gamma_2(A)$ is the minimum, over factorizations $A = BC$ of A into a product of matrices B and C, of the maximum 2-norm of a row of B times the maximum 2-norm of a column of C. The γ_2 norm has several useful properties known in the literature [She12; LSŠ08]:

1. γ_2 is a norm, so $\gamma_2(A) \ge 0$ (with equality if and only if A is the all-zeros matrix) and $\gamma_2(A + \lambda B) \le \gamma_2(A) + |\lambda|\gamma_2(B)$.
2. $\gamma_2(A \otimes B) = \gamma_2(A)\gamma_2(B)$, where $\otimes$ denotes the tensor (Kronecker) product
3. $\gamma_2(A \circ B) \le \gamma_2(A)\gamma_2(B)$, where $\circ$ denotes the Hadamard (entry-wise) product
4. $\gamma_2(J) = 1$ where J is the all-ones matrix
5. $\|A\|_\infty \le \gamma_2(A) \le \|A\|_\infty \sqrt{\mathrm{rank}(A)}$.

In the above, A and B are matrices of the same dimensions, and λ is a scalar. $\gamma_2(A)$ can be thought of as a smoother version of rank.

Let $F\colon \mathcal{X} \times \mathcal{Y} \to \{+1,-1\}$ be a (possibly partial) communication function. We identify F with its *communication matrix*, which is a matrix with rows indexed by $\mathcal{X}$ and columns indexed by $\mathcal{Y}$, with the (x,y) entry being $F(x,y) \in \{+1,-1\}$ if $(x,y) \in \mathrm{Dom}(F)$ and being $*$ if $(x,y) \notin \mathrm{Dom}(F)$. This way, F is a $\{+1,-1,*\}$-valued matrix.

For such a matrix F, we say that a real-valued matrix A *approximates* F (to bias $1/3$) if $|A[x,y]| \le 1$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$ and $F(x,y)A[x,y] \ge 1/3$ for all $(x,y) \in \mathrm{Dom}(F)$. The *approximate γ_2 norm* of F, denoted $\tilde{\gamma}_2(F)$, is defined as the minimum value of $\gamma_2(A)$ over all matrices which approximate F to bias $1/3$. It is not hard to see that this minimum is attained, as the set of such matrices is compact.

We will actually care about the logarithm of the approximate γ_2 norm, that is, about $\log \tilde{\gamma}_2(F)$. We note that the constant $1/3$ in the definition of this measure is arbitrary, as approximations to F can be amplified with only a constant factor overhead in the log-approximate-γ_2-norm (see, e.g., [BBGK18]). An annoying detail, however, is that such amplification can in general lose not just a multiplicative constant but also an additive constant, since $\tilde{\gamma}_2(F)$ may in general be less than 1 (meaning the logarithm of it will be less than 0). To avoid such complications, we will define our measure of interest as $M(F) \coloneqq \max\{1, \log \tilde{\gamma}_2(F)\}$ if F is not constant and $M(F) = 0$ if F is constant, and we will write $M_\gamma(F)$ for the bias γ version of $M(F)$ instead of the default bias $1/3$ version.

In order to get a minimax theorem analogous to Theorem 6.5, we will again use Theorem 5.8. Our set of algorithms $\mathcal{A}$ will be the set of bounded real matrices A (that is, real matrices A of the same dimensions as F which satisfy $|A[x,y]| \leq 1$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$). The cost of a matrix A will be $\text{cost}(A) \coloneqq \max\{1, \log \gamma_2(A)\}$ if A is not a multiple of the all-ones matrix J, and otherwise $\text{cost}(A) = 0$ if $A = \lambda J$. We define $\text{bias}_F(A,(x,y)) = F(x,y)A[x,y]$ for $(x,y) \in \text{Dom}(F)$.

We show that $\mathcal{A}_T$ is convex for each $T \in [0,\infty)$. For $T < 1$, the set $\mathcal{A}_T$ is the set of all matrices of the form λJ for $\lambda \in [-1,1]$, which is clearly convex. For $T \geq 1$, suppose $A, B \in \mathcal{A}_T$ and let $\lambda \in (0,1)$. Then $\text{cost}(\lambda A + (1-\lambda)B)$ is either 0, 1, or $\log \gamma_2(\lambda A + (1-\lambda)B)$. In the former two cases, we clearly have $\lambda A + (1-\lambda)B \in \mathcal{A}_T$, so consider the latter case. We have $\log \gamma_2(\lambda A + (1-\lambda)B) \leq \log(\lambda \gamma_2(A) + (1-\lambda)\gamma_2(B)) \leq \log \max\{\gamma_2(A), \gamma_2(B)\} = \max\{\log \gamma_2(A), \log \gamma_2(B)\} \leq \max\{\text{cost}(A), \text{cost}(B)\} \leq T$. Hence $\mathcal{A}_T$ is convex. It is also clear that $\text{bias}_F(\cdot,(x,y))$ is linear, so (1) is satisfied.

By taking A to equal F inside $\text{Dom}(F)$ and to be 0 elsewhere, we get $\text{bias}_F(A) = 1$, so (2) is satisfied. By our definition of $\text{cost}(A)$, we have $\text{cost}(A) \geq 1$ or $\text{cost}(A) = 0$, with the latter happening only if A is a convex combination of J and $-J$, so (3) is satisfied.

As usual, it remains to handle (4). We do so in the following lemma.

Lemma 6.6. *Let P a probability distribution over matrices $A_1, A_2, \ldots, A_k$ with probability p_i for A_i. Suppose that $\sum_{i=1}^k 2^i p_i \leq 5$, and that for all i, we have $\text{cost}(A_i) \leq 2^i T$ for some real number $T \geq 1/10$. Suppose further that $\text{bias}_F(P) \geq 2^{-k-1}$. Then there is some bounded matrix A which approximates F to bias $1/3$ and satisfies $\text{cost}(A) \leq 2^k T \cdot \text{poly}(k)$ (with the constants in the poly being universal).*

Proof. Let ρ be the polynomial from the proof of Theorem 6.5 with respect to the probabilities $p_1, p_2, \ldots, p_k$. This is a polynomial in k variables with the property that if values $\beta_1, \beta_2, \ldots, \beta_k$ are plugged in and $|\sum_i p_i \beta_i| \geq 2^{-k-1}$, then $\rho(\beta_1, \beta_2, \ldots, \beta_k)$ returns the sign of $\sum_i p_i \beta_i$ to bounded error. The polynomial ρ further has the property that it is bounded (i.e. it returns values in $[-1,1]$ when given inputs in $[-1,1]^k$), and that if you plug in any polynomials q_i in place of β_i, with $\deg(q_i) \leq 2^i$, then the degree of the composed polynomial is at most $2^k \text{poly}(k)$.

This latter property means that the *weighted degree* of ρ with weights $(2^1, 2^2, \ldots, 2^k)$ is at most $O(2^k \text{poly}(k))$. Here the term weighted degree means that we count the degree of each monomial of ρ differently depending on the variables in that monomial: the i-th variable gets weight 2^i, so a monomial of the form $\beta_1^{c_1}\beta_2^{c_2}\ldots\beta_k^{c_k}$ will have weighted degree $2^i c_1 + 2^2 c_2 + \cdots + 2^k c_k$. We know that the weighted degree of ρ, meaning the maximum weighted degree of one of its monomials, is at most $O(2^k \text{poly}(k))$.

We will now use this polynomial ρ to construct a matrix A which approximates F and has γ_2 norm that is not too large. The idea is to simply apply ρ to the matrices $A_1, A_2, \ldots, A_k$, using the Hadamard product for multiplication and the usual matrix addition and scalar multiplication. Since γ_2 is a norm, we know that $\gamma_2(\rho(A_1, A_2, \ldots, A_k))$ is the sum, over all monomials of ρ, of the absolute value of the coefficient of that monomial multipled by the γ_2-norm of the Hadamard product defined by that monomial. This is upper bounded by the sum of absolute coefficients of ρ (which we'll denote C) multiplied by the γ_2 norm of the largest monomial.

The γ_2 norm of a single monomial $\beta_1^{c_1}\ldots\beta_k^{c_k}$ composed with matrices $A_1, \ldots, A_k$ is at most $\gamma_2(A_1)^{c_1}\ldots\gamma_2(A_k)^{c_k}$, since the γ_2 norm is sub-multiplicative under the Hadamard product. Hence $\log \gamma_2(\rho(A_1, \ldots A_k))$ is at most $\log C$ plus the maximum value of $c_1 \log \gamma_2(A_1) + \cdots + c_k \log \gamma_2(A_k)$ for some monomial $(c_1, c_2, \ldots, c_k)$ of ρ. Since $\log \gamma_2(A) \leq \text{cost}(A)$ for all bounded matrices A, and since $\text{cost}(A_i) \leq 2^i T$, this maximum is at most the maximum of $T \cdot (2^1 c_1 + \cdots + 2^k c_k)$ over monomials of ρ, which is at most $O(2^k T \text{poly}(k))$.

We now upper bound C, the sum of absolute coefficients of ρ. Recall that ρ was constructed as an average of different polynomials with different values of the constants y_i. Let ρ' be the polynomial

within that set we averaged over which has the largest sum of absolute coefficients. Then to upper bound C it suffices to upper bound the sum of absolute coefficients of ρ'. To do so, we essentially want to replace all coefficients of ρ' with their absolute values, and then plug in all ones for the variables. We note that $(9/10) + y_i$ will be at most 1 for the values of y_i used in ρ', which means that if we replace the terms $(9/10)q_i + y_i$ with simply q_i, we would only increase the sum of absolute coefficients (here we treat q_i as variables).

Let the resulting polynomial be ρ''. Then ρ'' is simply the result of composing the polynomial s with the polynomials $r_{i,t}$. Since s is a bounded multilinear polynomial of degree $O(k^2)$, its sum of absolute coefficients is at most $2^{O(k^2)}$, and it is not hard to see that the sum of absolute coefficients of ρ'' will be at most $2^{O(k^2)}$ times $D^{O(k^2)}$, where D is the maximum sum of absolute coefficients over the polynomials d_t^ℓ with $\ell = k - i + O(\log k)$. In other words, $\log C \leq O(k^2) + O(k^2 D)$, where D is the sum of absolute coefficients of some such polynomial d_t^ℓ.

The polynomial d_t^ℓ is a single variate bounded polynomial of degree at most $O(2^\ell \log k)$, which, using $\ell \leq k + O(\log k)$, is at most $2^k \operatorname{poly}(k)$. A bounded univariate polynomial of this degree must have sum of absolute coefficients at most $4^{2^k \operatorname{poly}(k)}$ by [She13] (Lemma 4.1). Hence $\log D \leq 2^k \operatorname{poly}(k)$, so $\log C \leq 2^k \operatorname{poly}(k)$.

We conclude that if $A = \rho(A_1, A_2, \ldots, A_k)$, then $\log \gamma_2(A) \leq 2^k (T+1) \operatorname{poly}(k)$, and hence $\operatorname{cost}(A) \leq 2^k (T+1) \operatorname{poly}(k)$. This is at most $O(2^k T \operatorname{poly}(k))$ since we have $T \geq 1/10$. Further, each entry $A[x, y]$ is equal to $\rho(A_1[x, y], A_2[x, y], \ldots, A_k[x, y])$, which means that A is bounded (since ρ is bounded and the matrices A_i are bounded), and for $(x, y) \in \operatorname{Dom}(F)$, we have $F(x, y)A[x, y] \geq 1/3$ by the guarantees on A_i and on ρ. □

Using Theorem 5.8, we can now conclude the following theorem.

Theorem 6.7. *Let $F \colon \mathcal{X} \times \mathcal{Y} \to \{+1, -1\}$ be a (possibly partial) communication function. Then there is a distribution μ over $\operatorname{Dom}(F)$ such that for any bounded real matrix A (meaning $|A[x, y]| \leq 1$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$), we have*

$$\mathbb{E}_{(x,y)\sim\mu}[F(x, y)A[x, y]] \leq \frac{\log \gamma_2(A)}{\tilde{\Omega}(\log \tilde{\gamma}_2(F))}.$$

Note that for bounded matrices, $\log \gamma_2(A) \leq \log \operatorname{rank}(A)$. We also have, from [LS09],

$$\log \widetilde{\operatorname{rank}}(F) \leq 6 \log \tilde{\gamma}_2(F) + O(\log \log |\mathcal{X} \times \mathcal{Y}|).$$

This means we can write a minimax theorem for logrank as well.

Theorem 6.8. *Let $F \colon \mathcal{X} \times \mathcal{Y} \to \{+1, -1\}$ be a (possibly partial) communication function, and suppose that $\log \widetilde{\operatorname{rank}}(F) \geq C \log \log |\mathcal{X} \times \mathcal{Y}|$ where C is a universal constant. Then there is a distribution μ over $\operatorname{Dom}(F)$ such that for any bounded real matrix A (meaning $|A[x, y]| \leq 1$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$), we have*

$$\mathbb{E}_{(x,y)\sim\mu}[F(x, y)A[x, y]] \leq \frac{\log \operatorname{rank}(A)}{\tilde{\Omega}(\log \widetilde{\operatorname{rank}}(F))}.$$

In other words, μ is such that if A has low rank compared to F, then A cannot correlate well with F under μ, and hence A does not approximate F very well against μ.

7 Circuit complexity

A Boolean circuit C is a collection of gates connected to each other and to bits of its input x by wires, with a single output wire representing the value of $C(x)$. The *size* of a circuit is the number of gates in the circuit, and the *depth* of a circuit is the length of the longest path between an input bit and an output wire. A *randomized Boolean circuit* is a probability distribution over Boolean circuits, and the *size* of a randomized Boolean circuit is defined to be the expected size of a Boolean circuit drawn from that distribution.

In Section 7.1, we examine the randomized circuit complexity of partial Boolean functions when it is computed by circuits of unbounded fan-in and unlimited depth. In Section 7.2, we show that the main result also holds in the NC^1 setting of logarithmic-depth circuits whose gates each have fan-in at most 2. Finally, in Section 7.3 we establish the strengthening of the hardcore lemma.

7.1 General circuits

In this section, let $\mathrm{R}(f)$ denote the minimum size of a randomized Boolean circuit of unbounded fan-in and unlimited depth that computes the partial Boolean function f with error at most $\frac{1}{3}$ on every input $x \in \mathrm{Dom}(f)$. Similarly, let $\mathrm{R}^{\mu}_{\dot{\gamma}}(f)$ denote the minimum size of randomized Boolean circuits that compute f with error at most $\dot{\gamma} = \frac{1-\gamma}{2}$ when the input is drawn from μ. We establish a relation between those two complexity measures via the study of forecasting circuits.

Definition 7.1. *A* forecasting circuit *is a randomized Boolean circuit with one modification: instead of having a single output wire, the forecasting circuit has $k+1$ output wires that represent the binary encoding of a value in the range $\{0, \frac{1}{2^k}, \frac{2}{2^k}, \ldots, \frac{2^k-1}{2^k}, 1\}$.*

The *resolution* of a forecasting circuit is k when it has $k+1$ output wires. (Or, equivalently, when it outputs values that are multiples of 2^{-k}.) The *score* of a forecasting circuit is computed in the same way as we did for forecasting algorithms in previous sections. The *size* of a randomized forecasting circuit is, as in the the case of randomized Boolean circuits, the expected number of gates in a circuit drawn from the distribution. Forecasting circuits can be defined for each model of randomized Boolean circuits; in this section, we consider forecasting circuits with unbounded fan-in and unlimited depth.

We begin by showing that if there is a Boolean circuit that computes a function with non-negligible advantage over random guessing, then there is also a forecasting algorithm with non-trivial score.

Proposition 7.2. *For any partial function $f : \{0,1\}^n \to \{0,1\}$, if there is a size $s \geq 1$ and parameter $\gamma \geq \frac{4}{R(f)+1}$ for which there is a randomized Boolean circuit R of average size s that satisfies $\Pr_{C\sim R}[C(x) \neq f(x)] \leq \dot{\gamma}$ for every $x \in \mathrm{Dom}(f)$, then there is also a randomized forecasting circuit R' with resolution $\lceil \log R(f) \rceil$, average size at most $s+1$, and h-score*

$$\mathrm{score}(R', x) = \mathop{\mathbb{E}}_{C'\sim R'}[\mathrm{score}(C'(x), f(x))] \geq \gamma^2/8$$

for each $x \in \mathrm{Dom}(f)$.

Proof. For each circuit C in the support of R, define C' to be the forecasting circuit of resolution $k = \lceil \log R(f) \rceil$ and size $\mathrm{size}(C) + 1$ which outputs the value

$$\frac{1 + (-1)^{C(x)}\gamma'}{2}$$

on input $x \in S$ where $\gamma' = \frac{2m}{2^k}$ for the largest integer m such that $\gamma' \leq \gamma$. The definition of γ' guarantees that $\gamma - \frac{2}{2^k} \leq \gamma' \leq \gamma$. The value of k and the lower bound on γ in the proposition statement imply that $\gamma - \frac{2}{2^k} \geq \gamma - \frac{2}{R(f)+1} \geq \frac{\gamma}{2}$, so $\frac{\gamma}{2} \leq \gamma' \leq \gamma$.

This circuit C' can be constructing by adding a single extra $\neg$ gate to the output wire of C: the output of C and its negations can then be combined with constant value wires to generate the two required output values of the forecasting circuit. (Namely, if the two output values $(1 \pm \gamma')/2$ of C' are denoted by $z^{(0)}$ and $z^{(1)}$, then the ith output bit of C' is a hardcoded constant value 0 or 1 when $z^{(0)} = z^{(1)}$ and otherwise it is either $C(x)$ or $\neg C(x)$ when $z^{(0)} \neq z^{(1)}$.)

The randomized forecasting circuit R' is then defined to be the distribution on circuits obtained by drawing $C \sim R$ and outputing the modified circuit C' as described above. Following the same argument as in Lemma 3.15, the score of this randomized forecasting circuit satisfies

$$\operatorname{score}(R', x) \geq \gamma'^2/2 \geq \gamma/8. \qquad \square$$

In the second step in the proof of Theorem 7.8, we show that the minimax theorem applies in this setting.

Lemma 7.3. *Fix any $k \geq 1$ and let $\mathcal{R}_k$ denote the set of all randomized forecasting circuits with resolution k. Then for partial function $f : \{0,1\}^n \to \{0,1\}$, if we let Δ denote the set of distributions over $\operatorname{Dom}(f)$, we have*

$$\inf_{R \in \mathcal{R}_k} \max_{x \in \operatorname{Dom}(f)} \frac{\operatorname{size}(R)}{\operatorname{score}(R,x)^+} = \max_{\mu \in \Delta} \inf_{R \in \mathcal{R}_k} \frac{\operatorname{size}(R)}{\operatorname{score}(R,\mu)^+}.$$

Proof. The lemma follows from Theorem 2.18, and the argument showing that the conditions of that theorem are satisfied follows closely the analogous argument of Theorem 4.2.

First, we want to show that $\mathcal{R}_k$ can be viewed as a convex subset of a real topological space V. We can do so with the same construction as in Theorem 4.2, though here we can also use a slightly simpler constructions: fix $V = \mathbb{R}^{|\operatorname{Dom}(f)|+1}$, and for each randomized forecasting circuit $R \in \mathcal{R}_k$ define $v_R(x) = \operatorname{score}(R, x)$ for each $x \in \operatorname{Dom}(f)$ and define the $|\operatorname{Dom}(f)| + 1$th coordinate of v_R to be $\operatorname{cost}(R, x)$. That the resulting set is convex follows directly from the fact that a vector $v' = \lambda v_{R_1} + (1-\lambda) v_{R_2}$ for any $R_1, R_2 \in \mathcal{R}_k$ corresponds to the vector of the randomized forecasting circuit $R' = \lambda R_1 + (1-\lambda) R_2$.

The linearity of cost and score measures in both R and μ follows from their definition.

Lastly, the notions of cost and score satisfy the well-behaved condition of Theorem 2.18. First, because the existence of a circuit of size at most $2^{|S|}$ that computes f exactly implies the existence of a finite-cost and non-zero score randomized forecasting circuit for any distribution μ on $\operatorname{Dom}(f)$. Second, because the cost of circuits does not depend on the input, and third because the definition of cost immediately implies that the mixture of a zero-cost and a nonzero-cost randomized circuit gives a nonzero-cost randomized circuit. $\square$

The next step is the main one in the proof of the theorem: we want to show that the score of forecasting circuits can be amplified efficiently.

Lemma 7.4. *For every partial Boolean function $f : \{0,1\}^n \to \{0,1\}$, when we set $k = \lceil \log R(f) \rceil$ then*

$$\inf_{R \in \mathcal{C}_k} \max_{x \in \operatorname{Dom}(f)} \frac{\operatorname{size}(R)}{\operatorname{score}(R,x)^+} = \widetilde{\Omega}\big(R(f)\big)$$

where the $\widetilde{\Omega}$ hides terms that are polylogarithmic in $R(f)$.

The proof of the lemma uses the following bounds regarding the circuit complexity of basic arithmetic operations.

Proposition 7.5 ([BCH86; Alt88])**.** *For any two numbers a, b represented to accuracy 2^{-k} in binary, then the values*

$$ab, \qquad \frac{a}{1-a}, \qquad \ln(a), \qquad e^a, \text{ and } \qquad \frac{1}{1+a}$$

can all be computed to additive accuracy 2^{-k} by circuits of size polynomial in k and depth $O(\log k)$.

We also need another result regarding the circuit complexity of iterated multiplication up to fixed accuracy.

Proposition 7.6. *When $a_1, \ldots, a_m$ and $b_1, \ldots, b_m$ are k-bit integers, then there is a circuit of size $O(m \log m + mk + k^c)$ for some constant $c \geq 1$ and depth $O(\log k + \log m)$ that computes the ratio*

$$\frac{a_1 \cdots a_m}{b_1 \cdots b_m}$$

up to multiplicative accuracy 1 ± 2^{-k}.

Proof. This result can be obtained by computing $\ln \frac{a_1 \cdots a_m}{b_1 \cdots b_m} = \sum_{i=1}^{m} \ln a_i - \ln b_i$ to additive accuracy 2^{-k}. The computation of each of the values $\ln a_i$ and $\ln b_i$ for $1 \leq i \leq m$ up to additive accuracy $\frac{2^{-k}}{2m}$ can be done with a circuit of size polynomial in $n := k + \log m + 1$ and depth $O(\log n)$. The sum of the $2m$ terms can be done with a circuit for iterated addition of size $O(mn) = O(m \log m + mk)$ and depth $O(\log m + \log n) = O(\log m + \log k)$ to compute the natural log of the ratio up to additive error 2^{-k}. [Ofm62] (See also [Pip87; Weg87] and the references therein.) Finally, a circuit of size polynomial in k and depth logarithmic in k can be used to compute the exponential of the final ratio. □

Using these propositions, we can complete the proof of the lemma.

Proof of Lemma 7.4. Let R be a randomized forecasting circuit which comes arbitrarily close to the infimum on the left-hand side.

Consider the randomized forecasting circuit R' obtained by drawing m forecasting circuits $C_1, \ldots, C_m$ independently at random from R and combining their output values using the formula

$$C'(x) = \left(1 + \prod_{i \leq m} \frac{1 - C_i(x)}{C_i(x)}\right)^{-1}.$$

Fixing $m = \max_x 1/\operatorname{score}(R, x)^+$, we obtain a randomized circuit R' with score $\operatorname{score}(R', x)^+ = \Omega(1)$ for each $x \in S$ and average size

$$\operatorname{size}(R') = \operatorname{size}(R) \cdot m + O(m \log m + mk + k^c)$$

for some universal constant $c \geq 1$. Then the proof is completed by noting that $m = O(\frac{\mathrm{R}(f)}{\operatorname{size}(R)})$. □

Finally, we show that when there is a forecasting circuit with score γ, there is also a Boolean circuit with error at most $\dot{\gamma}$.

Proposition 7.7. *For any partial function $f : \{0,1\}^n \to \{0,1\}$, if there is a size $s \geq 1$ and parameter γ for which there is a randomized forecasting circuit R with k output wires, size s, depth d and $\operatorname{score}(R, x) \geq \gamma$ for each $x \in \operatorname{Dom}(f)$, then there is also a randomized Boolean circuit R' of size $s + O(k)$ and depth $d + O(1)$ that satisfies $\Pr_{C \sim \mathcal{R}}[C(x) = f(x)] \geq \frac{1+\gamma}{2}$ for every $x \in \operatorname{Dom}(f)$.*

Proof. Given a forecasting circuit C that outputs the value p on input x, we want to design a randomized Boolean circuit R_C that outputs the value 1 with probability p and 0 with probability $1-p$ on input x.

We can do this by adding k random inputs $r_1, \ldots, r_k$ that are used to generate a uniformly random value $r \in \{\frac{1}{2^k}, \frac{2}{2^k}, \ldots, 1\}$. Then if the value p in the output of the circuit is 0, we output zero; otherwise we use a comparator circuit to return 1 if and only if $r \leq p$. This value has the desired bias p, and using standard constructions (see, e.g. [Weg87; Vol99]) we can implement the comparator circuit with $O(k)$ gates in a circuit of constant depth (in the unbounded fan-in model; or $O(\log n)$ depth in the bounded fan-in model).

The final randomized Boolean circuit R' is defined by drawing a forecasting circuit C from R and outputting R_C. The bound on the error of R' is then obtained as in the argument of Lemma 3.15. □

Putting the above lemmas and propositions together completes the proof of the following theorem.

Theorem 7.8. *Fix $n \in \mathbb{N}$. For every partial function $f : \{0,1\}^n \to \{0,1\}$, there is a distribution μ on $\mathrm{Dom}(f)$ such that for all $\gamma \in (0,1]$,*

$$\mathrm{R}^{\mu}_{\dot{\gamma}}(f) = \tilde{\Omega}\big(\gamma^2 \,\mathrm{R}(f)\big).$$

7.2 Circuits with bounded depth

Define $\mathrm{RNC1}(f)$ to be the minimum size of a randomized Boolean circuit of fan-in two and logarithmic depth that computes the partial Boolean function f with error at most $\frac{1}{3}$ on every input $x \in \mathrm{Dom}(f)$. Similarly, let $\mathrm{RNC1}^{\mu}_{\dot{\gamma}}(f)$ denote the minimum size of a randomized Boolean circuit with the same fan-in and depth restrictions that computes f with error at most $\dot{\gamma} = \frac{1-\gamma}{2}$ when the input is drawn from μ.

The constructions of Proposition 7.2, Lemma 7.4, and Proposition 7.7 can all be achieved with circuits of fan-in 2 that add only logarithmic depth overhead to the base circuits, so the analogue of Theorem 7.8 also holds for the class of circuits of fan-in two and logarithmic depth.

Theorem 7.9. *Fix $n \in \mathbb{N}$. For every partial function $f : \{0,1\}^n \to \{0,1\}$, there is a distribution μ on $\mathrm{Dom}(f)$ such that for all $\gamma \in (0,1]$,*

$$\mathrm{RNC1}^{\mu}_{\dot{\gamma}}(f) = \tilde{\Omega}\big(\gamma^2 \,\mathrm{RNC1}(f)\big).$$

In fact, we can say even more about the efficiency of the transformations in each constructions: all three of them can be accomplished with constant-depth and polynomial-size overhead when the circuits have threshold gates. For Proposition 7.2, this is because only a single additional gate is required. For Lemma 7.4, this is because the functions in Proposition 7.5 can all be computed the the required accuracy with threshold circuits of polynomial size and constant depth [RT92] and the iterated addition problem can be solved by a threshold circuit of constant depth and size $O(m \log m(k + \log m))$ [CSV84]. And for Proposition 7.7, this is because comparison can also be completed with polynomial-size and constant-depth circuits. Therefore, letting $\mathrm{RTC0}_{\epsilon}(f)$ denote the minimum size of a randomized constant-depth threshold circuit with unbounded fan-in that computes f with error probability at most $\frac{1}{3}$ on every input and $\mathrm{RNC1}^{\mu}_{\dot{\gamma}}(f)$ denote the minimum size of the same type of circuit that computes $f(x)$ correctly with probability $\frac{1+\gamma}{2}$ when x is drawn from μ, we obtain the following result.

Theorem 7.10. *Fix $n \in \mathbb{N}$. For every partial function $f : \{0,1\}^n \to \{0,1\}$, there is a distribution μ on $\mathrm{Dom}(f)$ such that for all $\gamma \in (0,1]$,*

$$\mathrm{RTC0}^{\mu}_{\dot{\gamma}}(f) = \tilde{\Omega}\big(\gamma^2 \,\mathrm{RTC0}(f)\big).$$

7.3 Hardcore lemma

In order to complete the proof of the hardcore lemma as stated in Theorem 1.9, we need the following variant of the ratio minimax theorem that applies to the setting where we consider a compact convex set of distributions, not just the set of all distributions over the function's domain.

Theorem 7.11. *Let V be a real topological vector space, and let $\mathcal{R} \subseteq V$ be convex. Let S be a nonempty finite set, and let Δ be a compact and convex set of probability distributions over S, viewed as a subset of $\mathbb{R}^{|S|}$. Let* $\mathrm{cost}\colon \mathcal{R} \times \Delta \to [0,\infty]$ *be semicontinuous and saddle, and let* $\mathrm{score}\colon \mathcal{R} \times \Delta \to [-\infty,\infty)$ *be such that its negation,* $-\mathrm{score}$*, is semicontinuous and saddle. Suppose* cost *and* score *are well-behaved. Then using the convention $r/0 = \infty$ for all $r \in [0,\infty]$, we have*

$$\inf_{R\in\mathcal{R}} \max_{\mu\in\Delta} \frac{\mathrm{cost}(R,\mu)}{\mathrm{score}(R,\mu)^+} = \max_{\mu\in\Delta} \inf_{R\in\mathcal{R}} \frac{\mathrm{cost}(R,\mu)}{\mathrm{score}(R,\mu)^+}.$$

Proof. The proof is identical to the one for (the first part of) Theorem 2.18, since that argument only uses the fact that the set of all distributions over S is convex and compact. □

From this theorem we obtain the following variant of Lemma 7.3 for distributions with min-entropy δ.

Lemma 7.12. *Fix any $k \geq 1$ and let $\mathcal{R}_k$ denote the set of all randomized forecasting circuits with resolution k. Then for every $\delta > 0$ and function $f : \{0,1\}^n \to \{0,1\}$, if we let Δ_δ denote the set of distributions over $\{0,1\}^n$ with min-entropy δ, we have*

$$\inf_{R\in\mathcal{R}_k} \max_{\mu\in\Delta_\delta} \frac{\mathrm{size}(R)}{\mathrm{score}(R,\mu)^+} = \max_{\mu\in\Delta_\delta} \inf_{R\in\mathcal{R}_k} \frac{\mathrm{size}(R)}{\mathrm{score}(R,\mu)^+}.$$

We are now ready to complete the proof of Theorem 1.9.

Proof of Theorem 1.9. Fix $s' = c \cdot s/\log\frac{1}{\delta}$ for some constant c to be fixed later. By Lemma 7.12, the two cases to consider are the following.

Case 1: $\max_{\mu\in\Delta_\delta} \inf_{R\in\mathcal{R}_k} \frac{\mathrm{size}(R)}{\mathrm{score}(R,\mu)^+} \geq s'$.

Fix a distribution μ with min-entropy δ for which the maximum is attained. Then every randomized forecasting circuit R has score

$$\mathrm{score}(R,\mu) \leq \frac{\mathrm{size}(R)}{s'}.$$

By Proposition 7.2, this implies that every randomized circuit R' with $\mathrm{size}(R') \leq \epsilon^2 s'/64$ has success probability

$$\Pr_{C\sim R, x\sim\mu}[C(x) = f(x)] \leq \frac{1 + 8\sqrt{\mathrm{size}(R')/s'}}{2} \leq \frac{1+\epsilon}{2},$$

and the theorem holds in this case.

Case 2: $\inf_{R \in \mathcal{R}_k} \max_{\mu \in \Delta_\delta} \frac{\text{size}(R)}{\text{score}(C,\mu)^+} < s'$.

Fix a randomized forecasting circuit R that satisfies

$$\frac{\text{size}(R)}{\text{score}(C,\mu)^+} < s'$$

for each distribution μ over $\{0,1\}^n$ with min-entropy δ. Set $\alpha = \text{size}(R)/s'$ and define $T \subseteq \{0,1\}^n$ to be the set of inputs x for which $\text{score}(R,x) < \frac{\alpha}{2}$. Then

$$|T| \leq \delta(1 - \tfrac{\alpha}{2})2^n$$

since otherwise the score of R on the distribution μ' that is uniform over any set $T' \supseteq T$ of size $|T'| = \delta 2^n$ (and thus has min-entropy δ) would be bounded above by $\text{score}(R,\mu') < (1-\frac{\alpha}{2}) \cdot \frac{\alpha}{2} + \frac{\alpha}{2} < \alpha$, contradicting the definition of R.

By Lemma 7.4, there is a forecasting circuit R' which satisfies $\text{size}(R') = O(s')$ and $\text{score}(R,x) = \Omega(1)$ for each $x \in \{0,1\}^n \setminus T$. Then by Proposition 7.7 there is a randomized Boolean circuit of size $O(s')$ that errs with probability at most $\frac{1}{3}$ on each $x \in \{0,1\}^n \setminus T$, and by standard success amplification it also means that there is a circuit C of size $s'' = O(s' \log \frac{1}{\delta})$ with error less than δ. Choosing the value c in the definition of s' appropriately, we then get that this circuit has size at most s, contradicting the premise of the theorem and therefore showing that Case 2 cannot occur. □

Acknowledgements

We thank Justin Thaler for discussions and references related to approximate polynomial degree and its amplification. We also thank Andrew Drucker, Mika Göös, and Li-Yang Tan for correspondence about their ongoing work [BDG+20]. We thank anonymous reviewers for many helpful comments.

A Proofs related to the minimax theorem

Lemma 2.8 (An upper semicontinuous function on a compact set attains its max)**.** *Let X be a nonempty compact topological space, and let $\phi : X \to \overline{\mathbb{R}}$ be a function. Then if ϕ is upper semicontinuous, it attains its maximum, meaning there is some $x \in X$ such that for all $x' \in X$, $\phi(x') \leq \phi(x)$. Similarly, if ϕ is lower semicontinuous, it attains its minimum.*

Proof. The lower semicontinuous case follows from the upper semicontinuous case simply by negating ϕ, so we focus on the upper semicontinuous case. Let $z = \sup_{x \in X} \phi(x)$, where $z \in \overline{\mathbb{R}}$. Let x_0 be any element of X. If $\phi(x_0) = z$, we are done, so assume $\phi(x_0) < z$; in particular, $z > -\infty$. We define a sequence $x_1, x_2, \ldots$ as follows. If $z < \infty$, define x_i to be any element of X such that $\phi(x_i) > z - 1/i$. If $z = \infty$, define x_i to be any element of X such that $\phi(x_i) > i$. Moreover, for each $i \in \mathbb{N}$, let $U_i = \{ x \in X : \phi(x) < \phi(x_i)\}$. Note that any $x \in X$ for which $\phi(x) < z$ must be in U_i for some $i \in \mathbb{N}$; hence if the supremum z is not attained, the sets U_i form a cover for X (meaning $\bigcup_{i \in \mathbb{N}} U_i = X$).

The key claim is that the U_i sets are all open if ϕ is upper semicontinuous. This is is because if $x \in U_i$, then $\phi(x) < \phi(x_i)$, and by the definition of upper semicontinuity, there is a neighborhood U of x on which $\phi(\cdot)$ is still less than $\phi(x_i)$; thus there is a neighborhood U of x contained in U_i, so that U_i is open. In this case, if the supremum z is not attained, the collection $\{U_i\}_{i \in \mathbb{N}}$ is an open cover of X, and by the definition of compactness, it has a finite subcover. Let i be the largest index of some U_i in this subcover. Then it follows that $\phi(x) < \phi(x_i)$ for all $x \in X$, which is a contradiction. Hence the supremum z must be attained as a maximum, as desired. □

Lemma 2.9 (A pointwise infimum of upper semicontinuous functions is upper semicontinuous)**.** *Let X be a topological space, let I be a set, and let $\{\phi_i\}_{i \in I}$ be a collection of functions $\phi_i : X \to \overline{\mathbb{R}}$. Then if each ϕ_i is upper semicontinuous, the function $\phi(x) = \inf_{i \in I} \phi_i(x)$ is also upper semicontinuous. Similarly, if each ϕ_i is lower semicontinuous, the pointwise supremum is lower semicontinuous.*

Proof. Note that the case where ϕ_i are all lower semicontinuous follows from the case where they are all upper semicontinuous simply by negating the functions, since negation flips upper and lower semicontinuity and flips infimums and supremums. We focus on the case where ϕ_i are all upper semicontinuous.

Fix $x \in X$. If $\phi(x) = \infty$, ϕ is upper semicontinuous at x by definition. If $\phi(x) < \infty$, fix any $y > \phi(x)$. By the definition of $\phi(x)$ as an infimum, there is some $i \in I$ such that $\phi_i(x) < y$. By the upper semicontinuity of $\phi_i(\cdot)$, there is a neighborhood U of x such that for all $x' \in U$, we have $\phi_i(x') < y$. Then for all $x' \in U$, we clearly have $\phi(x') = \inf_{i \in I} \phi_i(x') < y$. Thus ϕ is upper semicontinuous at x, as desired. □

Lemma A.1. *Let V be a real vector space, and let $X \subseteq V$. The convex hull of X is the set of all $v \in V$ which can be written as a convex combinatotion of vectors in x; that is, v for which there exist $k \in \mathbb{N}$, $x_1, x_2, \ldots, x_k \in X$, and $\lambda_1, \lambda_2, \ldots, \lambda_k \in [0, 1]$ with $\lambda_1 + \lambda_2 + \cdots + \lambda_k = 1$ such that $v = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_k x_k$.*

Proof. This is a well-known characterization of the convex hull, which can be shown as follows: let Y be the set of all finite convex combinations of points in X; that is, Y contains all points in V of the form $\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_k x_k$, where $k \in \mathbb{N}$, $x_1, x_2, \ldots, x_k \in X$, and $\lambda_1, \lambda_2, \ldots, \lambda_k \in [0, 1]$ with $\lambda_1 + \lambda_2 + \cdots + \lambda_k = 1$. Then Y is clearly convex, since for all $y_1, y_2 \in Y$ and $\lambda \in (0, 1)$, we know that y_1 and y_2 are finite convex combinations of points in x, meaning that $\lambda y_1 + (1 - \lambda) y_2$ is also a finite convex combination of points in X. Furthermore, if Z is any other convex set containing X, then it's easy to show by induction that Z contains all convex combinations of k points in X

for each $k \in \mathbb{N}$; hence Z must be a superset of Y. It follows that $\mathrm{Conv}(X)$, the intersection of all convex sets containing X, must exactly equal Y. □

Lemma 2.10 (Quasiconvex functions on convex hulls)**.** *Let V be a real vector space, let $X \subseteq V$, and let $\phi\colon \mathrm{Conv}(X) \to \overline{\mathbb{R}}$ be a function. If ϕ is quasiconvex, then*

$$\sup_{x \in \mathrm{Conv}(X)} \phi(x) = \sup_{x \in X} \phi(x).$$

Similarly, if ϕ is quasiconcave, then

$$\inf_{x \in \mathrm{Conv}(X)} \phi(x) = \inf_{x \in X} \phi(x).$$

Proof. The quasiconcave case follows from the quasiconvex case by negating ϕ; hence it suffices to prove the quasiconvex case. It is clear that $\sup_{x \in \mathrm{Conv}(X)} \phi(x)$ is at least $\sup_{x \in X} \phi(x)$, so we only need to show the latter is at least the former. To this end, let $y^* := \sup_{x \in \mathrm{Conv}(X)} \phi(x)$, and let $\hat{x} \in \mathrm{Conv}(X)$ be such that $\phi(\hat{x})$ is arbitrarily close to y^*. We must show that $\sup_{x \in X} \phi(x) \geq \phi(\hat{x})$, or equivalently, that there is some $x \in X$ with $\phi(x) \geq \phi(\hat{x})$.

Using Lemma A.1, we can now write $\hat{x} \in \mathrm{Conv}(X)$ as $\hat{x} = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_k x_k$, with $k \in \mathbb{N}$, $x_1, x_2, \ldots, x_k \in X$, and $\lambda_1, \lambda_2, \ldots, \lambda_k \in [0, 1]$ with $\lambda_1 + \lambda_2 + \cdots + \lambda_k = 1$. Furthermore, assume that $\lambda_i > 0$ for each $i \in [k]$ (we can remove $\lambda_i x_i = 0$ from the linear combination otherwise). Now, note that by quasiconvexity, we have $\phi(\lambda x_1 + (1 - \lambda) x_2) \leq \max\{\phi(x_1), \phi(x_2)\}$. It is not hard to show by induction that $\phi(\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_k x_k) \leq \max\{\phi(x_1), \phi(x_2), \ldots, \phi(x_k)\}$. Hence there is some $x \in X$ such that $\phi(x) \geq \phi(\hat{x})$, as desired. □

Lemma 2.15. *Let V be a real topological vector space, and let $X \subseteq V$ be convex. For a function $\psi\colon X \to \overline{\mathbb{R}}$, let ψ^+ denote the function $\psi^+(x) = \max\{\psi(x), 0\}$. Then this operation on ψ preserves convexity, quasiconvexity, quasiconcavity, upper semicontinuity, and lower semicontinuity, but not concavity.*

We actually prove a stronger statement, where the maximum is taken with an arbitrary constant.

Lemma A.2. *Let V be a real topological vector space, and let $X \subseteq V$ be convex. Let $\psi\colon X \to \overline{\mathbb{R}}$ be a function, let $c \in \mathbb{R}$ be a constant, and let $\psi'\colon X \to \overline{\mathbb{R}}$ be the function $\psi'(x) = \max\{\psi(x), c\}$. Then if ψ is convex, ψ' is convex; if ψ is quasiconvex, ψ' is quasiconvex; if ψ is quasiconcave, ψ' is quasiconcave; if ψ is upper semicontinuous, ψ' is upper semicontinuous; and if ψ is lower semicontinuous, ψ' is lower semicontinuous.*

Proof. Let $x, y \in X$, and let $\lambda \in (0, 1)$. Then

$$\psi'(\lambda x + (1 - \lambda) y) = \max\{\psi(\lambda x + (1 - \lambda) y), c\}.$$

If this maximum equals c, it is certainly at most $\lambda \max\{\psi(x), c\} + (1 - \lambda) \max\{\psi(y), c\}$, since these two latter maximums are each at least c. Hence the inequalities for convexity and quasiconvexity always hold when the original maximum equals c. Alternatively, if $\max\{\psi(\lambda x + (1 - \lambda) y), c\} = \psi(\lambda x + (1 - \lambda) y)$, then using $\psi(x) \leq \psi'(x)$ and $\psi(y) \leq \psi'(y)$, we see that convexity of ψ gives the inequality for convexity of ψ', and quasiconvexity of ψ gives the inequality for quasiconvexity of ψ'.

Next, suppose ψ is quasiconcave. Without loss of generality, say that $\psi(x) \leq \psi(y)$. Then $\psi'(\lambda x + (1 - \lambda) y) = \max\{\psi(\lambda x + (1 - \lambda) y), c\} \geq \max\{\psi(x), c\} = \psi'(x) \geq \min\{\psi'(x), \psi'(y)\}$, and ψ' is quasiconcave.

Preservation of lower semicontinuity follows from Lemma 2.9, where we note that c is continuous as a function from X to $\overline{\mathbb{R}}$. It remains to show upper semicontinuity is preserved. Suppose ψ is upper semicontinuous, and let $x \in X$. If $\psi'(x) = \infty$, upper semicontinuity at x vacuously holds. Fix $y > \psi'(x)$. Since $\psi'(x) \geq c$, we have $y > c$, so $\psi(x) = \psi'(x) > y$, and upper semicontinuity gives us a neighborhood U of x on which $\psi(\cdot)$ is less than y. Since $y > c$, we have $\psi'(\cdot) = \max\{c, \psi(\cdot)\} < y$ on U. Hence ψ' is upper semicontinuous. □

Theorem A.3 (Sion's minimax [Sio58])**.** *Let V_1 and V_2 be real topological vector spaces, and let $X \subseteq V_1$ and $Y \subseteq V_2$ be convex. Let $\alpha : X \times Y \to \mathbb{R}$ be semicontinuous and quasisaddle. If either X or Y is compact, then*

$$\inf_{x \in X} \sup_{y \in Y} \alpha(x, y) = \sup_{y \in Y} \inf_{x \in X} \alpha(x, y).$$

Theorem 2.11 (Sion's minimax for extended reals)**.** *Let V_1 and V_2 be real topological vector spaces, and let $X \subseteq V_1$ and $Y \subseteq V_2$ be convex. Let $\alpha : X \times Y \to \overline{\mathbb{R}}$ be semicontinuous and quasisaddle. If either X or Y is compact, then*

$$\inf_{x \in X} \sup_{y \in Y} \alpha(x, y) = \sup_{y \in Y} \inf_{x \in X} \alpha(x, y).$$

Proof. First, note that the inf-sup is always at least the sup-inf. This is because these expressions can be thought of as two players, one choosing x and trying to minimize $\alpha(x, y)$, and the other choosing y and trying to maximize y; in the inf-sup, the sup player chooses y after already knowing x, and therefore has more information and is better positioned to maximize $\alpha(x, y)$ than in the sup-inf, where the inf player goes second.

Now, let

$$a \coloneqq \sup_{y \in Y} \inf_{x \in X} \alpha(x, y), \qquad b \coloneqq \inf_{x \in X} \sup_{y \in Y} \alpha(x, y).$$

We have $a, b \in \overline{\mathbb{R}}$, and $a \leq b$. We wish to show $a = b$. Suppose by contradiction that $a < b$. Then we can pick $a', b' \in \mathbb{R}$ such that $a < a' < b' < b$. We then define $\alpha' : X \times Y \to \mathbb{R}$ by $\alpha'(x, y) \coloneqq a'$ if $\alpha(x, y) \leq a'$, $\alpha'(x, y) \coloneqq b'$ if $\alpha'(x, y) \geq b'$, and $\alpha'(x, y) \coloneqq \alpha(x, y)$ if $\alpha(x, y) \in [a', b']$.

Note that $\alpha'(x, y) = \max\{a', \min\{b', \alpha(x, y)\}\}$. By Lemma A.2, we know that taking a maximum with a constant preserves quasiconvexity, quasiconcavity, and upper and lower semicontinuities. By negating the function, it also follows that taking a minimum with a constant preserves these properties. From this it follows that α' is quasisaddle and semicontinuous, since α has these properties.

Now, since $a = \sup_{y \in Y} \inf_{x \in X} \alpha(x, y)$ and since $a' > a$, we know that for all $y \in Y$, there exists some $x \in X$ for which $\alpha(x, y) < a'$. This means that for all $y \in Y$, there exists $x \in X$ for which $\alpha'(x, y) = a'$. Hence $\sup_{y \in Y} \inf_{x \in X} \alpha'(x, y) = a'$. Similarly, since $b = \inf_{x \in X} \sup_{y \in Y} \alpha(x, y)$ and since $b' < b$, we know that for all $x \in X$, there exists some $y \in Y$ for which $\alpha(x, y) > b'$. This means that for all $x \in X$, there exists $y \in Y$ for which $\alpha'(x, y) = b'$. Hence $\inf_{x \in X} \sup_{y \in Y} \alpha'(x, y) = b'$. By Theorem A.3, we then have

$$b' = \inf_{x \in X} \sup_{y \in Y} \alpha'(x, y) = \sup_{y \in Y} \inf_{x \in X} \alpha'(x, y) = a'.$$

But this is a contradiction, since we picked $a' < b'$. We conclude that we must have had $a = b$ to begin with, as desired. □

B Distance measures

Lemma 3.3. hs, Brier, *and* ls *are proper scoring rules.* bias *is a scoring rule which is not proper.*

Proof. It is clear that all of the functions from Definition 3.2 are smooth on $(0,1)$ and increasing on $[0,1]$, where we interpret $\mathrm{hs}(0) = \mathrm{ls}(0) = -\infty$. It is also clear that all these functions evaluate to 1 at 1 and to 0 at $1/2$. It remains to show that Brier, ls, and hs are proper. To do so, we need to show that $ps(q) + (1-p)s(1-q)$ is uniquely optimized at $q = p$ when s is one of these functions and $p \in (0,1)$. Fix such $p \in (0,1)$, and observe that the critical points of the expression we wish to maximize are the points q such that $ps'(q) = (1-p)s'(1-q)$.

For $\mathrm{ls}(q) = 1 - \log(1/q) = 1 + (\log e)\ln q$, the critical points q satisfy $(\log e)p/q = (\log e)(1-p)/(1-q)$, or $p/(1-p) = q/(1-q)$. Noting that the function $x/(1-x)$ is increasing on $(0,1)$, and hence injective on $(0,1)$, we conclude that the only critical point is $q = p$. Moreover, at the boundaries $q = 0$ and $q = 1$, we clearly have $p\,\mathrm{ls}(q) + (1-p)\,\mathrm{ls}(1-q) = -\infty$, whereas in the interior the expression is finite. Hence the unique maximum must occur at $q = p$.

For $\mathrm{hs}(q) = 1 - \sqrt{(1-q)/q} = 1 - \sqrt{1/q - 1}$, we have $\mathrm{hs}'(q) = 1/2\sqrt{q^3(1-q)}$, so the critical points q satisfy $p/2\sqrt{q^3(1-q)} = (1-p)/2\sqrt{(1-q)^3 q}$, or $p/q = (1-p)/(1-q)$, which once again only occurs at $q = p$. At the boundaries, we once again have $p\,\mathrm{hs}(q) + (1-p)\,\mathrm{hs}(1-q) = -\infty$ for $q = 0$ or $q = 1$, so the unique maximum occurs at $q = p$.

Finally, for $\mathrm{Brier}(q) = 1 - 4(1-q)^2 = -4q^2 + 8q - 3$, we have $\mathrm{Brier}'(q) = 8(1-q)$, so the critical points q satisfy $8p(1-q) = 8(1-p)q$, which again implies $q = p$. This time, the boundary points are finite, but we can use the second order condition: the second derivative of $p\,\mathrm{Brier}(q) + (1-p)\,\mathrm{Brier}(1-q)$ is $p\,\mathrm{Brier}''(q) + (1-p)\,\mathrm{Brier}''(1-q)$. Noting that $\mathrm{Brier}''(q) = -8$, this is $-8p - 8(1-p) = -8 < 0$. Hence the critical point is a maximum, and since it is unique (with the boundaries 0 and 1 not being critical even if we extend the domain of the function), we conclude it is the unique maximum. □

Lemma B.1. *For any $x \in [0,1]$, we have*

$$\frac{x^2}{2} \le 1 - \sqrt{1-x^2} \le 1 - H\left(\frac{1+x}{2}\right) \le x^2 \le x.$$

Additionally, x^2 and $1 - \sqrt{1-x}$ are convex functions on $[0,1]$.

Proof. $x^2 \le x$ is clearly true for $x \in [0,1]$. To see that $x^2/2 \le 1 - \sqrt{1-x^2}$, note that this is equivalent to $y/2 \le 1 - \sqrt{1-y}$ for $y \in [0,1]$ (by setting $y = x^2$); the latter is clearly true at $y = 0$, so it suffices to show the right hand side grows faster. Taking derivatives, it suffices to show $1/2 \le 1/2\sqrt{1-y}$, which is clearly true for $y \in [0,1]$.

Next, write

$$1 - H((1+x)/2) = 1 - ((1+x)/2)\log 2/(1+x) - ((1-x)/2)\log 2/(1-x)$$

$$= 1 - (1+x)/2 - (1-x)/2 + ((1+x)/2)\log(1+x) + ((1-x)/2)\log(1-x)$$

$$= \frac{1}{\ln 4}((1+x)\ln(1+x) + (1-x)\ln(1-x)).$$

Let $\alpha(x) = (1+x)\ln(1+x) + (1-x)\ln(1-x)$. We show that $\alpha(x)/x^2$ is increasing and that $\alpha(x)/(1-\sqrt{1-x^2})$ is decreasing; this suffices to show the desired inequalities, since it means we only need to check $x = 1$, where the inequalities $1 - \sqrt{1-x^2} \le 1 - H((1+x)/2) \le x^2$ hold with equality.

The derivative of $\alpha(x)$ is $\ln(1+x)-\ln(1-x)$. The derivative of $\alpha(x)/x^2$ is therefore $x^2\ln(1+x)-x^2\ln(1-x)-2x(1+x)\ln(1+x)-2x(1-x)\ln(1-x)$ divided by $x^4>0$ (for $x\in(0,1)$). This simplifies to $-2x\ln(1-x^2)-x^2\ln((1+x)/(1-x))$. This is positive if and only if $2\ln(1-x^2)+x\ln((1+x)/(1-x))$ is negative. This expression equals 0 at $x=0$, so it suffices to show it is decreasing on $(0,1)$. The derivative is $-2x/(1-x^2)+\ln((1+x)/(1-x))$, which is again 0 at $x=0$, so it again suffices to show the derivative is negative on $(0,1)$. The derivative of this expression is $-4x^2/(1-x^2)^2$, which is finally a quantity that is clearly negative, completing the argument; hence $\alpha(x)/x^2$ is increasing on $[0,1]$.

The derivative of $\alpha(x)/(1-\sqrt{1-x^2})$ is

$$(1-x-\sqrt{1-x^2})\ln(1-x)-(1+x-\sqrt{1-x^2})\ln(1+x)$$

divided by some denominator which is positive on $(0,1)$. This equals

$$-x\ln(1-x^2)-(1-\sqrt{1-x^2})\ln((1+x)/(1-x)).$$

Note that $\ln(1-x^2)=-x^2-x^4/2-\cdots-x^{2i}/i-\ldots$ and that $\ln((1+x)/(1-x))=\ln(1+x)-\ln(1-x)=2x+2x^3/3+\cdots+2x^{2i-1}/(2i-1)+\ldots$, so the expression equals

$$x^2\sum_{i=1}^{\infty}x^{2i-1}/i-(1-\sqrt{1-x^2})\sum_{i=1}^{\infty}x^{2i-1}/(i-1/2)=(\sqrt{1-x^2}-(1-x^2))\sum_{i=1}^{\infty}-x^{2i-1}/i(2i-1)<0.$$

Hence $\alpha(x)/(1-\sqrt{1-x^2})$ is decreasing on $[0,1]$, as desired.

It is clear that x^2 and $1-\sqrt{1-x}$ are convex functions on $[0,1]$, as their second derivatives are $2>0$ and $(1/4)(1-x)^{-3/2}>0$ (for $x\in(0,1)$) respectively. □

Lemma 3.6 (Relations between distance measures)**.** *When applied to fixed ν_0, ν_1, and w, the distance measures satisfy*

$$\frac{\mathrm{S}^2}{2}\le 1-\sqrt{1-\mathrm{S}^2}\le \mathrm{h}^2\le \mathrm{JS}\le \mathrm{S}^2$$

as well as

$$\Delta^2\le \mathrm{S}^2\le\Delta.$$

We also have $\mathrm{JS}\le \mathrm{h}^2/\ln 2$ *and* $\mathrm{S}^2\le(\ln 4)\,\mathrm{JS}$.

Proof. We use Lemma B.1. The chain $\mathrm{h}^2\le\mathrm{JS}\le\mathrm{S}^2\le\Delta$ follows from the inequalities there, while the inequalities $\Delta^2\le\mathrm{S}^2$ and $1-\sqrt{1-\mathrm{S}^2}\le\mathrm{h}^2$ follow from Jensen's inequality combined with the convexity of x^2 and $1-\sqrt{1-x}$.

Finally, to show inequality $\mathrm{JS}\le\mathrm{h}^2/\ln 2$ we only need to compute the limit of $\alpha(x)/(1-\sqrt{1-x^2})$ as $x\to 0$, since this ratio is decreasing with x (where $\alpha(x)$ is defined as in the proof of Lemma B.1). To do that it suffices to use $\alpha(x)=x^2+O(x^4)$ and $1-\sqrt{1-x^2}=x^2/2+O(x^4)$, so the limit is 2. Hence the limit of $(1-H((1+x)/2))/(1-\sqrt{1-x^2})$ as $x\to 0$ is $1/\ln 2$, meaning this ratio is always at most $1/\ln 2$. Similarly, to show the inequality $\mathrm{S}^2\le(\ln 4)\,\mathrm{JS}$, we only need to compute the limit of $\alpha(x)/x^2$ as $x\to 0$. Again using $\alpha(x)=x^2+O(x^4)$, the limit is 1, so the ratio $(1-H((1+x)/2))/x^2$ is always at least $1/\ln 4$. □

Lemma 3.11. *If $x\in[0,1]$ and $k\in[1,\infty)$, we have*

$$\frac{1}{2}\min\{kx,1\}\le 1-(1-x)^k\le\min\{kx,1\}.$$

Proof. Set $f(x) \coloneqq 1-(1-x)^k$. Clearly, when $x \in [0,1]$, we have $f(x) \in [0,1]$, so $f\colon [0,1] \to [0,1]$. Note $f(0)=0$, $f(1)=1$, and that $f(x)$ is increasing on $[0,1]$. If $k=1$, we have $f(x)=x$, and the inequalities trivially hold; therefore, assume $k>1$. Then $f'(x)=k(1-x)^{k-1}$ and $f''(x)=-k(k-1)(1-x)^{k-2}$, meaning that $f(x)$ is concave on $[0,1]$; we also have $f'(0)=k$ and $f''(0)=-k(k-1)$. From this we conclude that $f(x) \le kx$, proving the upper bound (as $f(x) \le 1$ is clear).

For the lower bound, note that $f'''(x)=k(k-1)(k-2)(1-x)^{k-3}$, which is non-negative on $[0,1]$. This means that $f''(x) \ge -k(k-1)$ on $[0,1]$, that $f'(x) \ge k-k(k-1)x$ on $[0,1]$, and that $f(x) \ge kx-(k(k-1)/2)x^2 = kx(1-(k-1)x/2)$ on $[0,1]$. If $(k-1)x \le 1$, we get $f(x) \ge kx/2$. If $(k-1)x \ge 1$, we have $f(x) \ge 1-e^{-kx} \ge 1-1/e \ge 1/2$. This completes the proof. □

Lemma 4.4 (Hellinger distance of disjoint mixtures)**.** *Let μ be a distribution over a finite support A, and for each $a \in A$, let ν_0^a and ν_1^a be two distributions over a finite support S_a. Let ν_0^μ and ν_1^μ denote the mixture distributions where $a \leftarrow \mu$ is sampled, and then a sample is produced from ν_0^a or ν_1^a respectively. Assume the sets S_a are disjoint for all $a \in A$. Then*

$$\mathrm{h}^2(\nu_0^\mu, \nu_1^\mu) = \mathbb{E}_{a \leftarrow \mu}[\mathrm{h}^2(\nu_0^a, \nu_1^a)].$$

Proof. Note that the squared-Hellinger distance is one minus the fidelity, that is, $\mathrm{h}^2(\mu_1,\mu_2) = 1-F(\mu_1,\mu_2)$ where $F(\mu_1,\mu_2)=\sum_x \sqrt{\mu_1[x]\mu_2[x]}$ (this is easy to check from the definition of h^2). Now write

$$\begin{aligned}
\mathrm{h}^2(\nu_0^\mu, \nu_1^\mu) &= 1 - \sum_{x \in \bigcup_a S_a} \sqrt{\nu_0^\mu[x]\nu_1^\mu[x]} \\
&= 1 - \sum_{a \in A} \sum_{x \in S_a} \sqrt{\mu[a]\nu_0^a[x]\mu[a]\nu_1^a[x]} \\
&= 1 - \mathbb{E}_{a \leftarrow \mu} \left[\sum_{x \in S_a} \sqrt{\nu_0^a[x]\nu_1^a[x]} \right] \\
&= \mathbb{E}_{a \leftarrow \mu} \left[1 - \sum_{x \in S_a} \sqrt{\nu_0^a[x]\nu_1^a[x]} \right] \\
&= \mathbb{E}_{a \leftarrow \mu} \left[\mathrm{h}^2(\nu_0^a, \nu_1^a) \right].
\end{aligned}$$

□

C Quantum amplitude estimation

We show the following strengthening of Theorem 5.1, which follows from [BHMT02].

Theorem C.1 (Amplitude estimation)**.** *Suppose we have access to a unitary U (representing a quantum algorithm) which maps $|0\rangle$ to $|\psi\rangle$, as well as access to a projective measurement Π, and we wish to estimate $p \coloneqq \|\Pi|\psi\rangle\|_2^2$ (representing the probability the quantum algorithm accepts). Fix $\epsilon, \delta \in (0, 1/2)$. Then using at most $(100/\epsilon)\cdot\ln(1/\delta)$ controlled applications of U or $U^\dagger$ and at most that many applications of $I-2\Pi$, we can output $\tilde{p} \in [0,1]$ such that $|\tilde{p}-p| \le \epsilon$ with probability at least $1-\delta$.*

Further, this can be tightened to a bound that depends on p, as follows. For any positive real number T, there is an algorithm which depends on ϵ, δ, and T (but not on p) which uses at most T applications of the unitaries (as above) and outputs $\tilde{p} \in [0,1]$ with the following guarantee: if T is at least $\lfloor (100/\epsilon)\sqrt{\max\{p,\epsilon\}}\cdot\ln(1/\delta)\rfloor$, then $|\tilde{p}-p| \le \epsilon$ with probability at least $1-\delta$.

Proof. [BHMT02] showed that an algorithm which makes M controlled calls to the unitary $U(I - 2|0\rangle\langle 0|)U^{-1}(I - 2\Pi)$ and one additional call to U can output $\tilde{p}$ such that

$$|\tilde{p} - p| \leq \frac{2\pi\sqrt{p(1-p)}}{M} + \frac{\pi^2}{M^2}$$

with probability at least $8/\pi^2 \geq 4/5$. If we pick M such that $M \geq 8/\sqrt{\epsilon}$ and $M \geq 8\sqrt{p}/\epsilon$, then this is at most $(\pi/4 + \pi^2/64)\gamma \leq \gamma$. Note that M must be an integer, and that the number of applications of U or U^{-1} is $2M + 1$. Hence to get this success probability, it suffices to have $T \geq 3 + (16/\epsilon)\sqrt{\max\{p, \epsilon\}}$, or $T \geq (19/\epsilon)\sqrt{\max\{p, \epsilon\}}$.

To generalize to other success probabilities, we amplify this algorithm by repeating $2k+1$ times and returning the median estimate. The probability that this is still wrong is the probability that at least $k + 1$ out of $2k + 1$ of the estimates were wrong, which is

$$\sum_{i=1}^{k+1} \binom{2k+1}{k+1-i} q^{k+i}(1-q)^{k+1-i} \leq q^{k+1}(1-q)^k \sum_{i=1}^{k+1} \binom{2k+1}{k+1-i}$$

$$= q^{k+1}(1-q)^k 2^{2k} = q(1-(1-2q)^2)^k \leq qe^{-k(1-2q)^2}.$$

Hence to get this below δ, we just need $k \geq (1/(1-2q)^2)\ln(1/q\delta)$, or $k \geq 2.6\ln(1/\delta) - 4$. Since k must be an integer, but we can always choose it so that $2k+1$ is at most $5.2\ln(1/\delta)$. Multiplying this by the bound from before, we get that it suffices for T to be at most $(100/\epsilon)\sqrt{\max\{p, \epsilon\}} \cdot \ln(1/\delta)$, as desired. □

References

[Alt88] Helmut Alt. Comparing the combinational complexities of arithmetic functions. *Journal of the ACM* (1988). DOI: 10.1145/42282.214084 (p. 49).

[AR20] Scott Aaronson and Patrick Rall. Quantum Approximate Counting, Simplified. *Proceedings of the 3rd Symposium on Simplicity in Algorithms (SOSA)*. 2020. DOI: 10.1137/1.9781611976014.5. arXiv: 1908.10846 (p. 33).

[BB19] Eric Blais and Joshua Brody. Optimal Separation and Strong Direct Sum for Randomized Query Complexity. *Proceedings of the 34th Conference on Computational Complexity (CCC)*. 2019. DOI: 10.4230/LIPICS.CCC.2019.29. arXiv: 1908.01020 (p. 5).

[BB20] Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions. *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2020. arXiv: 2002.10809 (pp. 4, 7, 8).

[BBGK18] Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. Classical Lower Bounds from Quantum Upper Bounds. *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2018. DOI: 10.1109/focs.2018.00040. arXiv: 1807.06256 (p. 44).

[BCH86] Paul W. Beame, Stephen A. Cook, and H. James Hoover. Log Depth Circuits for Division and Related Problems. *SIAM Journal on Computing* (1986). Previous version in FOCS 1984. DOI: 10.1137/0215070 (p. 49).

[BDG+20] Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. The Power of Many Samples in Query Complexity. *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2020. DOI: 10.4230/LIPIcs.ICALP.2020.9. arXiv: 2002.10654 (pp. 8, 52).

[BGK+18] Mark Braverman, Ankit Garg, Young Kun Ko, Jieming Mao, and Dave Touchette. Near-Optimal Bounds on the Bounded-Round Quantum Communication Complexity of Disjointness. *SIAM Journal on Computing* (2018). Previous version in FOCS 2015. DOI: 10.1137/16m1061400. arXiv: 1505.03110 (p. 5).

[BHK09] Boaz Barak, Moritz Hardt, and Satyen Kale. The Uniform Hardcore Lemma via Approximate Bregman Projections. *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*. 2009. DOI: 10.1137/1.9781611973068.129 (p. 8).

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Proceedings of an AMS Special Session on Quantum Computation and Information (CONM)*. 2002. DOI: 10.1090/conm/305/05215. arXiv: quant-ph/0005055 (pp. 33, 58, 59).

[BNRW07] Harry Buhrman, Ilan Newman, Hein Rohrig, and Ronald de Wolf. Robust Polynomials and Quantum Algorithms. *Theory of Computing Systems* (2007). Previous version in STACS 2005. DOI: 10.1007/s00224-006-1313-z. arXiv: quant-ph/0309220 (p. 41).

[Bra15] Mark Braverman. Interactive Information Complexity. *SIAM Journal on Computing* (2015). Previous version in STOC 2012. DOI: 10.1137/130938517 (p. 5).

[BSS05] Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. Preprint, 2005. URL: pdfs.semanticscholar.org/d670/6b6e626c15680688b0774419662f2341caee.pdf (pp. 5, 20).

[CSV84] Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. Constant Depth Reducibility. *SIAM Journal on Computing* (1984). DOI: 10.1137/0213028 (p. 50).

[GKKT17] Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably Learning the ReLU in Polynomial Time. *Proceedings of the 30th Annual Conference on Learning Theory (COLT)*. 2017. arXiv: 1611.10258 (p. 41).

[GR07] Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association* (2007). DOI: 10.1198/016214506000001437 (p. 5).

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 1995. DOI: 10.1109/sfcs.1995.492584 (pp. 5, 8).

[Jac11] Dunham Jackson. "Über die Genauigkeit der Annäherung stetiger Funktionen durch ganze rationale Funktionen gegebenen Grades und trigonometrische Summen gegebener Ordnung". PhD thesis. University of Göttingen, 1911. URL: gdz.sub.uni-goettingen.de/id/PPN30230648X (p. 41).

[KS03] Adam R. Klivans and Rocco A. Servedio. Boosting and Hard-Core Set Construction. *Machine Learning* (2003). Previous version in FOCS 1999. DOI: 10.1023/a:1022949332276 (p. 8).

[LS09] Troy Lee and Adi Shraibman. An Approximation Algorithm for Approximation Rank. *Proceedings of the 24th Conference on Computational Complexity (CCC)*. 2009. DOI: 10.1109/ccc.2009.25. arXiv: 0809.2093 (p. 46).

[LSŠ08] Troy Lee, Adi Shraibman, and Robert Špalek. A Direct Product Theorem for Discrepancy. *Proceedings of the 23rd Conference on Computational Complexity (CCC)*. 2008. DOI: 10.1109/ccc.2008.25 (p. 44).

[MCAL17] Marianthi Markatou, Yang Chen, Georgios Afendras, and Bruce G. Lindsay. Statistical Distances and Their Role in Robustness. *New Advances in Statistics and Data Science* (2017). DOI: 10.1007/978-3-319-69416-0_1. arXiv: 1612.07408 (p. 21).

[MMR94] G. V. Milovanovic, D. S. Mitrinovic, and Th. M. Rassias. *Topics in Polynomials: Extremal Problems, Inequalities, Zeros*. World Scientific, 1994. ISBN: 978-981-02-0499-0. DOI: 10.1142/1284 (p. 41).

[Ofm62] Yuri P. Ofman. On the algorithmic complexity of discrete functions. *Doklady Akademii Nauk* (1962) (p. 49).

[Pip87] Nicholas Pippenger. The complexity of computations by networks. *IBM Journal of Research and Development* (1987). DOI: 10.1147/rd.312.0235 (p. 49).

[RT92] John H. Reif and Stephen R. Tate. On Threshold Circuits and Polynomial Computation. *SIAM Journal on Computing* (1992). DOI: 10.1137/0221053 (p. 50).

[RW11] Mark D. Reid and Robert C. Williamson. Information, Divergence and Risk for Binary Experiments. *Journal of Machine Learning Research* (2011). arXiv: 0901.0356. URL: http://jmlr.org/papers/v12/reid11a.html (p. 21).

[Sha03] Ronen Shaltiel. Towards proving strong direct product theorems. *Computational Complexity* (2003). Previous version in CCC 2001. DOI: 10.1007/s00037-003-0175-x. ECCC: 2001/009 (p. 4).

[She12] Alexander A. Sherstov. Strong Direct Product Theorems for Quantum Communication and Query Complexity. *SIAM Journal on Computing* (2012). Previous version in STOC 2011. DOI: 10.1137/110842661. arXiv: 1011.4935 (p. 44).

[She13] Alexander A. Sherstov. Making Polynomials Robust to Noise. *Theory of Computing* (2013). Previous version in STOC 2012. DOI: 10.4086/toc.2013.v009a018. ECCC: 2012/037 (pp. 41, 46).

[Sio58] Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics* (1958). DOI: 10.2140/pjm.1958.8.171 (pp. 5, 14, 55).

[Tøp00] Flemming Tøpsoe. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory* (2000). DOI: 10.1109/18.850703 (p. 21).

[TTV09] Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. Regularity, Boosting, and Efficiently Simulating Every High-Entropy Distribution. *Proceedings of the 24th Conference on Computational Complexity (CCC)*. 2009. DOI: 10.1109/ccc.2009.41. ECCC: 2008/103 (p. 8).

[Ver98] Nikolai K. Vereshchagin. Randomized Boolean decision trees: Several remarks. *Theoretical Computer Science* (1998). DOI: 10.1016/S0304-3975(98)00071-1 (pp. 5, 10).

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer Berlin Heidelberg, 1999. ISBN: 978-3-642-08398-3. DOI: 10.1007/978-3-662-03927-4 (p. 50).

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. Wiley, 1987. ISBN: 3-519-02107-2. URL: eccc.weizmann.ac.il/static/books/The_Complexity_of_Boolean_Functions/ (pp. 49, 50).

[Yao77] Andrew Yao. Probabilistic computations: toward a unified measure of complexity. *Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (1977). DOI: 10.1109/SFCS.1977.24 (pp. 3, 5).