Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 185 (2020)

Quantum learning algorithms imply circuit lower bounds

Srinivasan Arunachalam
IBM T. J. Watson Research Center
srinivasan.arunachalam@ibm.com

Alex B. Grilo
Sorbonne Université, CNRS, LIP6
Alex.Bredariol-Grilo@lip6.fr

Tom Gur
University of Warwick
tom.gur@warwick.ac.uk

Igor C. Oliveira
University of Warwick
igor.oliveira@warwick.ac.uk

Aarthi Sundaram
Microsoft Quantum
aarthi.sundaram@microsoft.com

November 19, 2021

Abstract

We establish the first general connection between the design of quantum algorithms and circuit lower bounds. Specifically, let $\mathfrak{C}$ be a class of polynomial-size concepts, and suppose that $\mathfrak{C}$ can be PAC-learned with membership queries under the uniform distribution with error $1/2-\gamma$ by a time T quantum algorithm. We prove that if $\gamma^2 \cdot T \ll 2^n/n$, then $\mathsf{BQE} \nsubseteq \mathfrak{C}$, where $\mathsf{BQE} = \mathsf{BQTIME}[2^{O(n)}]$ is an exponential-time analogue of BQP. This result is optimal in both γ and T, since it is not hard to learn any class $\mathfrak{C}$ of functions in (classical) time $T = 2^n$ (with no error), or in quantum time $T = \mathsf{poly}(n)$ with error at most $1/2 - \Omega(2^{-n/2})$ via Fourier sampling. In other words, even a marginal quantum speedup over these generic learning algorithms would lead to major consequences in complexity lower bounds. As a consequence, our result shows that the study of quantum learning speedups is intimately connected to fundamental open problems about algorithms, quantum computing, and complexity theory.

Our proof builds on several works in learning theory, pseudorandomness, and computational complexity, and on a connection between non-trivial classical learning algorithms and circuit lower bounds established by Oliveira and Santhanam (CCC 2017). Extending their approach to quantum learning algorithms turns out to create significant challenges, since extracting computational hardness from a quantum computation is inherently more complicated. To achieve that, we show among other results how pseudorandom generators imply learning-to-lower-bound connections in a generic fashion, construct the first conditional pseudorandom generator secure against uniform quantum computations, and extend the local list-decoding algorithm of Impagliazzo, Jaiswal, Kabanets and Wigderson (SICOMP 2010) to quantum circuits via a delicate analysis. We believe that these contributions are of independent interest and might find other applications.

ISSN 1433-8092

Contents

1 Introduction

One of the salient goals of quantum computing is to understand which computational problems admit quantum speedups over classical algorithms. The canonical example is Shor's algorithm [Sho94] for factoring, which achieves an exponential speedup over the best known classical algorithms. Another notable example is Grover's algorithm [Gro96], which sheds light on quantum complexity theory by showing that expressive languages such as the quintessential NP-complete problem Formula-SAT admits quantum algorithms of time complexity $\widetilde{O}(2^{n/2})$, whereas in general, there are no known classical algorithms that outperform the $\widetilde{O}(2^n)$-time brute force search. This paper investigates the implications of quantum speedups within the setting of *learning theory*.

Quantum learning theory forms the theoretical foundations which allow us to understand the potential power and limitations of quantum machine learning. At its core, this field studies quantum algorithms that are given quantum access (typically, quantum queries or quantum samples) to an unknown circuit f from a fixed concept class $\mathfrak{C}$, and the goal is to output a hypothesis h that well-approximates f, in which case we say that $\mathfrak{C}$ can be quantumly learned. Due to the massive success of machine learning and the great potential of quantum computing, quantum learning received much attention over the last two decades [BJ98, SG04, AS05, ABG06, AS07, OW16, OW17, AW18, GKZ19, ACL+19] (cf. survey [AW17] and references therein).

In this setting, quantum algorithms have two main advantages over their classical counterparts: making queries in superposition, and using quantum computation to process the information obtained from these queries. Note that a large number of negative results about the power of *classical* learning algorithms do not extend to the quantum setting (e.g., [Kha93, NR99]), since the underlying hardness assumptions, based on problems such as factoring and discrete logarithm, break for quantum computations. In addition, in some learning models and for some learning tasks, quantum algorithms are strictly faster than classical algorithms [SG04], under standard cryptographic assumptions. While it is possible to rule out polynomial-time or quasi-polynomial time quantum learning algorithms for some concept classes using stronger cryptographic assumptions [AGS21], our understanding of the possibilities and limitations of *sub-exponential* time quantum learnability is still limited. This motivates the following fundamental question:

Are quantum speedups for learning expressive concept classes possible?

Our main result exposes an intrinsic connection between complexity theory and quantum learning theory, showing that obtaining any quantum learner that does marginally better than certain "trivial" learners (see Section 1.1) would imply circuit lower bounds for languages computable in BQE, the quantum analogue of E.[1] To our knowledge, this is the first result connecting the design of general quantum algorithms to proving lower bounds:

Main result (Informal)**.** *If a class $\mathfrak{C}$ of polynomial-size concepts can be learned under the uniform distribution with membership queries and with error at most $1/2-\gamma$ in quantum time $o(\gamma^2 \cdot 2^n/n)$, then* $\mathsf{BQE} \not\subseteq \mathfrak{C}$.

While it seems extremely unlikely that a large class such as BQE can be simulated by classical Boolean circuits of polynomial size, showing this and similar results for exponential time classes seems to be out of reach of current techniques. For comparison, the recent $\mathsf{NE} \not\subseteq \mathsf{ACC}^0$ lower bound due to Williams [Wil14] is widely recognized to be a milestone in complexity theory.

Our result admits two possible interpretations. For a pessimist, it explains the difficulty of designing provably correct quantum learning algorithms for expressive concept classes, given that

[1] Recall that $\mathsf{E} = \mathsf{DTIME}[2^{O(n)}]$, and analogously $\mathsf{BQE} = \mathsf{BQTIME}[2^{O(n)}]$.

establishing non-uniform circuit lower bounds is a notoriously difficult problem. Contrarily, for an optimist, it indicates a potential path to new lower bounds by exploring the power of quantum computation. For instance, if depth-2 threshold circuits of polynomial size can be learned in $o(2^n/n)$ quantum time under the uniform distribution, a new complexity lower bound would follow.

Our starting point in the proof of this result is a connection of a similar nature between classical learning algorithms and circuit lower bounds due to Oliveira and Santhanam [OS17]. The extension to quantum learning algorithms turns out to require significant technical work. En route to that, we obtain new results concerning local list-decoding for quantum circuits, construct the first PRG secure against uniform quantum computations, and provide a new general method to establish learning-to-lower-bound connections. We next describe our contributions in more detail.

1.1 Main result

Before we proceed to formally state our result, we first discuss the model of quantum learning and provide a brief overview of circuit lower bounds in complexity theory.

Quantum learning. We consider the standard PAC learning model under the uniform distribution with quantum membership queries. Our main result derives a consequence from the *existence* of learning algorithms, and restricting our model to the uniform distribution and allowing queries only makes it stronger (since learning algorithms are easier to design under these assumptions). Although we consider quantum learning algorithms, we emphasize that our results are concerned with the learnability of (classical) *Boolean functions* $f\colon \{0,1\}^n \to \{0,1\}$ from a fixed concept class $\mathfrak{C}$. Here, we use $\mathfrak{C}[s(n)]$ to refer to concepts defined over n-bit inputs and of size at most $s(n)$.

In more detail, a quantum learning algorithm $\mathcal{A}$ for $\mathfrak{C}$ running in time T is described by a uniform sequence of quantum oracle circuits Q_n of size at most $T(n)$. We say that $\mathcal{A}$ learns $\mathfrak{C}[s]$ with probability δ and error ε if for every n and every $f \in \mathfrak{C}[s(n)]$, Q_n with oracle access to f outputs with probability at least δ the description of a (classical) Boolean circuit C such that $\Pr_x[C(x) \neq f(x)] \leq \varepsilon$. Note that having oracle access to f means that Q_n can query f in superposition via a unitary map O_f whose action on basis states is defined as $O_f : |x, b\rangle \to |x, b \oplus f(x)\rangle$, for every $x \in \{0,1\}^n$ and $b \in \{0,1\}$. We stress here that we do not require the hypothesis circuit C to be from the same class, i.e, the learning algorithm can be improper (also known as representation independent). (Our result also holds for learners that output a *quantum circuit* that approximates f on average to error at most ε, and we refer to the body of the paper for details.)

It is instructive to contrast our main result with two learners that are "trivial" in a certain sense, i.e., they do not exploit the structure of the concept class. The first one is simply a brute-force (classical) learner that queries all possible inputs of the unknown function, and outputs a hypothesis consisting of its truth-table. This learner can be implemented in time $T = \widetilde{O}(2^n)$ and achieves optimal error parameter $\varepsilon = 0$. On the other hand, the second algorithm explores the basic fact that any function $f\colon \{0,1\}^n \to \{0,1\}$ can be approximated by a parity function (or its negation) with advantage $\gamma = \Omega(2^{-n/2})$ (i.e., with error $\varepsilon \leq 1/2 - \Omega(2^{-n/2})$). For this reason, any class $\mathfrak{C}$ of Boolean concepts can be learned with probability $\delta = \Omega(1)$ and error $\varepsilon \leq 1/2 - \Omega(2^{-n/2})$ using Fourier sampling in time $T = \mathsf{poly}(n)$ (see Appendix A for details). (We stress that this highly efficient learner is only available in the quantum setting, thanks to Fourier sampling.)

Complexity Theory. Establishing (non-uniform) circuit lower bounds is one of the holy grails of the theory of computation. Despite over 50 years of extensive research, we still have a very poor understanding of the limitations of Boolean circuits. While significant progress has been made with

respect to very restricted circuit models, such as small-depth circuits with OR, AND, and NOT gates, the power of more expressive circuit classes remains mysterious.

To give two concrete examples, firstly, we don't know how to rule out that every algorithm running in time $2^{O(n)}$ can be simulated by (classical) Boolean circuits of linear size. Secondly, as mentioned above, it was not long ago that Williams [Wil14] obtained the first separation between non-deterministic exponential time $\mathsf{NE} = \mathsf{NTIME}[2^{O(n)}]$ and the class $\mathsf{ACC}^0[m]$ of polynomial-size constant-depth circuits consisting of AND, OR, NOT, and MOD_m gates. This illustrates how difficult it is to understand the power of *non-uniform* computations, even in the setting of exponential time classes such as E, NE and BQE.[2]

Indeed, while the possibility that a large class such as BQE can be simulated by classical Boolean circuits of linear size seems unlikely, proving such statements remains notoriously hard. Given the scarcity of techniques to establish such complexity separations, it is of interest to obtain new approaches for circuit lower bounds and to connect this question to other research areas in algorithms and complexity.

In the following, we restrict our attention to reasonable classes $\mathfrak{C}$ of circuits that can be efficiently simulated by general Boolean circuits and are closed under restrictions, i.e., if f is computable by $\mathfrak{C}$-circuits of size $s(n)$, then f is computable by general Boolean circuits of size $\mathsf{poly}(s(n))$, and the function obtained by restricting some of the variables of f to constants 0 and 1 is also computable by $\mathfrak{C}$-circuits of size $s(n)$. Note that virtually any circuit class of interest, including depth-2 threshold circuits, ACC^0, and polynomial-size formulas, satisfies these properties.

Theorem 1 (Non-trivial quantum learning algorithms yield non-uniform complexity lower bounds)**.** *There is a universal constant $\lambda \geq 1$ for which the following holds. Let $\mathfrak{C}$ be a concept class. Let $\gamma\colon \mathbb{N} \to [0, 1/2]$ and $T : \mathbb{N} \to \mathbb{N}$ satisfy $\gamma(n) \geq \lambda \cdot 2^{-n/2}$ and $T(n) \leq \gamma(n)^2 \cdot 2^n/\lambda n$. Suppose that, for every $k \geq 1$, the class $\mathfrak{C}[n^k]$ can be learned in quantum time $T(n)$ with probability $\geq 1/100$ and error $\varepsilon \leq 1/2 - \gamma(n)$. Then, for every $k \geq 1$, we have $\mathsf{BQE} \not\subseteq \mathfrak{C}[n^k]$.*

The confidence probability $1/100$ is not essential, and we adopted this constant in order to simplify the statement and focus on the trade-off between running time and accuracy. Two interesting settings of parameters are (1) $\gamma(n) = 0.49$ and $T(n) = o(2^n/n)$, and (2) $\gamma(n) = n^{\omega(1)} \cdot 2^{-n/2}$ and $T(n) = n^{\omega(1)}$. The first case shows that strong learners that beat the trivial brute-force learning algorithm by a polynomial factor (with respect to the running time) imply lower bounds, while the second setting shows that polynomial-time learners that perform marginally better than a Fourier-sampling based learner (with respect to the error parameter) also imply lower bounds. For this reason, we view Theorem 1 as a result essentially stating that *non-trivial quantum learnability of a class of polynomial-size circuits yields complexity lower bounds.*

As alluded to above, the connection established in Theorem 1 can be interpreted in two ways. On the one hand, it provides an explanation for why it is difficult to design provably correct non-trivial quantum learners, as they would imply dramatic consequences to complexity theory, showing new circuit lower bounds that are notoriously hard to prove. On the other hand, this connection significantly strengthens the paradigm of proving circuit lower bounds via (classical)

[2]Note that existing circuit lower bounds against circuits of size n^k, such as the result from [San09] showing that $\mathsf{MA}/1 \not\subseteq \mathsf{SIZE}[n^k]$, do not provide a hard language in BQE. The naive simulation of a language $L \in \mathsf{MA}$ in exponential time results in an algorithm that runs in time 2^{n^c} for some constant $c \geq 1$ that can be larger than k (and we can easily diagonalize againts circuits of size n^k in time 2^{n^c} when $c > k$). Additionally, we do not consider in this paper the computation of a hard problem with advice, as in the $\mathsf{MA}/1$ lower bound, which requires 1 bit of non-uniform advice.

learning algorithms [OS17] by capitalizing on the power of quantum learning algorithms, which might be vastly stronger than their classical counterparts.[3]

In the subsequent section, we discuss our techniques in detail and explain our additional contributions.

1.2 Techniques

The first results showing that learning algorithms imply lower bounds appeared in the pioneering work of Fortnow and Klivans [FK09]. These initial results, however, required a strong assumption on the resources of the learning algorithm. For instance, for randomized learners, lower bound consequences could only be obtained from *polynomial-time* learners. Different learning assumptions were explored in a sequence of subsequent works [HH13, KKO13, Vol14, Vol16, OS17, OS18, CRTY20]. We review these works in detail in Section 1.4, where we present a self-contained exposition of existing connections between learning algorithms and lower bounds. Here we focus on one of the strongest (and most relevant in our context) connections obtained before this work.

Oliveira and Santhanam [OS17, Theorem 14] showed that if a class $\mathcal{C}$ of polynomial-size concepts can be PAC-learned under the uniform distribution to error $\varepsilon \leq 1/2 - \gamma$ by a *randomized* algorithm running in time $\gamma^2 \cdot 2^n/n^{\omega(1)}$, then for every $k \geq 1$ we have $\mathsf{BPE} \not\subseteq \mathcal{C}[n^k]$. In contrast, Theorem 1 can be seen as an analogue of this connection in the *quantum* setting, where algorithms can be more powerful. Intuitively, this means that the task of extracting a computational lower bound from a learning algorithm becomes more delicate. (Indeed, as mentioned above, there is even another "trivial" learner that is only available in the quantum setting and proceeds via Fourier sampling.) To accomplish that, we make additional conceptual and technical contributions of independent interest:

- Our paper introduces a *new paradigm* to establish learning-to-lower-bound connections that employs a pseudorandom generator (PRG) in a black-box way. Thus, even without its quantum counterpart, we simplify and extend existing results.
- We propose and analyze the *first* PRG with sub-exponential stretch that is secure against uniform quantum computations. Moreover, we base its security on a weaker *uniform* hardness assumption.
- We prove a near-optimal uniform hardness amplification result for quantum circuits by a delicate extension of techniques and analysis from Impagliazzo, Jaiswal, Kabanets, and Wigderson [IJKW10].
- We introduce a *new computational model* between classical and quantum computation: *inherently probabilistic circuits*. It highlights an important difference between classical and quantum algorithms, and provides a way to mitigate the complex task of analyzing quantum computations.

In the next sections, we provide more details about the proof of Theorem 1, explain the role of the contributions mentioned above, and contrast our techniques to prior work.

[3] Even if in the short term we are not able to design new quantum learning algorithms, the mere existence of a *connection* between learning and lower bounds has been used to establish *unconditional* complexity lower bounds (see Section 1.4 and [Oli19, Section 3.1]). Thus, if the "pessimist" interpretation is the correct one, it is still possible that the connection established in Theorem 1 can be indirectly used as a key ingredient of a lower bound proof.

1.2.1 The classical proof and our new perspective

Before discussing the quantum perspective, it is instructive to review the approach for showing the classical connection between randomized learners and lower bounds.

Randomized learners. Techniques from computational complexity theory and from the theory of pseudorandomness play a key role in the proof from [OS17] that non-trivial randomized learners yield complexity lower bounds. In a bit more detail, their argument proceeds roughly as follows:

1. First, they show that *sub-exponential* time randomized learners for a class $\mathfrak{C}$ imply that $\mathsf{BPE} \not\subseteq \mathfrak{C}$. This part refines ideas from [FK09, KKO13] that rely on results from structural complexity theory (e.g., a special PSPACE-complete language [TV07] and diagonalization).

2. This is followed by a proof that the existence of "*non-trivial*" randomized learners for a class $\mathfrak{C}$ of polynomial-size concepts implies the existence of *sub-exponential* time randomized learners for $\mathfrak{C}$. This implication relies in part on connections between learning theory and the theory of pseudorandomness introduced by [CIKK16].

The proof of the learning-to-lower-bound connection immediately follows from Items 1 and 2 above.

The quantum case: A new perspective. In contrast to [OS17], here we take a more direct path to show that "non-trivial" quantum learning algorithms for $\mathfrak{C}$ imply lower bounds against $\mathfrak{C}$. While several technical tools from the theory of pseudorandomness that we use still correspond to quantum extensions of core results behind [OS17], our approach is conceptually very different. In more detail, we are able to show that PRGs against *uniform* (classical or quantum) computations can be used to establish a learning-to-lower-bound connection in a *black-box* way. In particular, we do not follow the 2-step approach outlined above.

The benefits of our new perspective are twofold: (a) on the one hand, stronger PRG statements immediately lead to stronger connections between learning algorithms and lower bounds, and (b) it allows for a more modular proof of the learning-to-lower-bound connection. In particular, with our perspective the argument becomes more manageable in the quantum setting, where new technical difficulties are present compared with the randomized case.

At a very high level, we use a PRG to generate a "hard" function that is not correctly learned by the quantum learning algorithm. Consequently, this function does not belong to the circuit class $\mathfrak{C}$, and it can be used to define a language that cannot be computed by $\mathfrak{C}$-circuits of bounded size. What makes the approach viable is that a PRG that fools *uniform* computations is sufficient. We discuss our proof in more detail in Section 1.2.3. Before that, at a more conceptual level, we explain some of the challenges associated with the transition from randomized to quantum computations.

1.2.2 Challenges in the quantum setting

Our goal in Theorem 1 is to show that the existence of a quantum learning algorithm for a class $\mathfrak{C}$ of polynomial-size circuits can be used to construct a function $h \in \mathsf{BQE}$ that cannot be computed by $\mathfrak{C}$-circuits of size n^k. As mentioned above, and explained in more detail in Section 1.2.3 below, this will be achieved through the design of a PRG against uniform quantum computations, along with other ideas. In more detail, we show that if a certain language L is hard for sub-exponential time uniform quantum computations, i.e., $L \notin \mathsf{BQTIME}[2^{n^\gamma}]$ for some $\gamma > 0$, then there is a generator $G\colon \{0,1\}^\ell \to \{0,1\}^m$ computable in deterministic time $2^{O(\ell)}$, where $\ell = \mathsf{poly}(n)$ and $m = 2^{n^{\Omega(1)}}$, that is able to "fool" uniform quantum circuits of size $\mathsf{poly}(m)$.

Generators of this form are know in the realm of classical computations (see [IW01, TV07, CRTY20]), i.e., under an appropriate hardness assumption against $\mathsf{BPTME}[2^{n^\gamma}]$, it is possible to fool uniform probabilistic computations running in time $\mathsf{poly}(m)$. The proof of these results, essentially, proceeds as follows. If there is a sequence of (uniform) circuits $\{C_m\}$ of bounded complexity that distinguish the m-bit output of G from a random m-bit string, then there is a faster uniform algorithm for the hard problem L, which leads to a contradiction. In order to quantize this argument, we aim to construct a PRG secure against uniform *quantum* circuits. This leads to the natural question: *What can go wrong in these classical proofs when each C_m is a quantum circuit?*

Randomized circuits versus quantum circuits. It turns out that there is a general property of probabilistic computations that is not available to quantum computations. To be precise, consider the standard model of randomized Boolean circuits, i.e., a Boolean circuit $C(x, y)$, where x is the input string and y is the random string. We say that C computes a Boolean function $g\colon \{0,1\}^m \to \{0,1\}^k$ on an input $x \in \{0,1\}^m$ if $\Pr_y[C(x, y) = g(x)] \geq 2/3$. Similarly, we can also discuss the correlation between C and g, captured by $\eta = \Pr_{x,y}[C(x, y) = g(x)]$. An extremely useful property of this model is that, by an averaging argument, there exists a fixed string y' such that $\Pr_x[C(x, y') = g(x)] \geq \eta$, i.e., there is a *deterministic* circuit $C_{y'}(x) = C(x, y')$ that correctly computes g on an η-fraction of inputs. This often allows one to reduce the analysis of randomized Boolean circuits to the deterministic case. Additionally, $C_{y'}$ "forces" $C(x, y)$ to commit to a single consistent output on every x, which can be relevant if C is used as a subroutine in other computations.

As a concrete example, note that this idea is crucial in the proof of $\mathsf{BPP} \subseteq \mathsf{P/poly}$, which is a simple combination of reducing the failure probability of a randomized circuit and fixing its random input. (In contrast, it is open if $\mathsf{BQP} \subseteq \mathsf{P/poly}$.) Importantly, while quantum computations share superficial similarities with the model of randomized computations, there is no obvious way of reducing a quantum computation to a distribution of "deterministic" quantum computations. This has an important effect on the design of algorithms as well as on their analysis, creating several difficulties in our proof. We explain one of the most significant of these below.

Uniform hardness amplification for quantum circuits. A core component in several PRG constructions is to amplify the hardness of a function $f\colon \{0,1\}^n \to \{0,1\}$ that is only assumed to be mildly hard. This is a standard element of PRGs that follow the hardness versus randomness framework of Nisan and Wigderson [NW94], and we need it here as well. To be precise, suppose we start with f that is weakly hard for quantum circuits of bounded size, i.e., for every circuit $\mathcal{A}$ we have

$$\mathop{\mathbb{E}}_{x\sim\{0,1\}^n}\left[\Pr_{\mathcal{A}}\left[\mathcal{A}(x) = f(x)\right]\right] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x\sim\{0,1\}^n}\left[\|\Pi_{f(x)}\mathcal{A}\,|x, 0^\ell\rangle\|^2\right] \leq 1 - \delta,$$

where $\delta = 1/n$ for instance. In the notation above, $\|\Pi_{f(x)}\mathcal{A}\,|x, 0^\ell\rangle\|^2$ is the probability of obtaining $f(x)$ when measuring the first qubit of the circuit $\mathcal{A}$ on input $|x, 0^\ell\rangle$, and we average its success probability over a random input x. Our goal is to define a function $g\colon \{0,1\}^{n'} \to \{0,1\}^k$ from f that is exponentially harder than f, i.e., for every quantum circuit $\mathcal{B}$ of bounded size, we have

$$\mathop{\mathbb{E}}_{y\sim\{0,1\}^{n'}}\left[\Pr_{\mathcal{B}}\left[\mathcal{B}(y) = g(y)\right]\right] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{y\sim\{0,1\}^{n'}}\left[\|\Pi_{g(y)}\mathcal{B}\,|y, 0^{\ell'}\rangle\|^2\right] \leq \varepsilon,$$

where $\varepsilon = 2^{-n^{\Omega(1)}}$ for instance. We have two additional requirements beyond the generalization from classical to quantum computation: (1) we need a hardness amplification scheme with a near-optimal setting of parameters (so that our PRG achieves good parameters), and (2) since we are

constructing a PRG under a *uniform* hardness assumption, it is also crucial to control the amount of "non-uniformity" in the argument that establishes the correctness of the construction (i.e., how a circuit $\mathcal{B}$ violating the conclusion can be used to construct a circuit $\mathcal{A}$ that violates the hypothesis).

Our main technical contribution is to prove a quantum analogue of the near-optimal (uniform) hardness amplification result from Impagliazzo, Jaiswal, Kabanets, and Wigderson [IJKW10] which, in the language of coding theory, can be interpreted as a local-list decoding algorithm for direct product codes. In other words, their work considers $g = f^k$, i.e., $g\colon \{0,1\}^{n'} \to \{0,1\}^k$ is defined as the computation of k independent copies of f (thus $n' = kn$, and one can think of $k = \mathsf{poly}(n)$).

The extension of [IJKW10] to quantum circuits is challenging for the following reason. In the classical setting, given (without loss of generality) a *deterministic* circuit B such that

$$\Pr_{y\sim\{0,1\}^{n'}}[B(y) = g(y)] > \varepsilon, \tag{1}$$

there is only one way to distribute the correlation between B and g: there is a set $S \subseteq \{0,1\}^{n'}$ such that B is correct on S and wrong on $\overline{S}$, where $|S| > \varepsilon \cdot 2^{n'}$. On the other hand, for a quantum circuit $\mathcal{B}$, the correlation between $\mathcal{B}$ and g can be distributed in an arbitrary way among the inputs. For instance, perhaps on every input string $y \in \{0,1\}^{n'}$, $\Pr_{\mathcal{B}}[\mathcal{B}(y) = g(y)] = 2\varepsilon$. Going beyond that, we might also have quantum circuits that "interpolate" between these two extreme examples in an arbitrary way. Unfortunately, as opposed to randomized circuits, there is no way for us to "fix the quantumness" of $\mathcal{B}$ and reduce the analysis to the deterministic case, i.e., to the simpler setting of Equation (1). (For instance, one could query $\mathcal{B}$ on each input only once, memorizing its output after that. This, however, would not provide a *fixed* hard language if we employ $\mathcal{B}$ as a sub-routine when extracting lower bounds from quantum learning algorithms. Hence this trick would not work.) This more general scenario in the quantum setting affects the original analysis from [IJKW10] and introduces several fundamental "asymmetries" in the argument.

An inherently random "channel" and a quantization of [IJKW10]. From the perspective of coding theory, what we are trying to achieve is local list-decoding in a setting where querying a coordinate (an input for $\mathcal{B}$) does not produce a deterministic outcome. It turns out that this problem can be investigated without reference to quantum circuits. To achieve that, we explore a computational model that captures the scenario described above: *inherently probabilistic circuits* (see Section 2.7 for the definition and discussion of related work). Roughly speaking, this is a model for randomized computations where "the random input cannot be fixed", i.e., the random input remains inaccessible and the randomized circuit is only accessed as a black-box. It can be seen as an intermediate model between classical randomized circuits and quantum circuits. Most importantly for us, this model captures nearly all the difficulties when quantizing building blocks of the PRG.

Employing an intricate analysis that extends [IJKW10], we are able to show that their local list decoding algorithm (with a natural modification) works in the setting of inherently probabilistic circuits, with a minor loss in the parameters. The result can then be translated to the setting of quantum computations without much effort, which provides the near-optimal uniform hardness amplification needed in our PRG construction. We refer the reader to Section 4.3 for the technical details. Having established the above, we are ready for a high-level proof overview of Theorem 1.

1.2.3 Overview of the proof of Theorem 1

Let $\mathfrak{C}$ be a circuit class, and suppose that polynomial-size concepts from $\mathfrak{C}$ can be learned with error parameter $\varepsilon \leq 1/2 - \gamma$ in quantum time $o(\gamma^2 \cdot 2^n/n)$. For a given $k \geq 1$, we argue that there

is a language $L \in \mathsf{BQE} = \mathsf{BQTIME}[2^{O(n)}]$ such that $L \notin \mathfrak{C}[n^k]$.

In contrast with known proofs of existing learning-to-lower-bound connections, our black-box "PRG-based" approach relies on the following basic principle from the theory of pseudorandomness: If we have a property $\mathcal{P}$ of objects that is "dense" in the set of all possible objects, and a PRG G that is able to "fool" a class of "tests" that contain $\mathcal{P}$ (e.g., when $\mathcal{P}$ is easy to compute and G fools predicates of bounded complexity), then some output of G must be an object that satisfies $\mathcal{P}$. For us, $\mathcal{P}$ is simply the class of functions that are not in $\mathfrak{C}[n^k]$. We would like to find a (fixed) function in $\mathcal{P}$ and compute according to it in $\mathsf{BQTIME}[2^{O(n)}]$. We start off with the "ideal" plan for the proof, then discuss difficulties when implementing this strategy and how to overcome them.

Ideal plan for the proof of Theorem 1:

1. From a quantum learning algorithm $\mathcal{A}$ that *finds* "structure", we get a (uniform) quantum algorithm $\mathcal{B}$ that *decides* the "absence of structure" on an input string $f \in \{0,1\}^{2^n}$. (Since a typical *random* string is "unstructured", a *pseudorandom* string is likely to be accepted by $\mathcal{B}$.)
2. Assuming that there exists a PRG $G\colon \{0,1\}^{O(n)} \to \{0,1\}^{2^n}$ secure against uniform quantum computations, we use it with $\mathcal{B}$ as a "test" to find a function $h_n\colon \{0,1\}^n \to \{0,1\}$ that is not in $\mathfrak{C}[n^k]$, i.e., a function that "lacks structure". Given an n-bit input x, the hard language L outputs $h_n(x)$.
3. We show that a pseudorandom generator G with the desired parameters and guarantees exists.

Intuitively, if we could implement these steps then L would indeed be a hard function, and the time complexity of L can be bounded using upper bounds on the complexities of computing G and $\mathcal{B}$.

Implementing Step 1: (promise) quantum natural properties. In order to formalize Step 1, we introduce the notion of (promise) *quantum natural properties*, a generalization of the central concept from [RR97] to quantum computations. Informally, a natural property (useful) against a class $\mathfrak{C}$ of functions is an algorithm B that: (a) rejects every function $g \in \mathfrak{C}$, when g is viewed as a 2^n-bit string; (b) the set of strings accepted by B is dense in $\{0,1\}^{2^n}$; and (c) B runs in time $\mathsf{poly}(2^n) = 2^{O(n)}$ on an input string $f \in \{0,1\}^{2^n}$. Due to these properties, algorithm B can be seen as a way to efficiently tell apart "structured" strings (those that encode functions from $\mathfrak{C}$) from a dense set of "unstructured" strings. A *quantum* natural property against $\mathfrak{C}$ is simply a natural property against $\mathfrak{C}$ that is decided by a quantum algorithm. To our knowledge, this is the first time that quantum natural properties are considered in the literature.

We show that quantum learners with parameters as in Theorem 1 imply the existence of quantum natural properties against $\mathfrak{C}[n^k]$. The argument follows an idea that has appeared in a few different works (e.g., [Oli13, Vol14, OS17, AGS21]): to test if an input string $f \in \{0,1\}^{2^n}$ is not in $\mathfrak{C}[n^k]$, one can simulate the learning algorithm $\mathcal{A}$ when its oracle computes according to f (now viewed as function), and accepts if the hypothesis output by $\mathcal{A}$ and f have agreement estimated to be less than $1/2 + \gamma$. For a function $f \in \mathfrak{C}[n^k]$, when $\mathcal{A}$ learns f its hypothesis will have a larger correlation with f, and so f is rejected. On the other hand, it is possible to prove that if $\mathcal{A}$ runs in quantum time $o(\gamma^2 \cdot 2^n/n)$, then a dense set of Boolean functions are not learned by $\mathcal{A}$ even with a large error $\varepsilon = 1/2 - \gamma$ (simply because such functions cannot be approximated by hypotheses of size $o(\gamma^2 \cdot 2^n/n)$). Consequently, any such function will pass the low-correlation test with high probability and be accepted, no matter how many times the learner is simulated.

We directly extend this idea to the quantum setting in Section 3.1 while also addressing a subtlety that arises in the case of randomized and quantum learners. The resulting quantum

algorithm $\mathcal{B}$ computing a natural property against $\mathfrak{C}[n^k]$ might accept some input strings $f \in \{0,1\}^{2^n}$ with a probability that is not bounded away from $1/2$. This happens because we cannot control the behavior of the algorithm $\mathcal{A}$ on functions near the "border" of $\mathfrak{C}[n^k]$ (i.e., those inputs that are not as hard as a "random" function but are not in $\mathfrak{C}[n^k]$). Hence, we only get a *promise* quantum natural property: we are guaranteed that strings from the dense set are accepted with high probability and strings corresponding to functions in $\mathfrak{C}[n^k]$ are rejected with high probability.

Challenges in implementing Step 2. Unfortunately, the strategy described in Step 2 is problematic for a number of reasons:

(*i*) The PRG G constructed in Step 3 requires a hardness assumption, while Step 2 needs a language L that is *unconditionally* not in $\mathfrak{C}[n^k]$. This can only be achieved if G provably works (i.e. it does not depend on an unproven assumption).

(*ii*) Since $\mathcal{B}$ only computes a *promise* quantum natural property, it is not immediately clear how to use the output of G and the test $\mathcal{B}$ to fix with high probability a *unique* "hard" function h_n. This is needed so the language L is well defined.

(*iii*) The (conditional) PRG G that we are able to construct in Step 3 is much weaker: it will stretch $\mathsf{poly}(n)$ bits to $m(n) = 2^{n^\lambda}$ bits for some $\lambda > 0$, and it is only guaranteed to fool uniform quantum computations of time $\mathsf{poly}(m)$ on infinitely many choices of n.

These issues create technical difficulties that are addressed in Section 3.2 with some modifications to our original plan, as we explain next.

Issue (*i*). To resolve this and relax the conditions on the PRG, we consider two possible scenarios:

(*a*) Classical computations performed in polynomial space can be simulated by quantum algorithms running in sub-exponential time $2^{n^{o(1)}}$.

(*b*) Item (*a*) does not hold, i.e., there is a language $L \in \mathsf{PSPACE}$ and $\alpha > 0$ such that $L \notin \mathsf{BQTIME}[2^{n^\alpha}]$.

While we cannot currently decide which of the two scenarios holds, we obtain lower bounds against $\mathfrak{C}[n^k]$ in *each scenario*, which is sufficient for the purpose of proving Theorem 1. (We note that such "win-win" arguments have appeared in many previous works, including [FK09, KKO13, OS17].) In more detail, if (*a*) holds, we employ standard diagonalization techniques from complexity theory to get a language $L \in \mathsf{PSPACE} \setminus \mathfrak{C}[n^k]$, then show that $L \in \mathsf{BQE}$ via a sub-exponential time quantum simulation that is granted to exist by assumption (*a*). We can therefore assume for the rest of the proof that (*b*) holds. In other words, we have a hardness hypothesis against quantum computations that we can hope to use to construct a pseudorandom generator.

Issue (*ii*). We fix this issue as follows. Suppose, for simplicity, that we did manage to construct a PRG $G\colon \{0,1\}^{O(n)} \to \{0,1\}^{2^n}$ of exponential stretch that is computable in time exponential in n. A 2^n bit string output by G can be seen as the truth table for a function h_n. Since each seed for G produces a corresponding function, G encodes a collection of functions. What we know from the pseudorandomness of G and the existence of the promise quantum natural property is that at least one of these functions must lie outside $\mathfrak{C}[n^k]$. Then, the hard language L could be defined by, say, the "first" of the h_n that lies outside of $\mathfrak{C}[n^k]$ – the set of "easy functions". However, finding the h_n that satisfies this condition in $\mathsf{BQTIME}[2^{O(n)}]$ remains problematic.[4] Instead, we define a

[4]This is because $\mathcal{B}$ computes a *promise* natural property, and we cannot easily estimate the exact probability that $\mathcal{B}$ accepts a string to consistently compute the "first" such string.

language L over $O(n)+n$ input bits that *simultaneously* computes according to all output functions of G. More precisely, the hard language L is defined over a pair of input strings w and x, where w is a seed for G of length $O(n)$, and x is an n-bit string. We then set $L(w,x) = f_w(x)$, where f_w is the function defined by $G(w) \in \{0,1\}^{2^n}$. As explained above, given that the PRG G is secure against the uniform quantum computation performed by $\mathcal{B}$, it must produce at least one string $y = G(w)$ encoding a function h_n which lies outside the set $\mathfrak{C}[n^k]$. Since L computes h_n when we fix its first input string to its corresponding string w, L cannot be computed by $\mathfrak{C}$-circuits of size n^k.

Issue (*iii*). Finally, this issue is handled using a more careful description of the hard language L. The proof makes use of the fact that we are able to get a quantum natural property against $\mathfrak{C}[n^d]$ for each fixed choice of d, since by assumption we have learning algorithms for arbitrarily large polynomial-size concepts from $\mathfrak{C}$. Thus, a potential loss in the stretch of the generator (from 2^n to just 2^{n^λ} for some $\lambda > 0$) can be compensated by considering a natural property against harder functions, together with standard translation arguments from complexity theory. The weaker infinitely often guarantee of the generator can also be addressed with a careful implementation, and we refer to the formal proof for these technical details. We highlight that the construction of better (conditional) pseudorandom generators immediately leads to a tighter connection between the circuit size in the lower bound and the circuit size in the learning assumption.[5]

Revised Step 3: (conditional) PRG against uniform quantum computations. This leaves us with the last and most technical part of the proof: the construction of a PRG of sub-exponential stretch $2^{n^{\Omega(1)}}$ that fools uniform quantum computations infinitely often, assuming $\mathsf{PSPACE} \not\subseteq \mathsf{BQSUBEXP}$ (obtained from Item (b) above). In Section 1.2.1, we explained that establishing this result in the quantum setting is more delicate than in the classical scenario. Here we focus on the different components employed in the proof and how they fit together in the PRG construction.

First, we stress that our PRG $G_n \colon \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ is defined *classically*, i.e., it is computed by a *deterministic* algorithm running in time exponential in ℓ. The extension to quantum circuits lies in its security analysis, where we argue that if there is a sequence $\{C_m\}$ of uniform quantum circuits of size $\mathsf{poly}(m)$ that distinguish the output of G from random, then $\mathsf{PSPACE} \subseteq \mathsf{BQSUBEXP}$.

As alluded to above, we employ the hardness versus randomness paradigm. More precisely, our construction relies on the beautiful insight of Impagliazzo and Wigderson [IW01] on how to extend this paradigm to the *uniform* case, where the non-uniform advice appearing in the security proof can be eliminated via a clever recursive approach. This allows one to prove security based on a *uniform* hardness assumption. Following [IW01], in order to achieve this we also base the generator on a problem that is *downward-self-reducible* and *self-correctable* (such properties are useful to eliminate advice). However, we deviate from their work in terms of some other technical tools and parameters of our PRG construction, which is formally shown in Section 5.

In more detail, we construct a family of generators G_n^λ, each parameterized by a fixed $\lambda > 0$. Each G_n^λ employs a special PSPACE-complete language $L^\star$ on n input bits from Trevisan and Vadhan [TV07] (which has the self-reducibiliy properties cited above), and applies the well-known Nisan-Wigderson generator [NW94] with sub-exponential stretch $m(n) = 2^{n^\lambda}$ to an "amplified" version $\mathsf{Amp}(L^\star)$ of $L^\star$ defined over $\mathsf{poly}(n)$ input bits. $\mathsf{Amp}(L^\star)$ is obtained from $L^\star$ as follows. First, we consider its k-product $(L^\star)^k \colon \{0,1\}^{kn} \to \{0,1\}^k$, which is discussed in Section 1.2.2 in the context of hardness amplification and a result from [IJKW10]. Then, we convert $(L^\star)^k$ into

[5] Indeed, this has happened in practice, where techniques employed to design a better PRG also led to a new learning-to-lower-bound connection [CRTY20]. Our work shows that this is not a coincidence, i.e., better PRGs lead to better connections in a black-box way.

a Boolean function by a standard application of the Goldreich-Levin construction [GL89], which XORs the output of $(L^\star)^k$ with a new input string $r \in \{0,1\}^k$. This completes the sketch of the definition of G_n^λ. By a careful choice of parameters $k = \mathsf{poly}(n)$ and seed length $\ell(n) = \mathsf{poly}(n)$, it is not hard to prove that $G_n^\lambda \colon \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ can be computed in time $2^{O(\ell(n))}$.

The non-trivial aspect here is to prove that G_n^λ is quantum secure for some choice of λ. It is enough to argue that, if for every $\lambda > 0$ the generator G_n^λ can be broken, then $L^\star \in \mathsf{BQTIME}[2^{n^{o(1)}}]$. Since this language is PSPACE-complete, this violates our uniform hardness assumption. So we focus next on a fixed $\lambda > 0$ and its corresponding generator G_n^λ, and assume that there is a uniform sequence of quantum circuits $\{C_m\}$ of size $\mathsf{poly}(m)$ that distinguishes its output from random. Our goal is to conclude that $L^\star \in \mathsf{BQTIME}[2^{n^{O(\lambda)}}]$. To achieve this, we need *quantum* analogues of the "reconstruction" procedures of [NW94, GL89, IJKW10] implemented in a uniform way as in [IW01].

Quantum Nisan-Wigderson reconstruction. We observe that the original analysis of [NW94] can be adapted without difficulty in our setting. In Section 4.1, we first verify the result in the intermediate model of inherently probabilistic circuits and then extend it to the quantum case.

Quantum Goldreich-Levin reconstruction. We observe that a quantum analogue of [GL89] was established by Adcock and Cleve [AC02]. In Section 4.2, we adapt their argument to match our notation and setting of parameters. For this reason, we do not use inherently probabilistic computations here, which are convenient in the investigation of the other building blocks of our PRG construction.

Quantum Impagliazzo-Jaiswal-Kabanets-Wigderson reconstruction. As explained above, this turns out to be the most complicated aspect of the security proof, since it is non-trivial to extend the original analysis from [IJKW10] to the quantum setting. We refer to Section 1.2.2 above for an informal explanation, and to Section 4.3 for more details. We remark that it is not hard to quantize the well-known XOR Lemma for hardness amplification (e.g., Impagliazzo's proof [Imp95]), but in this work we need a *uniform* hardness amplification result with near-optimal parameters.

It remains to put together these tools to conclude that the existence of a quantum distinguisher $\{C_m\}$ implies that $L^\star \in \mathsf{BQTIME}[2^{n^{O(\lambda)}}]$. As explained above, this is done in a uniform way following the approach of [IW01]. However, controlling the *uniformity* of the final sequence of quantum circuits that compute $L^\star$ is delicate. Recall that to show that $L^\star \in \mathsf{BQTIME}[T]$ we need to produce in uniform *deterministic* time T, the description of a quantum circuit Q_n for $L^\star$ on inputs of length n. In the recursive construction of [IW01] that provides an algorithm to compute $L^\star$ on an input string $x \in \{0,1\}^n$, one "learns" how to compute $L^\star$ on every input length from 1 to n, by producing a sequence of *randomized* circuits $D_1, \ldots, D_n$ that compute $L \cap \{0,1\}^i$ for $i \in \{1, \ldots, n\}$. In order to compute D_i from D_{i-1}, it is necessary to simulate D_{i-1} on some inputs, which requires *randomness*. Similarly, the natural way to proceed in the quantum case is by generating a uniform sequence of quantum circuits $Q_1, \ldots, Q_n$ as above. Note that simulating Q_{i-1} to produce Q_i now requires a *quantum* computation. However, we must be able to *deterministically* produce the code of quantum circuits in order to show that $L^\star \in \mathsf{BQTIME}[T]$. We address this issue in Section 4.4 by going to another "meta-level" in this simulation, where the circuit Q_i "incorporates" this recursive process from i to 1 by manipulating the *codes* of quantum circuits Q_{i-1} to Q_1. (Formally, this is not too different from [IW01], which also manipulates descriptions of circuits, but in the quantum case one needs to be more explicit.) We observe that it is not hard to implement this idea if the uniform versions of the quantum reconstruction procedures described above are stated in a convenient way.

This completes the sketch of Step 3, and our overview of the proof of Theorem 1.

1.3 Directions and open problems

The most ambitious direction is to address the possibility that quantum computation might lead to faster learning algorithms for expressive classes of concepts.

Question 1. *Is there a quantum learning algorithm for Boolean circuits of size $O(n)$ that runs in time $o(2^n/n)$ and with constant probability achieves error $\varepsilon \leq 0.49$ under the uniform distribution?*

Addressing this and related problems might be out of reach given our current techniques. We focus below on directions that we find particularly interesting and possibly fruitful.

It has been suggested to us that it might also be possible to extend existing algorithms for local-list decoding of Reed-Muller codes to the inherently probabilistic setting. (This would provide an alternative presentation of our PRG construction.) While we have not verified all the details, we believe that this is quite plausible. It would be interesting to understand if there is a more general phenomenon in place here, i.e., whether certain classes of ECCs and decoding algorithms can be extended to the inherently random setting in a generic fashion, and to investigate further applications of such codes.

Our results offer the exciting possibility that new circuit lower bounds might follow through the design of quantum algorithms. Recall that Williams [Wil14] established via the satisfiability-to-lower-bound connection that $\mathsf{NE} \not\subseteq \mathsf{ACC}^0$. Similarly, can we use our new learning-to-lower-bound connection to show that ACC^0 circuits cannot compute all functions in BQE? Using our techniques (cf. Corollary 3.6), it would be sufficient to provide a positive answer to the following question.

Question 2. *Is there a (promise) quantum natural property useful against ACC^0 circuits?*

Note that in order to achieve this it suffices to construct a quantum natural property for the larger class SYM^+ of quasi-polynomial size depth-2 circuits consisting of an arbitrary symmetric gate at the top layer fed with AND gates of poly-logarithmic fan-in at the bottom layer [BT94]. Similarly, it is enough to get a quantum natural property for the class of functions that can be approximated by a torus polynomial of bounded degree [BHLR19], or for Boolean matrices of bounded "symmetric rank" [Wil18]. Perhaps quantum computations can be helpful in the design of natural properties for these or for other circuit classes (e.g., Boolean formulas of size $n^{3.01}$)?

We are also curious about the prospects of designing non-trivial quantum learners for restricted circuit classes. Servedio and Tan [ST17] explored this possibility in the context of classical computation, showing several examples where non-trivial savings can be achieved compared to the trivial "brute-force" learning algorithm that runs in time $O(2^n)$. As alluded to above, another regime of parameters in the quantum setting that might be interesting to explore is that of polynomial-time learners that achieve a non-trivial advantage $\gamma \gg 2^{-n/2}$.

Finally, the investigation of the classical learning-to-lower-bound connection established in [OS17] led to many other results, such as the existence of a learning speedup phenomenon [OS17, Lemma 1] and unconditional complexity lower bounds [Oli19]. It would be interesting to see if new consequences in quantum complexity theory and quantum learning theory can be obtained using the techniques introduced in this work.

1.4 Related work

There is a rich history of connections between circuit complexity theory and the investigation of learning algorithms. In many cases, *specific* circuit lower bound techniques have been used to design new algorithms. For instance, Linial, Mansour, and Nisan [LMN93] relied on the method of random restrictions to show that constant-depth polynomial-size circuits can be PAC-learned

in quasi-polynomial time under the uniform distribution from random examples. In a more recent development, Carmosino, Impagliazzo, Kabanets, and Kolokolva [CIKK16] showed that learning algorithms can be obtained from *any* lower bound technique that is "*constructive*" in the sense of the theory of natural proofs of Razborov and Rudich [RR97]. This allowed them to show that constant-depth polynomial-size circuits augmented with parity gates can be PAC learned in quasi-polynomial time under the uniform distribution from membership queries. These, and several other results (cf. Servedio and Tan [ST17]), show that a circuit lower bound against a circuit class $\mathcal{C}$, in most cases, can be converted into a learning algorithm for $\mathcal{C}$.

The first results in the opposite direction, i.e., showing that learning algorithms imply circuit lower bounds, were established by Fortnow and Klivans [FK09]. In their work, among other results, they proved that any *sub-exponential time* (i.e., $2^{n^{o(1)}}$), *deterministic* exact learning algorithm for $\mathcal{C}$ (using membership and equivalence queries) implies the existence of a function in E^{NP} that is not in $\mathcal{C}$. They also showed that if $\mathcal{C}$ is PAC learnable with membership queries under the uniform distribution or exact learnable in *randomized polynomial-time*, there is a function in BPE (an exponential time analog of BPP) that is not in $\mathcal{C}$. Note that these results have the appealing feature that they make no assumption about the techniques employed in the design of the learning algorithm. Nevertheless, there is an important drawback in the initial results of [FK09]: they make strong assumptions about the *resources* of the learning algorithm. For example, many learning algorithms are randomized and require at least quasi-polynomial time (e.g., [LMN93, CIKK16]), and the results from [FK09] do not apply in this case.

Over the last decade several works have addressed this and other aspects of the learning-to-lower-bound connection. Harkins and Hitchcock [HH13] eliminated the NP oracle and showed that $\mathsf{E} \not\subseteq \mathcal{C}$ from the existence of exact deterministic sub-exponential time learning algorithms with membership and equivalence queries. Shortly after, Klivans, Kothari, and Oliveira [KKO13] simplified these proofs, strengthened a few existing learning-to-lower-bound connections, and obtained results for additional learning models (e.g., learning from statistical queries). In particular, [KKO13] proved that $\mathsf{E} \not\subseteq \mathcal{C}$ if there are exact learners for $\mathcal{C}$ running in deterministic time $o(2^n)$, showing that "non-trivial" *deterministic* learners yield lower bounds.

In a different direction, Volkovich [Vol14] showed how to extract lower bounds in polynomial-time classes from randomized learners running in polynomial time. More precisely, [Vol14] shows that $\mathsf{BPP}/1$ (BPP with 1 bit of non-uniform advice per input length) is not contained in $\mathcal{C}[n^k]$ for all $k \geq 1$ if $\mathcal{C}$ can be PAC learned with membership queries under the uniform distribution by randomized learners running in polynomial time. In another work, Volkovich [Vol16] explores connections between algebraic complexity lower bounds and learning algorithms.

While extracting circuit lower bounds from non-trivial *deterministic* learners can be done using elementary techniques [KKO13], extending the result to *randomized* learners required advanced tools from complexity theory. This was first achieved by Oliveira and Santhanam [OS17], who proved that if for every $k \geq 1$ the class $\mathcal{C}[n^k]$ can be PAC learned under the uniform distribution with membership queries by a randomized algorithm running in time $2^n/n^{\omega(1)}$, then for every $k \geq 1$ we have $\mathsf{BPE} \not\subseteq \mathcal{C}[n^k]$. Their result admits an extension to faster learners with a weaker advantage γ, with parameters similar to our Theorem 1. A simpler proof that non-trivial randomized learning yields lower bounds, with a stronger consequence, was obtained by Oliveira and Santhanam [OS18] under the additional assumption that the learner has "pseudo-deterministic" behaviour. We refer to their work for details.

In a more recent work, Oliveira [Oli19] explored the learning-to-lower-bound connection in an *indirect* way to show that a natural problem in time-bounded Kolmogorov complexity cannot be solved in probabilistic polynomial time. This can be achieved by proving a lower bound *under the*

assumption that learning algorithms do not exist (since otherwise a useful lower bound immediately follows from an existing learning-to-lower-bound connection).

Chen, Rothblum, Tell, and Yogev [CRTY20] established a "fine-grained" learning-to-lower-bound connection with respect to circuit size in the setting of randomized learners. More precisely, they show that learning general Boolean circuits of size $n \cdot \mathsf{poly}(\log n)$ in randomized time $2^{n/\mathsf{poly}(\log n)}$ yields a function in BPE that cannot be computed by circuits of size $n \cdot \mathsf{poly}(\log n)$. While their running time assumption is much more stringent than $2^n/n^{\omega(1)}$, it is not necessary to learn circuits of large polynomial size to get interesting lower bound consequences.

In a parallel line of work, Williams [Wil13] employed completely different methods to establish that "non-trivial" (running in time $2^n/n^{\omega(1)}$) deterministic *satisfiability* algorithms for a class $\mathcal{C}$ of polynomial-size circuits imply that $\mathsf{NE} \not\subseteq \mathcal{C}$, where NE is an exponential time analogue of NP. The satisfiability-to-lower-bound connection and its extensions have been highly successful in establishing new circuit lower bounds for a variety of circuit classes (see e.g., [Wil14, MW18, CR20, CLW20] and references therein). Chen, Oliveira, and Santhanam [COS18] combined *learning* and *satisfiability* algorithms to strengthen lower bound consequences from learning algorithms in the particular case of $\mathcal{C} = \mathsf{ACC}^0$ (constant-depth polynomial-size circuits extended with modular gates).

In contrast to all these works, here we establish the first general connection between the design of *quantum* (learning) algorithms for an arbitrary class $\mathcal{C}$ of polynomial-size circuits and the existence of lower bounds against $\mathcal{C}$.

Organization. In Section 2, we fix notation and review a few basic definitions and results, as well as formalize our quantum learning model and the useful concept of inherently random circuits. In Section 3, we establish that non-trivial quantum learners imply lower bounds (Theorem 1), under the assumption that a conditional PRG against quantum computations exists. Section 4 develops the tools that are needed to establish the correctness of our PRG construction. Finally, Section 5 defines and analyses a PRG with the desired properties, which completes the proof of Theorem 1.

Acknowledgements. We thank Lijie Chen (MIT) for several comments and feedback on a preliminary version of this paper, and the anonymous reviewers of QIP'2021 and FOCS'2021 for many useful remarks about the presentation. S.A. was partially supported by the IBM Research Frontiers Institute and acknowledges support from the Army Research Laboratory, the Army Research Office under grant number W911NF-20-1-001. A.S. was partially supported by the Joint Centre for Quantum Information and Computer Science, University of Maryland, USA and acknowledges support from the Department of Defense. Part of this work was done while A.G. was affiliated to CWI and QuSoft. Part of this work was done while S.A, A.G. and A.S were participating in the program "The Quantum Wave in Computing" at Simons Institute for the Theory of Computing. This work received support from the Royal Society University Research Fellowship URF\R1\191059 and the UKRI Future Leaders Fellowship MR/S031545/1.

2 Preliminaries

2.1 Basic definitions and notation

We begin with standard notation that will be employed throughout this work.

– We use $\mathbb{N}$ to denote the set $\{1, 2, 3, \ldots\}$ of positive integers.
– We denote by $\mathcal{U}_n$ the uniform distribution over n-bit strings.

– When sampling a uniformly random element a from a set A, we might use $a \in A$ or $a \sim A$.
– We say that two Boolean functions $f, g\colon \{0,1\}^n \to \{0,1\}$ are λ-close if $\Pr_{x\sim\mathcal{U}_n}[f(x) \neq g(x)] \leq \lambda$.
– We say that f computes g with advantage γ if $\Pr_{x\sim\mathcal{U}_n}[f(x) = g(x)] \geq 1/2 + \gamma$.
– We use $\mathsf{negl}(n)$ to denote a function $g\colon \mathbb{N} \to \mathbb{N}$ such that, for every univariate polynomial $p(\cdot)$ with positive coefficients, there exists $n_0 \in \mathbb{N}$ such that $g(n) \leq 1/p(n)$ for every $n \geq n_0$.

We assume the familiarity of the reader with standard complexity classes (e.g. PSPACE) and basic notions from classical computational complexity theory and refer to a textbook such as [AB09] for more information. Some notions from quantum computing and quantum complexity theory are reviewed in Section 2.2.

2.1.1 Useful results

The following standard concentration bound will be used in some proofs.

Lemma 2.1 (Chernoff bound; see e.g. [AS16, Theorem A.1.15])**.** *Let $X_1, \ldots, X_k$ be i.i.d. $\{0,1\}$-valued random variables each taking value 1 with probability p. Let $X = \sum_i X_i$ and $\mu = pk$. Then for any $\delta \in (0,1)$ it follows that*

$$\Pr\left[X \leq (1-\delta)\mu\right] \leq e^{-\delta^2\mu/2}.$$

An alternate version of this bound will also be useful.

Lemma 2.2 (Chernoff bound; see e.g. [JLR11, Theorem 2.1])**.** *Let $X_1, \ldots, X_k$ be i.i.d. $\{0,1\}$-valued random variables each taking value 1 with probability p. Let $X = \sum_i X_i$ and $\mu = pk$. Then for any $t \geq 0$ it follows that*

$$\Pr\left[|X - \mathbb{E}[X]| \geq t\right] \leq \exp\Big(-\frac{t^2}{2(\mu + t/3)}\Big).$$

We will also need the following form of the Hoeffding bound [Hoe63].

Lemma 2.3 (Hoeffding bound [Hoe63])**.** *Let $F\colon \mathcal{V} \to [0,1]$ be a function over a finite set $\mathcal{V}$ with expectation $\mathbb{E}_{x\sim\mathcal{V}}[F(x)] = \alpha \in [0,1]$. Let R be a random subset of $\mathcal{V}$ of size t, and consider the random variable $X = \sum_{x\in R} F(x)$. Then the expectation of X is $\mu = \alpha t$, and for any $0 \leq \gamma \leq 1$,*

$$\Pr\left[|X - \mu| \geq \gamma\mu\right] \leq 2 \cdot e^{-\gamma^2\mu/3}.$$

2.1.2 Circuit and concept classes

For a typical Boolean circuit class $\mathcal{C}$ such as $\mathcal{AC}^0$, $\mathcal{TC}^0$, Formulas, etc., and for a size function $s\colon \mathbb{N} \to \mathbb{N}$ measuring number of gates, we use $\mathcal{C}[s]$ to denote the set of languages $L \subseteq \{0,1\}^*$ that can be computed by a sequence of $\mathcal{C}$-circuits of size $s(n)$. For circuit classes of bounded depth, we might use $\mathcal{C}_d$ when referring to circuits of depth at most d. We stress that our notation refers to *non-uniform* circuit classes, and consequently the complexity lower bound appearing in Theorem 1 is a non-uniform circuit complexity lower bound.

We will also employ circuit classes in the context of learning algorithms, where they are often referred to as concept classes. In this case, we will abuse notation and view $\mathcal{C}[s(n)]$ as the class of Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ that can be computed by $\mathcal{C}$ circuits over n input variables of size at most $s(n)$. For convenience, we use $\mathsf{size}_{\mathcal{C}}(f)$ to denote the minimum number of gates in a $\mathcal{C}$-circuit computing f. We omit the subscript $\mathcal{C}$ in $\mathsf{size}_{\mathcal{C}}(f)$ when we refer to general Boolean

circuits. For concreteness, in this case circuit size refers to number of gates in the model consisting of fan-in two circuits over AND, OR, and NOT gates.

Theorem 1 applies to a broad family of circuit classes investigated in computational complexity theory, including fixed-depth classes such as depth-2 circuits consisting of majority gates. The only assumptions needed on the circuit class $\mathcal{C}$ are that:

(*i*) Any function $f_n \in \mathcal{C}[s(n)]$ can be computed by a general Boolean circuit of size $\mathsf{poly}(s(n))$. This is the case, for instance, if every gate allowed in $\mathcal{C}$ can be simulated by a Boolean circuit of size polynomial in the number of input wires of the gate.

(*ii*) The class $\mathcal{C}$ is closed under restrictions of input variables to constants 0 and 1. In other words, if $f_n \in \mathcal{C}[s(n)]$ and we fix some input variables of f_n to obtain a function $f'_n \colon \{0,1\}^{n'} \to \{0,1\}$, then f'_n is also computed by a $\mathcal{C}$-circuit of size at most $s(n)$.

These assumptions are needed only in Section 3.2, and we rely on each of them as follows. We use Item (*i*) to show that if a language L is not computable by (unrestricted) Boolean circuits of size n^α for some $\alpha \geq 1$, then it cannot be computed by $\mathcal{C}$-circuits of size n^β, where $\beta = \alpha/C'$ for some universal positive constant C' that is independent of α. On the other hand, we will rely on Item (*ii*) to say that if a function $f \colon \{0,1\}^n \to \{0,1\}$ can be computed by $\mathcal{C}$-circuits of size polynomial in n, then any sub-function of f defined over $n' = n^{\Omega(1)}$ input variables via a restriction also admits $\mathcal{C}$-circuits of size polynomial in n'.

Note that Theorem 1 indeed applies to virtually any class of polynomial size circuits.

We refer to a standard reference such as [Juk12] for more background on Boolean functions and circuit complexity theory.

2.1.3 Classical learning algorithms

For the convenience of the reader, we review in this section (classical) learnability under the uniform distribution in the Probably Approximately Correct (PAC) model extended with membership queries. The quantum learning model that we consider in this work and its extensions are discussed in Section 2.3.

Definition 2.4. *Let $\mathcal{C}_n$ be a family of Boolean functions on n input variables, $\mathcal{C} = \cup_{n \geq 1} \mathcal{C}_n$, $T \colon \mathbb{N} \to \mathbb{N}$, and $\varepsilon, \delta \colon \mathbb{N} \to [0,1]$. We say that $\mathcal{C}$ can be (ε, δ)-learned in time T under the uniform distribution with membership queries if there exists a randomized algorithm $\mathcal{A}$ that satisfies the following guarantees:*

> *For all $n \in \mathbb{N}$, for every $f \in \mathcal{C}_n$, given n and query access to f, $\mathcal{A}$ runs in time $T(n)$ and with probability at least $1 - \delta(n)$ over its internal randomness outputs a hypothesis h (encoded as a general Boolean circuit) that satisfies*
>
> $$\Pr_{x \sim \mathcal{U}_n} [h(x) = f(x)] \geq 1 - \varepsilon(n).$$

It is also possible to focus on a fixed circuit class $\mathcal{C}$, and to consider learnability of functions computed by $\mathcal{C}$-circuits of an arbitrary size $s(n)$, for a fixed learning algorithm $\mathcal{A}$ that is independent of s but that is given the value $s_f = \mathsf{size}_{\mathcal{C}}(f)$ (or an upper bound on s_f) as part of its input. In this case, we allow the running time of $\mathcal{A}$ to depend on s_f. Similarly, we can provide the learner with δ and ε, and allow its running time to depend on these parameters.

We stress that the output hypothesis h produced by $\mathcal{A}$ is not required to be a "circuit" from $\mathcal{C}$, when $\mathcal{C}$ explicitly refers to a circuit class and a size measure.

We refer to a standard reference such as [KV94] for more background in computational learning theory.

2.2 Quantum computation

We assume the reader is familiar with the quantum computing framework and notation, such as Dirac's bra-kets. We refer to a standard text such as [NC10] for a detailed introduction to quantum computation. Here, we discuss concepts and notation of specific relevance to this paper.

A quantum circuit U on n-qubits is a sequence of quantum gates (i.e., unitary matrices). Throughout this paper we will assume our quantum gates are restricted to the one-qubit Hadamard gate H and 3-qubit Toffoli gate Toff defined as follows:

$$\mathsf{H} : |b_1\rangle \to \frac{1}{\sqrt{2}}\big(\,|0\rangle + (-1)^{b_1}\,|1\rangle\,\big), \qquad \mathsf{Toff} : |b_1, b_2, b_3\rangle \to |b_1, b_2, b_3 \oplus b_1 \cdot b_2\rangle$$

for $b_1, b_2, b_3 \in \{0,1\}$. The Toffoli gate is particularly useful for us as any classical circuit can be implemented as a quantum circuit using only Toffoli gates. Additionally, we choose this set of gates because $\{\mathsf{H}, \mathsf{Toff}\}$ is universal for approximating unitaries with only real entries, i.e., every unitary with real entries can be approximated arbitrarily well with only H and Toff gates [Aha03]. In fact, our results still go through as long as we have a gate set whose size is constant. In our case, the size of a quantum circuit is the number of Hadamard and Toffoli gates in the circuit.

The classical description of a quantum circuit U on n-qubits, denoted as $\mathsf{code}(U)$, is a canonical encoding of the sequence of gates (and the qubits they act on) in the circuit, represented as a binary string. We will repeatedly use the result that, given the description $\mathsf{code}(U)$ of a quantum circuit U of a predetermined size, there exists an *efficient* universal quantum circuit $\mathscr{U}$ that can simulate the action of U on any n-qubit input $|y\rangle$. Formally, we have the following definition and result.

Definition 2.5 (Universal quantum circuit)**.** *Fix $n \in \mathbb{N}$ and let $\mathcal{C}$ be the collection of quantum circuits on n-qubits of size $s(n)$. An $(n+m)$-qubit quantum circuit $\mathscr{U}$ is universal for $\mathcal{C}$ if for every circuit $U \in \mathcal{C}$ and associated $\mathsf{code}(U) \in \{0,1\}^*$,*

$$\mathscr{U}\left(|y\rangle \otimes |\mathsf{code}(U)\rangle\right) = (U\,|y\rangle) \otimes |\mathsf{code}(U)\rangle\,, \quad \textit{for every } y \in \{0,1\}^n.$$

There exist efficient constructions of $\mathscr{U}$ whose size has only a log-factor blow-up in the parameters n and $s(n)$ (see [BFGH10]). (We note that a polynomial blow-up would still be sufficient in our constructions.)

We say that a quantum circuit U computes a function $f : \{0,1\}^n \to \{0,1\}$ if the following holds: there exists $m \geq 0$ such that, for every $x \in \{0,1\}^n$, when measuring the first qubit of U on input $|x, 0^m\rangle$, we get $f(x)$ with probability at least $2/3$, i.e.,

$$\left\|\Pi_{f(x)} U\,|x, 0^m\rangle\,\right\|^2 \geq 2/3,$$

where $\Pi_{f(x)} \stackrel{\text{def}}{=} |f(x)\rangle\langle f(x)| \otimes \mathsf{Id}$. The extra m qubits are called auxiliary qubits. As in probabilistic computing, the constant $2/3$ is not important here as one can use standard amplification techniques, such as taking the majority vote over many repetitions, to amplify this probability to $1-\delta$ for some $\delta > 0$.

Often in this paper, we will abuse notation and write $\Pr[U(x) = f(x)]$ when referring to $\|\Pi_{f(x)} U\,|x, 0^m\rangle\,\|^2$ – the probability that the quantum circuit U on input x outputs $f(x)$ after measurement. Similarly, we might write

$$\Pr_{x \sim \{0,1\}^n,\, U}[U(x) = f(x)] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x \sim \{0,1\}^n}\left[\left\|\Pi_{f(x)} U\,|x, 0^m\rangle\,\right\|^2\right]$$

to refer to the *expected agreement* between the output of U and f over a random input string.

These definitions extend naturally to the case of functions with a non-Boolean output. For instance, if $g\colon \{0,1\}^n \to \{0,1\}^\ell$, then we use

$$\Pr_{x\sim\{0,1\}^n,\, U}[U(x) = g(x)] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x\sim\{0,1\}^n}\left[\left\|\Pi_{g(x)} U\,|x, 0^m\rangle\,\right\|^2\right],$$

where Π_w for a string $w \in \{0,1\}^\ell$ is defined as $= |w\rangle\langle w| \otimes \mathsf{Id}$.

Unlike classical probabilistic computation, quantum interference could result in the output qubit being entangled with junk values left in the auxiliary qubits when they are used in intermediate computations. This could impact the probability that $[U(x) = f(x)]$ on measurement – especially when U is used as a sub-routine in larger computations. However, by relying on the reversible nature of unitary computations, we can "*remove the garbage*" via standard techniques. In particular, given U on $n+m$ qubits that computes f,[6] construct the garbage free version $\tilde{U}$ on $n+m+1$ qubits as follows:

(*i*) Compute U on input $|x, 0^m\rangle$;

(*ii*) Copy the first qubit of U into the $(n+m+1)$-th qubit of $\tilde{U}$ with a quantum gate;

(*iii*) Un-compute U by applying $U^\dagger$ on the first $n+m$ qubits – i.e., apply each gate of U in reverse order. This will return $|x, \mathsf{junk}\rangle$ back to $|x, 0^m\rangle$;

(*iv*) Measure the $(n+m+1)$-th qubit to check if $\tilde{U}(x) = f(x)$ with $\widetilde{\Pi}_{f(x)} = |f(x)\rangle\langle f(x)| \otimes |x\rangle\langle x| \otimes |0^m\rangle\langle 0^m|$.

The lower bounds in this paper involve quantum complexity classes, which are defined next. Owing to the probabilistic nature of quantum circuits, all classes of interest are in the bounded error paradigm. We first consider the uniform class BQTIME (bounded-error quantum time).

Definition 2.6 (BQTIME)**.** *Let $f : \{0,1\}^* \to \{0,1\}$ and $t : \mathbb{N} \to \mathbb{N}$. We say that f is computable in bounded-error quantum $t(n)$-time if there exists a deterministic algorithm A which on input 1^n runs in time $\mathrm{poly}(n, t(n))$ and outputs the description of a quantum circuit U, represented as a string $\mathsf{code}(U)$, with at most $t(n)$ quantum gates such that U computes f. In this case, we write $f \in \mathsf{BQTIME}[t(n)]$.*

We also fix notation for the complexity classes that arise when $t(n)$ scales polynomially or exponentially in n.

Definition 2.7 (Bounded-error quantum complexity classes)**.** *Let $n \in \mathbb{N}$ and $\nu, c > 0$ be constants. Then, quantum polynomial time refers to the class $\mathsf{BQP} := \bigcup_{c>0} \mathsf{BQTIME}[n^c]$. Similarly, $\mathsf{BQE} := \bigcup_{c>0} \mathsf{BQTIME}[2^{c\cdot n}]$ refers to the class of languages computable in quantum $2^{O(n)}$-time. Quantum sub-exponential time is denoted as $\mathsf{BQSUBEXP} := \bigcap_{0<\nu<1} \mathsf{BQTIME}\left[2^{n^\nu}\right]$.*

2.3 Quantum learning algorithms and extensions

In this section we define the quantum learning model. In contrast to the classical model, a quantum learner is given quantum oracle access to a concept f. This means the learner is allowed

[6] We remark that in Section 4.4 when we need to remove garbage in unitary computations, we will always amplify the success probability of U, so as to ensure that U computes f with overwhelming probability before constructing $\tilde{U}$.

to perform a *quantum membership query*, which is defined by a unitary map O_f acting on $n+1$ qubits whose action on basis states is defined as

$$O_f : |x, b\rangle \to |x, b \oplus f(x)\rangle \,,$$

for every $x \in \{0,1\}^n$ and $b \in \{0,1\}$. Naturally, a quantum learner can perform a quantum computation in between quantum queries, and its goal is the same as for a classical learner (defined in Section 2.1.3), i.e., to output a hypothesis h that approximates the target concept $f \in \mathcal{C}$.

We can formalise this model as follows. A quantum learning algorithm $\mathcal{L}$ running in time G is described by a uniform sequence of quantum circuits Q_n, where each Q_n contains at most $G(n)$ gates. The quantum circuit Q_n expects as input $|0^m\rangle$ for some $m = m(n)$, and consists of a sequence of gates from $\{\mathsf{H}, \mathsf{Toff}, \mathcal{O}\}$, where $\mathcal{O}$ is defined over $n+1$ qubits and is considered as applying a single gate. In other words, when learning an unknown Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, $\mathcal{O}$ computes as the unitary map O_f described above. Finally, the output hypothesis of Q_n when computing with quantum membership query access to f (referred to as Q_n^f) is described by the string corresponding to the output measurement of Q_n^f. We note that intermediate measurements are also allowed in Q_n.

Definition 2.8 (Standard quantum learners)**.** *Let $\mathcal{C}_n$ be a family of Boolean functions on n input variables and $\mathcal{C} = \cup_{n\geq 1}\mathcal{C}_n$. Let $G : \mathbb{N} \to \mathbb{N}$. We say $\mathcal{C}$ can be $(\varepsilon(n), \delta(n))$-learned in time $G(n)$ if there exists a quantum learning algorithm $\mathcal{L}$ that satisfies the following:*

> *For all $n \in \mathbb{N}$, for every $f \in \mathcal{C}_n$, given quantum oracle access to f, $\mathcal{L}$ uses at most $G(n)$ gates and with probability at least $1 - \delta(n)$ (over the output measurement of $\mathcal{L}$) outputs a (classical) hypothesis h that satisfies*
>
> $$\Pr_{x\sim\mathcal{U}_n}[h(x) = f(x)] \geq 1 - \varepsilon(n).$$

We remark that one distinct advantage of having quantum oracle access to f is that one can efficiently *Fourier sample* for the squared Fourier distribution $\{\widehat{f^2}(S)\}_S$, which might be computationally hard for classical learners (we discuss this in more detail in Appendix A). For more on this quantum learning model, we refer the interested reader to [SG04, AW17].

In this paper, we consider a natural generalization of this model: we allow the quantum learner to output a *quantum* hypothesis, meaning that $\mathcal{L}$ is allowed to output a classical description of a quantum circuit $\widetilde{U_f}$ that *approximates* the function f. Recall that *computing* f on a given input x means that measuring the first qubit of U acting on an input $|x\rangle$ and auxiliary qubits set to $|0\rangle$, in the computational basis, produces $f(x)$ with probability $\geq 2/3$. On the other hand, our notion of a quantum circuit $\widetilde{U_f}$ that approximates f refers to the *expected agreement* between the output of $\widetilde{U_f}$ and f on a random input x, as defined in Section 2.2.

We formally define this model below.[7]

Definition 2.9 (Quantum learning with quantum hypothesis)**.** *Let $\mathcal{C}_n$ be a family of Boolean functions on n input variables and $\mathcal{C} = \cup_{n\geq 1}\mathcal{C}_n$. Let $G : \mathbb{N} \to \mathbb{N}$. We say $\mathcal{C}$ can be $(\varepsilon(n), \delta(n))$-learned in time $G(n)$ if there exists a quantum learning algorithm $\mathcal{L}$ that satisfies the following:*

[7] We only define the uniform distribution learning model below. Note that one can naturally define a quantum learner with quantum hypothesis under an *arbitrary* distribution by changing $\mathcal{U}_n$ to an arbitrary distribution $\mathcal{D}$.

For all $n \in \mathbb{N}$, for every $f \in \mathcal{C}_n$, given quantum oracle access to f, $\mathcal{L}$ uses at most $G(n)$ gates and with probability at least $1 - \delta(n)$ (over the output measurement of $\mathcal{L}$) outputs the description of a quantum circuit $\widetilde{U_f}$ that satisfies

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n} \left[\|\Pi_{f(x)} \widetilde{U_f} |x\rangle |0\rangle\|^2 \right] \geq 1 - \varepsilon(n) ,$$

where $\Pi_{f(x)} = |f(x)\rangle\langle f(x)| \otimes \mathsf{Id}$.

2.4 Quantum natural properties

Let $\mathcal{F}_n$ be the family of all Boolean functions on n input bits. We say that $\Gamma = \{\Gamma_n\}_{n \geq 1}$ is a combinatorial property of Boolean functions if $\Gamma_n \subseteq \mathcal{F}_n$ for every $n \geq 1$. For every function $f \in \mathcal{F}_n$ and $N = 2^n$, let $\mathsf{tt}(f) \in \{0,1\}^N$ be the truth table representing f. We associate with Γ a language $L_\Gamma \subseteq \{0,1\}^*$ defined as follows. A string x is in L_Γ if and only if $x = \mathsf{tt}(f)$ for some n and $f \in \Gamma_n$. Conversely, given a string $w \in \{0,1\}^N$, let fnc^w denote the boolean function $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{tt}(f) = w$.

Definition 2.10 (Natural property [RR97])**.** *Let Γ be a combinatorial property, $\mathcal{C}$ be a circuit class, $\mathfrak{D}$ be a (uniform or non-uniform) complexity class, and $s \colon \mathbb{N} \to \mathbb{N}$. We say that Γ is a $\mathfrak{D}$-natural property useful against $\mathcal{C}[s]$ if there exists $n_0 \in \mathbb{N}$ for which the following holds:*

- *Constructivity: $L_\Gamma \in \mathfrak{D}$, i.e., for every n, $f_n \stackrel{?}{\in} \Gamma_n$ is decidable in $\mathfrak{D}$.*
- *Largeness: For every $n \geq n_0$, $\Pr_{f \sim \mathcal{F}_n}[f \in \Gamma_n] \geq 1/2$.*
- *Usefulness: For every $n \geq n_0$, $\mathcal{C}_n[s(n)] \cap \Gamma_n = \emptyset$.*

If $\mathfrak{D} = \mathsf{BQP}$, we say that Γ is a quantum *natural property.*

Note for instance that for BPP and BQP-natural properties the corresponding algorithm (that decides L_Γ) is allowed to run in time polynomial in the input length $N = 2^n$. We will need to slightly relax the guarantees offered by a natural property in order to establish a connection to learning.

Definition 2.11 (Informal)**.** *We say that a circuit class $\mathcal{C}$ admits a* promise *natural property if the underlying algorithm (that decides L_Γ) in Definition 2.10 accepts every string in L_Γ with probability $\geq 2/3$ and rejects every string encoding a function in $\mathcal{C}_n[s(n)]$ with probability $\geq 2/3$. (For instance, the algorithm might accept some strings with probability close to $1/2$.)*

Definition 2.12 (Promise BQP-natural property)**.** *We say that there is a (promise) BQP-quantum natural property against $\mathcal{C}$-circuits of size s if there is a quantum algorithm $\mathcal{D}$ operating over inputs of the form $N = 2^n$ for which the following holds for every large enough parameter n:*

- *Constructiveness: $\mathcal{D}$ runs in time polynomial on its input length N.*
- *$s(n)$-hardness: For every $f \in \mathcal{C}_n[s(n)]$, $\mathcal{D}$ accepts $\mathsf{tt}(f)$ with probability at most $1/3$.*
- *Density: There is a set $\Gamma_n \subseteq \{0,1\}^N$ of density $|\Gamma_n|/2^N \geq 1/2$ such that, for every $w \in \Gamma_n$, $\mathsf{fnc}^w \notin \mathcal{C}_n[s(n)]$ and $\mathcal{D}$ accepts w with probability at least $2/3$.*

Remark 2.13. We stress that by the definition of BQP, a promise BQP-natural property implies a *classical* algorithm A such that $A(1^N)$ computes the quantum circuit D_N that deals with inputs of size N.

2.5 Self-reducibility

Definition 2.14 (Downward self-reducibility)**.** *A function $f\colon \{0,1\}^* \to \{0,1\}$ is said to be downward self-reducible if there is a deterministic polynomial-time oracle procedure A^f such that*:

1. *On any input x of length n, $A^f(x)$ only makes queries of length $< n$.*
2. *For every input x, $A^f(x) = f(x)$.*

Definition 2.15 (Random self-reducibility)**.** *A function $f\colon \{0,1\}^* \to \{0,1\}$ is said to be random self-reducible if there are constants $a, b, c \geq 1$ and polynomial-time computable functions $g\colon \{0,1\}^* \to \{0,1\}^*$ and $h\colon \{0,1\}^* \to \{0,1\}$ satisfying the following conditions*:

1. *For large enough n, for every $x \in \{0,1\}^n$ and for each $i \in \mathbb{N}$ such that $i \in [n^a]$, $g(i,x,r) \sim \mathcal{U}_n$ when $r \sim \mathcal{U}_{n^c}$.*
2. *For large enough n and for every function $\tilde{f}_n\colon \{0,1\}^n \to \{0,1\}$ that is $(1/n^b)$-close to f on n-bit strings, for every $x \in \{0,1\}^n$:*

$$f(x) = h\big(x, r, \tilde{f}_n(g(1,x,r)), \tilde{f}_n(g(2,x,r)), \ldots, \tilde{f}_n(g(n^a,x,r))\big)$$

with probability $\geq 1 - 2^{-2n}$ when $r \sim \mathcal{U}_{n^c}$.

Trevisan and Vadhan [TV07] construted a language in PSPACE that simultaneously satisfies the following conditions.[8]

Theorem 2.16 (Trevisan and Vadhan [TV07])**.** *There is a language $L^\star \subseteq \{0,1\}^*$ satisfying the following properties:*

- *$L^\star \in$ PSPACE. In particular, there is a positive constant $d_\star$ such that L on inputs of length n can be computed in time $O(2^{n^{d_\star}})$.*
- *$L^\star$ is complete for PSPACE with respect to deterministic polynomial-time reductions.*
- *$L^\star$ is downward-self-reducible and random-self-reducible, with parameters $a_\star$, $b_\star$, and $c_\star$ in Definition 2.15.*

2.6 Pseudorandomness

For $s \in \mathbb{N}$ and $\varepsilon \in [0,1]$, we say that a distribution $\mathcal{D}$ supported over $\{0,1\}^m$ is (s,ε)-*pseudorandom against quantum circuits* if for every quantum circuit C of size s defined over m input bits, we have

$$\left| \Pr_{x \sim \{0,1\}^m,\, C}[C(x) = 1] - \Pr_{y \sim \mathcal{D},\, C}[C(y) = 1] \right| \leq \varepsilon.$$

Note that each probability above refers to an appropriate random input and the randomness present in the output of C.

We will consider weaker forms of pseudorandomness that hold against *uniformly constructed* families of quantum circuits. For functions $s\colon \mathbb{N} \to \mathbb{N}$ and $\varepsilon\colon \mathbb{N} \to [0,1]$, we say that an ensemble $\{\mathcal{D}_m\}_{m \geq 1}$ of distributions $\mathcal{D}_m$ supported over $\{0,1\}^m$ is (s,ε)-*pseudorandom against uniform*

[8] The proof of this result in [TV07] refers to "self-correction" instead of "random-self-reducibility", but they ultimately rely on a decoding algorithm for polynomials making queries that are uniformly distributed on each coordinate.

quantum circuits if for every deterministic algorithm $A(1^m)$ that runs in time $s(m)$ and outputs a quantum circuit C_m over m input variables and of size at most $s(m)$,

$$\left| \Pr_{x \sim \{0,1\}^m, C_m} [C_m(x) = 1] - \Pr_{y \sim \mathcal{D}_m, C_m} [C_m(y) = 1] \right| \leq \varepsilon(m).$$

For technical reasons, the pseudorandom distributions we construct are supported over strings of length $m(n) = \lfloor 2^{n^\lambda} \rfloor$, for $n \in \mathbb{N}$ and a fixed $\lambda > 0$. (The index parameter n will play an important role in our constructions.) Moreover, we will only be able to show that the distributions $\mathcal{D}_{m(n)}$ are pseudorandom for infinitely many values of n. For these reasons, we refine the previous definition in the following way.

Let $m \colon \mathbb{N} \to \mathbb{N}$, $s \colon \mathbb{N} \to \mathbb{N}$ and $\varepsilon \colon \mathbb{N} \to [0,1]$. We say that an ensemble $\{\mathcal{D}_{m(n)}\}_{n \geq 1}$ of distributions $\mathcal{D}_m$ supported over $\{0,1\}^{m(n)}$ is *infinitely often* (s, ε)*-pseudorandom against uniform quantum circuits* if for every deterministic algorithm $A(1^m)$ that runs in time $s(m)$ and outputs a quantum circuit C_m over m input variables and of size at most $s(m)$, for infinitely many values of n and $m \stackrel{\text{def}}{=} m(n)$,

$$\left| \Pr_{x \sim \{0,1\}^m, C_m} [C_m(x) = 1] - \Pr_{y \sim \mathcal{D}_m, C_m} [C_m(y) = 1] \right| \leq \varepsilon(m).$$

Finally, our distributions are obtained from *pseudorandom generators*. Let $\ell \colon \mathbb{N} \to \mathbb{N}$. We say that a family of functions $\{G_n\}_{n \geq 1}$ is a *quick infinitely often* (s, ε)*-pseudorandom generator against uniform quantum circuits* of *seed length* $\ell(n)$ and *output length* $m(n)$ if the following conditions hold:

- (i) Stretch: Each function *stretches* an $\ell(n)$-bit input to $m(n)$-bits i.e., $G_n \colon \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$.
- (ii) Uniformity and Running Time: There is a deterministic algorithm A that when given 1^n and $x \in \{0,1\}^{\ell(n)}$ runs in time $O(2^{\ell(n)})$ and outputs $G_n(x)$.
- (iii) Pseudorandomness: For each $n \geq 1$, let $\mathcal{D}_{m(n)} = G_n(\mathcal{U}_{\ell(n)})$ be the distribution over $m(n)$-bit strings induced by evaluating G_n on a random $\ell(n)$-bit string. Then the corresponding ensemble $\{\mathcal{D}_{m(n)}\}_{n \geq 1}$ is infinitely often (s, ε)-pseudorandom against uniform quantum circuits.

For convenience, we will also say in this case that

$$G = \{G_n\}_{n \geq 1} \text{ is an infinitely often } (\ell, m, s, \varepsilon)\text{-generator.}$$

Under our notation, observe that $\ell(n)$ and $m(n)$ are functions indexed by n that control the "stretch" of G_n (i.e. G_n maps ℓ bits to m bits), and $s(m)$ and $\varepsilon(m)$ are pseudorandomness parameters of the induced distribution $\mathcal{D}_m = G_n(\mathcal{U}_\ell)$.

2.7 Inherently probabilistic computations

As an intermediate model between classical and quantum computations, it will be useful to consider *inherently probabilistic circuits*. Roughly speaking, these are computational devices whose randomness remains "inacessible", in the sense that there is no simple way of decomposing an inherently probabilistic circuit as a distribution of deterministic circuits.

To make this point more precise, we review the discussion from Section 1.2.2. Consider the standard model of randomized Boolean circuits. In other words, we consider a Boolean circuit $C(x, y)$, where x is the input string and y is the random string, and say that C computes a Boolean

function $g\colon \{0,1\}^m \to \{0,1\}^k$ on an input $x \in \{0,1\}^m$ if $\Pr_y[C(x,y) = g(x)] \geq 2/3$. Similarly, we can also discuss the correlation between C and g, captured by $\gamma \stackrel{\text{def}}{=} \Pr_{x,y}[C(x,y) = g(x)]$. An extremely useful trick in this model is that, by an averaging argument, there must exist a fixed string y' such that $\Pr_x[C(x,y') = g(x)] \geq \gamma$. In other words, there is a *deterministic* circuit $C_{y'}(x) \stackrel{\text{def}}{=} C(x,y')$ that correctly computes g on a γ fraction of inputs. This often allows one to reduce the analysis of randomized Boolean circuits to the deterministic case. Additionally, $C_{y'}$ "forces" $C(x,y)$ to commit to a single consistent output on every x, which can be relevant if C is needed as a subroutine in other computations.

As a concrete example, note that this idea is crucial in the proof of $\mathsf{BPP} \subseteq \mathsf{P/poly}$, which is a simple combination of reducing the failure probability of a randomized circuit and fixing its random input.

While quantum computations share superficial similarities with the model of randomized computations and their corresponding randomized circuits, there is no obvious way of reducing a quantum computation to a distribution of "deterministic" quantum computations. In some cases, this is part of the difficulty when extending classical results to the quantum setting. With this in mind, below, we introduce inherently probabilistic circuits, which serve as a useful intermediate model in our analysis of quantum computations in the context of error correction and hardness amplification.

Inherently probabilistic circuits. We adopt a definition of inherently probabilistic computations that is sufficient for our purposes. In order to be as general as possible and because of how we access these computations, we model an *inherently probabilistic circuit* $\mathcal{A}$ over m input bits and that produces ℓ output bits as a function $\mathcal{A}\colon \{0,1\}^m \to \mathfrak{F}$, where $\mathfrak{F}$ is the set of probability distributions supported over a fixed domain $\{0,1\}^\ell$. In other words, $\mathcal{A}$ assigns to each input $z \in \{0,1\}^m$ a distribution $\mathcal{A}(z)$ supported over $\{0,1\}^\ell$. We say that $\mathcal{A}$ computes a function $g\colon \{0,1\}^m \to \{0,1\}^\ell$ with probability at least ε if

$$\Pr_{\substack{z\sim\{0,1\}^m,\\ v\sim\mathcal{A}(z)}}[v = g(z)] \geq \varepsilon.$$

Note that nothing is assumed about the relation between distributions $\mathcal{A}(z_1)$ and $\mathcal{A}(z_2)$ for $z_1 \neq z_2$, and that there is no way of globally "fixing" the randomness of $\mathcal{A}$ across different input strings.

We will also need to employ inherently probabilistic computations as subroutines inside standard deterministic and randomized computations. We model this situation using standard Boolean circuits and "inherently probabilistic oracles gates", described next.

Deterministic Boolean circuits with oracle gates are defined in the standard way. More precisely, we use $C^{\mathcal{O}}$ to represent a deterministic circuit that in addition to its original gates can label certain gates of fan-in m and of fan-out ℓ by $\mathcal{O}$. The computation of $C^{\mathcal{A}}(x)$ on an input x is defined once we set $\mathcal{O}$ to an inherently probabilistic circuit $\mathcal{A}$. Formally, each gate of $C^{\mathcal{A}}(x)$ is now a *random variable* whose value depends on the input x and on the outcomes of the calls to $\mathcal{A}$, each distributed according to $\mathcal{A}(z)$ when the input to $\mathcal{A}$ is the string z (two calls of $\mathcal{A}$ over the same string z are distributed independently). For convenience, if $C^{\mathcal{O}}$ is an oracle circuit with k designated output gates, we use $C^{\mathcal{A}}(x)$ to denote the random variable supported over $\{0,1\}^k$ and distributed according to the bits produced by its output gates (under a fixed order of the output gates of $C^{\mathcal{O}}$).

If $g\colon \{0,1\}^n \to \{0,1\}^k$, $C^{\mathcal{O}}$ is a Boolean oracle circuit over n input bits with oracle gates of fan-in m and fan-out ℓ, and $\mathcal{A}$ is inherently probabilistic, we say that $C^{\mathcal{A}}$ computes g with probability at least ε if $\Pr_{x\sim\{0,1\}^n,\mathcal{A}}[C^{\mathcal{A}}(x) = g(x)] \geq \varepsilon$.

It will also be useful to consider *randomized* oracle circuits with access to an *inherently probabilistic oracle* $\mathcal{A}$. In this setting, $C^{\mathcal{O}}$ has, in addition to its input string x, an extra input y for random bits. The computation of $C^{\mathcal{A}}$ on an input pair (x,y) is defined as above, i.e., the inputs

x and y are considered as a single input string of $C^{\mathcal{A}}$. We can similarly consider the probability $\Pr_{x,y,\mathcal{A}}[C^{\mathcal{A}}(x,y) = g(x)]$ of computing g using $C^{\mathcal{A}}$. Note in this case that y can be fixed to a given string y', but the randomness associated with each call to $\mathcal{A}$ cannot be fixed.

When it is clear from the context, we might omit superscripts $\mathcal{O}$ and $\mathcal{A}$ when referring to such oracle circuit computations.

A somewhat similar model is explored by Kearns and Schapire [KS94] (see also [Yam92]) from a different perspective.[9] Their paper focuses on the learnability of functions of the form $c\colon X \to [0,1]$, where $c(x)$ can be interpreted as a probability distribution over $\{0,1\}$. In contrast, in this work we assign to each input x a *probability distribution over strings*. Most importantly, we are concerned with the subtleties of *computing* with such devices, which will be used as internal building blocks in other computations, while the focus of [KS94] and [Yam92] is on the *learnability* of probabilistic/stochastic functions.

We notice that any quantum circuit that is accessed through *classical queries* can be viewed as an inherently probabilistic circuit.

Lemma 2.17. *Let $\mathcal{R}$ be an inherently probabilistic circuit and $\mathcal{Q}$ a quantum circuit such that on every input x, the output distributions of $\mathcal{R}(x)$ and $\mathcal{Q}(x)$ are exactly the same.*[10] *Then for every probabilistic algorithm $\mathcal{A}$, which might have some classical input w, that is allowed to make (classical) queries to $\mathcal{R}$ or $\mathcal{Q}$, we have that for every y,*

$$\Pr_{\mathcal{A},\mathcal{R}}[\mathcal{A}^{\mathcal{R}}(w) = y] = \Pr_{\mathcal{A},\mathcal{Q}}[\mathcal{A}^{\mathcal{Q}}(w) = y].$$

Proof. Notice that since $\mathcal{A}$ is a classical algorithm and is only allowed to make classical queries to $\mathcal{Q}$, $\mathcal{A}$ only has access to samples of the output distribution of $\mathcal{Q}$. Since the output distribution of $\mathcal{Q}$ and $\mathcal{R}$ is exactly the same for every input, the result follows. □

3 Lower bounds from learning algorithms: a modular approach via PRGs

In this section, we show how to derive circuit lower bounds from quantum learning algorithms via two steps: first, in Section 3.1 we show that quantum learners imply quantum natural properties; then, in Section 3.2 we show that given a pseudorandom generator against uniform quantum computation, quantum natural properties imply circuit lower bounds.

3.1 Quantum natural properties from quantum learning algorithms

Theorem 3.1. *There is a universal constant $\lambda \geq 1$ for which the following holds. Let $\gamma\colon \mathbb{N} \to [0,1/2]$ be a constructive function such that $\gamma(n) \geq \lambda \cdot 2^{-n/2}$, and let $s(n) = \gamma(n)^2 \cdot 2^n/(\lambda \cdot n)$. If $\mathcal{C}[n^k]$ can be learned under the uniform distribution in quantum time at most $s(n)$ by a (δ,ε)-learner with $\delta(n) \leq 0.99$ and $\varepsilon(n) \leq 1/2 - \gamma(n)$, then there exists a promise quantum natural property against $\mathcal{C}[n^k]$.*

Proof. First, we prove the result under the assumption that $\delta \leq 1/10$. Then we show that this can be easily relaxed to learners that only succeed with probability $1/100$, as in the statement of the result.

[9] This reference has been brought to our attention by an anonymous reviewer.

[10] Notice that here $\mathcal{Q}(x)$ could also have a garbage register that is traced out.

By our assumption, there exists a quantum learning algorithm $\mathcal{A}^{|f\rangle}$ that runs in quantum time $s(n)$ for a function $f : \{0,1\}^n \to \{0,1\}$ computed by a circuit $f \in \mathcal{C}[n^k]$, and with probability at least $9/10$, the algorithm $\mathcal{A}^{|f\rangle}$ outputs a quantum hypothesis U such that

$$\mathop{\mathbb{E}}_{x}\left[\|\Pi_{f(x)}U\,|x,0\rangle\|^2\right] \geq \frac{1}{2} + \gamma(n)\,. \tag{2}$$

We show now how to construct a quantum natural property from $\mathcal{A}^{|f\rangle}$. Let $\mathcal{D}$ be the quantum algorithm that receives as input $\mathtt{tt}(f)$, the truth-table of f, and performs the following steps.

1. Simulate $\mathcal{A}^{|f\rangle}$ by answering all the queries $\sum_x \alpha_x\,|x,b\rangle \to \sum_x \alpha_x\,|x,b\oplus f(x)\rangle$ for arbitrary amplitudes $\{\alpha_x\}_x$ (this can be performed efficiently since $\mathcal{D}$ has access to the truth table of f).
2. If the simulation of $\mathcal{A}^{|f\rangle}$ does not output a hypothesis U, output 1.
3. Otherwise, set $T = \gamma(n)^{-3} + 100n$ and choose $x_1,\ldots,x_T \in \{0,1\}^n$ uniformly at random, invoke $U\,|x_i,0\rangle$ and measure the first bit for every $i \in [T]$; denote by $b_i \in \{0,1\}$ the output of the ith invocation.
4. Output 0 *if and only if*
$$\frac{1}{T}\sum_{i\in[T]}[b_i = f(x_i)] \geq \frac{1}{2} + \frac{1}{4}\cdot\gamma(n)\,. \tag{3}$$

We prove that $\mathcal{D}$ satisfies all three conditions of Definition 2.12. Constructiveness follows by noting that simulating $\mathcal{A}^{|f\rangle}$ can be done in $\mathrm{poly}(2^n)$ time, and $\mathcal{D}$ performs $T = \gamma(n)^{-3} + 100n$ invocations of the hypothesis U and a check that requires summing over T terms.

We proceed to show the hardness condition. Fix $f \in \mathcal{C}[n^k]$. We show that the algorithm $\mathcal{D}$ accepts $\mathtt{tt}(f)$ with probability at most $1/3$. By our assumption, with probability at least $9/10$, the learning algorithm $\mathcal{A}$ outputs U that satisfies Eq. (2). Suppose this is the case. Then, for every $i \in [T]$, let $q_i = \Pr[b_i = f(x_i)]$ (where the probability is taken over the randomness of x_i and algorithm $\mathcal{A}$) and note that $q_i \geq 1/2 + \gamma(n)$. By the Chernoff bound (in Theorem 2.1), we have

$$\Pr\left[\frac{1}{T}\sum_i [b_i = f(x_i)] < \frac{1}{2} + \frac{1}{4}\cdot\gamma(n)\right] \leq \exp\left(-\frac{\gamma(n)^2 T}{T(\frac{1}{2}+\gamma(n)) + \frac{T\gamma(n)}{12}}\right) \leq \exp\big(-O(1/\gamma(n))\big)\,, \tag{4}$$

where in the last inequality we use the fact that $T = \gamma(n)^{-3} + 100n$. Thus, with probability at least $(9/10)\cdot\big(1-\exp\big(-1/\gamma(n)\big)\big) \geq 2/3$, the algorithms $\mathcal{D}$ rejects.

We proceed to show density. For that, let us fix an arbitrary quantum circuit U of size $s(n)$. Also, for a fixed f, x, let $W_{f,x} = \Pr_U[U(x) = f(x)]]$. Then, we have that

$$\mathop{\mathbb{E}}_{f,x}[W_{f,x}] = \frac{1}{2},$$

since for any fixed x, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{f}\left[\mathop{\mathbb{E}}_{U}[U(x) = f(x)]\right] &= \frac{1}{2}\mathop{\mathbb{E}}_{f}\left[\mathop{\mathbb{E}}_{U}\left[U(x) = 0 | f(x) = 0\right] + \mathop{\mathbb{E}}_{U}\left[U(x) = 1 | f(x) = 1\right]\right] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{f}\left[\mathop{\mathbb{E}}_{U}\left[U(x) = 0\right] + \mathop{\mathbb{E}}_{U}\left[U(x) = 1\right]\right] \\
&= \frac{1}{2},
\end{aligned}$$

where in the second equality we have that the fact $\Pr_U[U(x) = b]$ is independent of f. We now consider drawing an independent random function f and use the Chernoff Bound (see Lemma 2.2) to bound the random variable $\sum_x W_{f,x}$.

$$\Pr_f\Big[\Big|\sum_{x\in\{0,1\}^n} W_{f,x} - \mathop{\mathbb{E}}_f\big[\sum_{x\in\{0,1\}^n} W_{f,x}\big]\Big| \geq t\Big] \leq \exp\Big(-\frac{t^2}{2(t/3 + 2^n\cdot\frac{1}{2})}\Big). \tag{5}$$

Setting $t = \gamma(n)\cdot 2^n/8$, we have,

$$\Pr_f\Big[\Big|\sum_x W_{f,x} - 2^n/2\Big|\Big| \geq \gamma(n)\cdot 2^n/8\Big] \leq \exp\Big(-\frac{\gamma(n)^2 2^{2n}/64}{2(\gamma(n)\cdot 2^n/24 + 2^n/2)}\Big),$$

which implies

$$\Pr_f\Big[\Big|\mathop{\mathbb{E}}_x[W_{f,x}] - 1/2\Big|\Big| \geq \gamma(n)/8\Big] \leq \exp\Big(-\frac{\gamma(n)^2 2^n}{64(\gamma(n)/12+1)}\Big).$$

This means that for every quantum circuit U (regardless its size), the fraction of functions that can be computed by it with advantage $\gamma(n)/8$ on a random x is at most $\exp\Big(-\frac{\gamma(n)^2 2^n}{64(\gamma(n)/12+1)}\Big)$.

We now count the number of unitaries of size $s(n)$. Observe that there exists a constant $\alpha < 50$ such that there exist at most $2^{\alpha s(n)\log(s(n))}$ circuits of size $s(n)$ if we fix, for example, the universal gateset $\{\mathsf{H}, \mathsf{Toff}\}$ as described in Section 2.2: this is true because one can define, for the i-th gate, if the gate is a Hadamard or a Toffoli, which requires just a single bit, and it takes $O(\log s(n))$ bits to denote the qubits on which the gate acts. Therefore, using a union bound, we have that the fraction of functions that can be $\gamma(n)/8$ approximated by $s(n)$ circuits is at most

$$\exp\Big(-\frac{\gamma(n)^2 2^n}{64(\gamma(n)/12+1)} + \alpha s(n)\log s(n)\Big) \leq \exp\Big(\gamma(n)^2 2^n\Big(-\frac{1}{64(\gamma(n)/2+1)} + \frac{O(n)}{\lambda\cdot n}\Big)\Big) < 1/2,$$

where the final inequality used that $\lambda \geq 1$ is a sufficiently large constant as in the theorem statement. To conclude the proof, finally observe that for a function f that cannot be approximated by an $s(n)$-sized circuit U, i.e., for all $s(n)$-sized circuits U

$$\Pr_{x,U}[U(x) = f(x)] < \frac{1}{2} + \frac{\gamma(n)}{8},$$

by the Chernoff bound (in Theorem 2.1), we have that by picking $x_1, \ldots, x_T$ uniformly at random and $b_i \sim U(x)$, we obtain

$$\Pr\Big[\frac{1}{T}\sum_i [b_i = f(x_i)] \geq \frac{1}{2} + \frac{1}{4}\cdot\gamma(n)\Big] \leq \exp\big(-O(1/\gamma(n))\big)\,. \tag{6}$$

Hence the distinguisher $\mathcal{D}$ accepts it with overwhelming probability. This completes the proof under our initial assumption that $\delta \leq 1/10$.

In order to obtain a promise quantum natural property from learners that succeed with probability $1/100$, we can modify the construction above as follows. The quantum algorithm $\mathcal{D}$ simply runs steps (1)-(3) above $\ell = O(1)$ times, obtaining hypotheses $U_1, \ldots, U_\ell$, and outputs 0 if and only if at least one of them satisfies the condition in step (4). It is not hard to see that the analysis presented above still holds under this modification, provided that ℓ is a large enough constant. □

3.2 Circuit lower bounds from quantum natural properties

In this section, we show that promise quantum natural properties imply circuit lower bounds. This connection between quantum natural properties and circuit lower bounds is demonstrated using pseudorandom generators against uniform circuits.

The key technical component we shall need is the following theorem, which we prove in Section 5, that shows a quantum-secure PRG conditioned on the assumption that $\mathsf{PSPACE} \not\subseteq \mathsf{BQSUBEXP}$.

Theorem 3.2 (Conditional PRG against uniform quantum computations)**.** *Suppose that there is a language $L \in \mathsf{PSPACE}$ and $\gamma > 0$ such that $L \notin \mathsf{BQTIME}[2^{n^\gamma}]$. Then, for some choice of constants $\alpha \geq 1$ and $\lambda \in (0, 1/5)$, there is an infinitely often $(\ell, m, s, \varepsilon)$-generator $G = \{G_n\}_{n\geq 1}$, where $\ell(n) \leq n^\alpha$, $m(n) = \lfloor 2^{n^\lambda} \rfloor$, $s(m) = 2^{n^{2\lambda}} \geq \mathsf{poly}(m)$ (for any polynomial), and $\varepsilon(m) = 1/m$.*

We shall also need the following diagonalization lemma.

Lemma 3.3 ([Diagonalization Lemma, see e.g. [OS17, Corollary 2])**.** *For every $k \in \mathbb{N}$, there exists a language $L_k \in \mathsf{PSPACE}$ such that L_k cannot be computed by Boolean circuits of size n^k whenever n is sufficiently large.*

Recall that $\mathsf{E} \stackrel{\text{def}}{=} \mathsf{DTIME}[2^{O(n)}]$ and $\mathsf{BQE} \stackrel{\text{def}}{=} \mathsf{BQTIME}[2^{O(n)}]$. Our goal for the rest of this section is to establish the following connection between quantum natural properties and circuit lower bounds.

Theorem 3.4 (Connection between natural properties and lower bounds)**.** *For every circuit class $\mathcal{C}$ that is closed under restrictions of input variables to constants 0 and 1, at least one of the following lower bounds holds:*

1. *For every $k \in \mathbb{N}$ there exists a language $L \in \mathsf{BQE}$ that cannot be computed by general Boolean circuits of size n^k, for sufficiently large input lengths n.*
2. *There exists a universal constant $\upsilon \geq 1$ for which the following holds. If there is a promise quantum natural property against $\mathcal{C}[n^a]$, for $a > 1$, then there exists a language $L \in \mathsf{E}$ that cannot be computed by $\mathcal{C}$-circuits of size $n^{a/\upsilon}$, for infinitely many input lengths n.*

Proof. The proof is via a win-win argument. Starting with the easy case, suppose that for every $\gamma > 0$ it holds that $\mathsf{PSPACE} \subseteq \mathsf{BQTIME}[2^{n^\gamma}]$. Fix $k \in \mathbb{N}$. By Lemma 3.3, there exists a language L_k that cannot be computed by Boolean circuits of size n^k (for any sufficiently large input length n). By our assumption, the language L_k can be computed by a quantum algorithm of time complexity 2^{n^γ}, and so $L_k \in \mathsf{BQE}$, and the lower bound in Item (1) of Theorem 3.4 holds.

Otherwise, suppose that there exists a language $L \in \mathsf{PSPACE}$ and $\gamma > 0$ such that $L \notin \mathsf{BQTIME}[2^{n^\gamma}]$. We show that this implies the lower bound in Item (2). Indeed, under this assumption, by Theorem 3.2, there exist constants $\alpha \geq 1$ and $\lambda \in (0, 1/5)$ such that there is an infinitely often $(\ell, m, s, \varepsilon)$-generator $\mathcal{G} = \{G_n\}_{n\geq 1}$, where $\ell(n) \leq n^\alpha$, $m(n) = \lfloor 2^{n^\lambda} \rfloor$, $s(m) = 2^{n^{2\lambda}} \geq \mathsf{poly}(m)$, and $\varepsilon(m) = 1/m$; that is,

(i) Each G_n is a function mapping $\{0,1\}^{\ell(n)}$ to $\{0,1\}^{m(n)}$.

(ii) There is a deterministic algorithm A that when given 1^n and $x \in \{0,1\}^{\ell(n)}$ runs in time $O(2^{\ell(n)})$ and outputs $G_n(x)$.

(iii) For each $n \geq 1$, let $\mathcal{D}_{m(n)} = G_n(\mathcal{U}_{\ell(n)})$ be the distribution over $m(n)$-bit strings induced by evaluating G_n on a random $\ell(n)$-bit string. Then the corresponding ensemble $\{\mathcal{D}_{m(n)}\}_{n\geq 1}$ is infinitely often (s, ε)-pseudorandom against uniform quantum circuits.

We define a language $L^{\mathcal{G}}$ using the generator $\mathcal{G} = \{G_n\}_{n\geq 1}$ and argue that it is hard to decide this language due to the *existence* of a quantum natural property, as assumed in Item (2) of the statement. Our language $L^{\mathcal{G}}$ is defined by the following deterministic algorithm D:

1. Encoding check: On an input $u \in \{0,1\}^n$, reject if u is not of the form

$$u = (1^t, w, x)$$

for some $t \in \mathbb{N}$ such that $|w| = \ell(t)$ and $|x| = t'$, where we denote $t' = \lfloor t^\lambda \rfloor$. (Note that $2^{t'}$ is the maximal power of 2 that is not greater than $m(t)$.)[11]

2. PRG invocation: Compute $G_t(w) \in \{0,1\}^{m(t)}$ and let g_w denote its leftmost $2^{t'}$ bits.

3. Emulate computation: Let $h_w \colon \{0,1\}^{t'} \to \{0,1\}$ be the function determined by g_w (i.e., fnc^{g_w}), and output $h_w(x)$.

To prove the lower bound in Item (2) of Theorem 3.4, we first observe that D runs in deterministic time complexity $2^{O(n)}$, and hence $L^{\mathcal{G}} \in \mathsf{E}$. To see that, observe that by construction, the runtime of D is dominated by Step (2), which computes $G_t(y)$ given a correctly encoded input. Note that for valid inputs we have $n \geq t + \ell(t) + t'$. As $\ell(t) < n$, the time complexity of D is at most $O(2^n)$.

To complete the proof, we need to prove the following lemma, which shows that the language $L^{\mathcal{G}}$ is hard for $\mathcal{C}$-circuits. Recall that we are under the assumption that there is a promise quantum natural property against $\mathcal{C}[n^a]$, where $a > 1$ is arbitrary but can be assumed to be sufficiently large without loss of generality.

Lemma 3.5. *There exists a constant $\upsilon = \upsilon(\alpha, \lambda) \geq 1$, where (α, λ) are the parameters of the* PRG *$\mathcal{G}$, such that $L^{\mathcal{G}} \notin \mathcal{C}\left[n^{\frac{a}{\upsilon}}\right]$ for infinitely many input lengths n.*

Proof. Set $\upsilon = 2 \cdot \frac{\alpha}{\lambda}$, and let $a > 1$. We prove the lemma by contradiction: suppose that there exists a finite set $\mathcal{N}$ such that for every $n \in \mathbb{N} \setminus \mathcal{N}$ it holds that $L^{\mathcal{G}} \in \mathcal{C}\left[n^{\frac{a}{\upsilon}}\right]$ over length-n inputs. We argue next that this circuit upper bound and the existence of a quantum natural property as in the statement of the theorem lead to a violation of property (*iii*) of $\mathcal{G}$ stated above.

Let F be a quantum algorithm that gets as input a string of length $N = 2^n$ and computes a (promise) quantum natural property in the sense of Definition 2.12 in $\mathrm{poly}(N)$ time. Recall that F accepts a dense set Γ_n of inputs and rejects functions computable by $\mathcal{C}$-circuits of size n^a, both with probability at least $2/3$. Performing error reduction on F using amplitude amplification, we can assume without loss of generality that on each string $z \in \Gamma_n \subseteq \{0,1\}^N$, we have $\|\Pi_1 F\,|z,0\rangle\|^2 \geq 1 - 2^{-2N}$, and on each string $z \in \{0,1\}^N$ such that $\mathsf{fnc}^z \in \mathcal{C}_n[n^a]$, we have $\|\Pi_1 F\,|z,0\rangle\|^2 \leq 2^{-2N}$.

We construct a deterministic algorithm $A(1^m)$ to generate quantum circuits that violate property (*iii*) of $\mathcal{G}$ (parameterized here by a parameter t). On an input 1^m, where $m = m(t) = \lfloor 2^{t^\lambda} \rfloor$, A writes $m = 2^{t'} + r$, where $r \in [0, 2^{t'} - 1]$. It outputs a quantum circuit C_F on m input bits that computes according to F applied to the $2^{t'}$-bit prefix of the input and ignores the remaining r input bits. To complete the proof of the lemma, we show that $A(1^m)$ outputs a quantum circuit

[11] Notice that there exists at most one value of t for which these constraints hold.

C_F of size at most $s(m) = 2^{t^{2\lambda}}$ that $\Omega(1)$-distinguishes $G_t(\mathcal{U}_{\ell(t)})$ from $\mathcal{U}_{m(t)}$ for every large enough t, thereby violating property (iii). In other words,

$$\left| \mathop{\mathbb{E}}_{y \sim \mathcal{U}_{m(t)}} [C_F(y) = 1] - \mathop{\mathbb{E}}_{\beta \sim \mathcal{U}_{\ell(t)}} [C_F(G_t(\beta)) = 1] \right| \geq \Omega(1). \tag{7}$$

For each $m = 2^{t'} + r$ of the form defined before,

(a) C_F rejects with overwhelming probability every m-bit string z whose $2^{t'}$-bit prefix encodes a function $f \in \mathcal{C}[(t')^a]$.

(b) C_F accepts with overwhelming probability every m-bit string z whose $2^{t'}$-bit prefix is in $\Gamma_{t'}$.

(c) $|C_F| = O(\text{poly}(2^{t'})) < s(m)$ as its run time is dominated by the size of F which runs in $\text{poly}(2^{t'})$ time on inputs of size $2^{t'}$.

Note that, as a consequence of the density of $\Gamma_{t'}$ and (b), we have

$$\mathop{\mathbb{E}}_{z \sim \mathcal{U}_{m(t)}} [C_F(z) = 1] = \mathop{\mathbb{E}}_{y \sim \mathcal{U}_{2(t')}} [F(y) = 1] \geq \frac{1}{2} \cdot \mathop{\mathbb{E}}_{y \sim \mathcal{U}_{2(t')}} \left[F(y) = 1 \mid y \in \Gamma_{t'} \right] \geq \frac{1}{2} \cdot (1 - 2^{-2N}).$$

To satisfy Equation (7), it is enough to argue that for each seed $w \in \{0,1\}^{\ell(t)}$ and the string $\tau \stackrel{\text{def}}{=} G_t(w)$, we have $\Pr_\tau[C_F(\tau) = 1] \leq 2^{-2N}$. According to Item (a) above, it suffices to show that g_τ, the $2^{t'}$-prefix of τ, is the truth-table of a function $h_w \in \mathcal{C}[(t')^a]$. For this we rely on the assumption that $L^{\mathcal{G}} \in \mathcal{C}[n^{\frac{a}{\upsilon}}]$ for every input length $n \in \mathbb{N} \setminus \mathcal{N}$. Details follow.

Let $n = t + \ell(t) + t' + O(1) \leq 2t^\alpha < 2(t')^{\alpha/\lambda}$ such that $n \in \mathbb{N} \setminus \mathcal{N}$, where n is an input length of $L^{\mathcal{G}}$ that contains inputs of the following form: $(1^t, w, x)$, where x is an arbitrary t'-bit string, and w is the seed that generates τ. Let g_τ be the $2^{t'}$-bit prefix of τ and $h_\tau \stackrel{\text{def}}{=} \mathsf{fnc}^{g_\tau}$ be a Boolean function on $t' < n$ inputs bits. Then, by the assumption that $L^{\mathcal{G}} \in \mathcal{C}[n^{\frac{a}{\upsilon}}]$ where $\mathcal{C}$ is a circuit class closed under restriction of input variables to constants 0 and 1 (see Item (ii) in Section 2.1.2), we obtain a $\mathcal{C}$-circuit that computes $h_\tau(x)$. Furthermore, this circuit is of size at most

$$n^{a/\upsilon} \leq (2t^\alpha)^{a/\upsilon} < \left(2\left(t'^{\frac{\alpha}{\lambda}}\right)\right)^{a/\upsilon} \leq 2^{a/\upsilon} \cdot (t')^{\frac{\alpha}{\lambda} \cdot \frac{a}{\upsilon}} < (t')^{2 \cdot \frac{\alpha}{\lambda} \cdot \frac{a}{\upsilon}} = (t')^a,$$

where we have used that $2z \leq z^2$ for $z \geq 2$ in the penultimate inequality and $2 \cdot \frac{\alpha}{\lambda} \cdot \frac{a}{\upsilon} \leq a$ as $\upsilon = 2 \cdot \frac{\alpha}{\lambda}$ in the last equality. □

This concludes the proof of Theorem 3.4. □

As an immediate corollary, we obtain the following theorem.

Corollary 3.6 (Lower bounds from quantum natural properties)**.** *Let $\mathcal{C}$ be a circuit class. Suppose that for every $a \geq 1$ there is a promise quantum natural property against $\mathcal{C}[n^a]$. Then for every constant $b \geq 1$ we have $\mathsf{BQE} \not\subseteq \mathcal{C}[n^b]$.*

Proof. From Theorem 3.4, note that there are two possibilities and at least one of them holds.

In the first case, for every $k \in \mathbb{N}$, there exists a language $L \in \mathsf{BQE}$ that cannot be computed by general Boolean circuits of size n^k. Since by assumption $\mathcal{C}$-circuits can be simulated by general Boolean circuits with only a polynomial blowup on circuit size (Section 2.1.2), it follows in this case that for every $b \geq 1$ we have $\mathsf{BQE} \not\subseteq \mathcal{C}[n^b]$.

In the second case, given $b \geq 1$, we let $a = b\upsilon$. Since by assumption there is a natural property against $\mathcal{C}[n^a]$, the second case of Theorem 3.4 gives a language in E that is not in $\mathcal{C}[n^{a/\upsilon}]$. Since $\mathsf{E} \subseteq \mathsf{BQE}$, the result follows. □

3.3 Non-trivial quantum learning yields non-uniform circuit lower bounds

Theorem 3.7 (Theorem 1, formally restated)**.** *There is a universal constant $\lambda \geq 1$ for which the following holds. Let $\mathfrak{C}$ be a concept class. Let $\gamma\colon \mathbb{N} \to [0,1/2]$ and $T : \mathbb{N} \to \mathbb{N}$ be arbitrary constructive functions with $\gamma(n) \geq \lambda \cdot 2^{-n/2}$ and $T(n) \leq \gamma(n)^2 \cdot 2^n/\lambda n$ for large enough n. Suppose that, for every $k \geq 1$, the class $\mathfrak{C}[n^k]$ can be learned in quantum time $T(n)$ with probability $\geq 1/100$ and advantage $\gamma(n)$. Then, for every $k \geq 1$, we have $\mathsf{BQE} \not\subseteq \mathfrak{C}[n^k]$.*

Proof. This proof combines results from the previous two sections to relate quantum learning algorithms to circuit lower bounds through the existence of quantum natural properties. The assumptions for the algorithm that learns $\mathfrak{C}[n^k]$ imply that it runs in time at most $T(n) \leq \gamma(n)^2 \cdot 2^n/\lambda n$, has a confidence probability $1 - \delta(n) \geq 1/100$ (i.e., $\delta \leq 0.99$) and error probability $\varepsilon(n) \leq 1/2 - \gamma(n)$. Then, from Theorem 3.1, for every constant $k \geq 1$, there is a promise quantum natural property against $\mathfrak{C}[n^k]$. Following our notation from Section 2.1.2, the concept class $\mathfrak{C}[n^k]$ also denotes a corresponding circuit class. Using Corollary 3.6, the existence of a quantum natural property against $\mathfrak{C}[n^k]$ for every $k \geq 1$ implies that for every constant $b \geq 1$, $\mathsf{BQE} \not\subseteq \mathfrak{C}[n^b]$. This completes the proof. □

Similarly to previous work [OS17], our arguments can be adapted to show a relation between the non-trivial learnability of a concept class $\mathcal{C}[s]$ of size-s circuits, where $s(n) = n^{\omega(1)}$, and lower bounds against $\mathcal{C}[s']$, where $s, s'\colon \mathbb{N} \to \mathbb{N}$ and s' is a function that depends on s. We have decided not to pursue the most general form of the result in this paper, as our proofs are already significantly involved and the relation between s and s' deteriorates as s becomes super-quasi-polynomial, due to the use of win-win arguments.

4 Technical tools

In this section, we discuss extensions to quantum computing of several fundamental results from complexity theory. This is needed to establish the correctness of our PRG construction.

In Section 4.1, we provide a (fairly straightforward) quantization of [NW94], where we show that a quantum distinguisher for the (candidate) PRG NW^f implies that f can be non-trivially approximated by quantum algorithms. In Section 4.2, we give a self-contained exposition of a quantum analogue of the Goldreich-Levin algorithm [GL89] discovered by [AC02]. In Section 4.3, we quantise the near-optimal uniform hardness amplification result of [IJKW10]. We remark that this is the most technically involved result in this section. Finally, in Section 4.4, we show how to use downward and random self-reducibility of languages to devise quantum algorithms to compute them, adapting ideas from [IW01].

4.1 Nisan-Wigderson generator against quantum adversaries

An ordered family $\mathcal{S} = (S_1, \ldots, S_t)$ of sets is a (t, m, n, α)-*design* if the following conditions are satisfied:

- $|\mathcal{S}| = t$, and for each $i \in [t]$, $S_i \subseteq [m]$ and $|S_i| = n$.
- For every distinct pair $i, j \in [t]$, $|S_i \cap S_j| \leq \alpha$.

Lemma 4.1 ([NW94, Lemma 2.5])**.** *There is an absolute constant $c \geq 1$ for which the following holds. For any positive integers n and t such that $n \leq t \leq 2^n$, there is an ordered family $\mathcal{S}$ that is a $(t, cn^2, n, \log t)$-design. Moreover, given n, t, and an index $i \in [t]$, the set S_i can be output in time $\mathsf{poly}(t)$.*

Definition 4.2 (Nisan-Wigderson generator [NW94])**.** *Let* $f\colon \{0,1\}^n \to \{0,1\}$*,* $m = cn^2$*, and* $n \le t \le 2^n$*. In addition, let* $\mathcal{S}$ *be a* $(t, m, n, \log t)$*-design. Then the* Nisan-Wigderson generator $\mathsf{NW}^f_{\mathcal{S}}\colon \{0,1\}^m \to \{0,1\}^t$ *is the function defined as*

$$\mathsf{NW}^f_{\mathcal{S}}(z) \stackrel{\text{def}}{=} f(z|_{S_1})f(z|_{S_2})\cdots f(z|_{S_t}),$$

where $z|_{S_i}$ *is the* n*-bit string formed by restricting* $z \in \{0,1\}^m$ *onto the coordinates given by the* i*-th set* $S_i \in \mathcal{S}$*.*

From now on we will only consider the designs given by Lemma 4.1, so in order to simplify notation, we omit the underlying collection $\mathcal{S}$ and simply write NW^f.

The next lemma verifies that the usual analysis of the Nisan-Wigderson generator extends to inherently probabilistic circuits.

Lemma 4.3 (Uniformity and correctness of the NW generator for inherently probabilistic circuits)**.** *Let* n *and* t *be positive integers such that* $1 \le n \le t \le 2^n$*,* $f\colon \{0,1\}^n \to \{0,1\}$*, and consider the corresponding function* $\mathsf{NW}^f\colon \{0,1\}^m \to \{0,1\}^t$*, where* $m = cn^2$*. Let* $\mathcal{D}$ *be an inherently probabilistic circuit defined over* t *input bits that satisfies*

$$\left| \Pr_{s\in\{0,1\}^m,\,\mathcal{D}} \left[\mathcal{D}(\mathsf{NW}^f(s)) = 1\right] - \Pr_{y\in\{0,1\}^t,\,\mathcal{D}} \left[\mathcal{D}(y) = 1\right] \right| \;\ge\; \gamma. \tag{8}$$

There exists a probabilistic algorithm $\mathcal{B}$ *with oracle access to* f *that, given an input* 1^t*, runs in time* $\mathsf{poly}(t)$ *and outputs a deterministic oracle circuit* $\mathcal{A}^{\mathcal{O}}$ *of size* $O(t^2)$ *for which the following holds. For any choice of* $\mathcal{D}$ *as above, with probability* $\Omega(\gamma/t^2)$ *over the internal randomness of* $\mathcal{B}^f$*, the generated circuit* $\mathcal{A}^{\mathcal{O}}$ *satisfies*

$$\Pr_{x\in\{0,1\}^n,\,\mathcal{D}} \left[\mathcal{A}^{\mathcal{D}}(x) = f(x)\right] \ge \frac{1}{2} + \frac{\gamma}{2t}. \tag{9}$$

Proof. We show that there exists a collection of deterministic oracle circuits $\mathcal{A}^{\mathcal{O}}$ such that, for any choice of $\mathcal{D}$, a noticeable fraction of such oracle circuits provide the desired advantage. The argument also establishes the existence of a uniform algorithm $\mathcal{B}^f$ with the required properties. Note that $\mathcal{B}^f$ does not have access to $\mathcal{D}$ and that the computation of each $\mathcal{A}^{\mathcal{O}}$ does not specify its oracle. However, in order to make the argument more concrete, in our exposition below we refer to the oracle $\mathcal{O}$ as $\mathcal{D}$, and consider a fixed $\mathcal{D}$ satisfying the assumption of the lemma. After that, we explain how the conclusion of the result follows from this argument.

We prove first that there exists a *fixed* $j \in [t]$ (depending on $\mathcal{D}$) and a sub-collection of deterministic oracle "next-bit predictor" circuits $\mathcal{P}^{\mathcal{D}}_{r,d}$ parameterized by $r \in \{0,1\}^{t-j}$ and a *fixed* $d \in \{0,1\}$ (depending on $\mathcal{D}$), each of size $O(t)$, such that

$$\Pr_{s\in\{0,1\}^m,r,\mathcal{D}} \left[\mathcal{P}^{\mathcal{D}}_{r,d}\big(\mathsf{NW}^f(s)_1, \ldots, \mathsf{NW}^f(s)_{j-1}\big) = \mathsf{NW}^f(s)_j\right] \ge 1/2 + \gamma/t. \tag{10}$$

For notational simplicity, from here onwards we abuse notation and write $z \sim \mathsf{NW}^f(s)$, meaning that $s \sim \{0,1\}^m$ is uniformly random and $z = \mathsf{NW}^f(s)$. Additionally, for $i \in [t]$ and $z \in \{0,1\}^t$, we let $z^{(i)}$ be a *random variable* with the first i bits being equal to z and the last $t-i$ bits being a uniformly random string $r \in \{0,1\}^{t-i}$. Also, let

$$p_i = \Pr_{z\sim\mathsf{NW}^f(s),\mathcal{D}} \left[\mathcal{D}(z^{(i)}) = 1\right],$$

and observe that

$$p_0 = \Pr_{y\in\{0,1\}^t,\mathcal{D}}[\mathcal{D}(y)=1] \quad \text{and} \quad p_t = \Pr_{z\sim \mathsf{NW}^f(s),\mathcal{D}}[\mathcal{D}(z)=1].$$

Therefore, by Eq. (8), we have $|p_t - p_0| \geq \gamma$. By the triangle inequality, there exists $j \in \{0,1,\ldots,t-1\}$ such that $|p_{j+1} - p_j| \geq \gamma/t$. Setting $\hat{z} = z^{(j)}$ for convenience, notice that,

$$\begin{aligned}
\gamma/t \;&\leq |p_{j+1} - p_j| \\
&= \left| \left(\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},z^{(j+1)}} \left[\mathcal{D}(z^{(j+1)})=1\right] - \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1\right] \right) \right| \\
&= \left| \left(\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} = z_{j+1}\right] \right.\right. \\
&\quad \left.\left. - \frac{1}{2} \left(\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} = z_{j+1}\right] + \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} \neq z_{j+1}\right] \right) \right) \right| \\
&= \left| \frac{1}{2} \left(\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} = z_{j+1}\right] - \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} \neq z_{j+1}\right] \right) \right| \\
&= \left| \frac{1}{2} \left(\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} = z_{j+1}\right] + \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})\neq 1 | \hat{z}_{j+1} \neq z_{j+1}\right] - 1 \right) \right|,
\end{aligned}$$

where in the second equality we use three simple facts:

$$\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},z^{(j+1)}} \left[\mathcal{D}(z^{(j+1)})=1\right] = \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} = z_{j+1}\right]$$

and

$$\begin{aligned}
&\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1\right] \\
&= \Pr_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}}[\hat{z}_{j+1} = z_{j+1}] \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 \middle| \hat{z}_{j+1} = z_{j+1}\right] \\
&\quad + \Pr_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}}[\hat{z}_{j+1} \neq z_{j+1}] \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 \middle| \hat{z}_{j+1} \neq z_{j+1}\right]
\end{aligned}$$

and $\Pr_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}}[\hat{z}_{j+1} = z_{j+1}] = \Pr_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}}[\hat{z}_{j+1} \neq z_{j+1}] = \frac{1}{2}$, since $\hat{z}_{j+1} = r_1$ is a uniformly random bit.

The inequality above motivates the following definition. Let $d \in \{0,1\}$ be such that $(-1)^d(p_{j+1} - p_j) > 0$. We define the "next-bit predictor" $\mathcal{P}^{\mathcal{D}}_{r,d}$, for $r \in \{0,1\}^{t-j}$, as follows. On input $z_1,\ldots,z_j$ (where $z \sim \mathsf{NW}^f(s)$), $\mathcal{P}^{\mathcal{D}}_{r,d}$ makes an oracle call to $\mathcal{D}$ on input $z_1,...,,z_j,r_1,...,r_{t-j}$ and let b be its output. If the output $b=1$, then $\mathcal{P}$ outputs $r_1 \oplus d$, otherwise $\mathcal{P}$ outputs $r_1 \oplus d \oplus 1$. It follows from the inequality above, the definition of $\mathcal{P}^{\mathcal{D}}_{r,d}$, and a simple manipulation (see e.g., [AB09, Proof of Theorem 9.11]) that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),r,\mathcal{D}} \left[\mathcal{P}^{\mathcal{D}}_{r,d}(z_1,\ldots,z_j) = z_{j+1}\right] - \frac{1}{2} \\
&= \frac{(-1)^d}{2} \left(\mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})=1 | \hat{z}_{j+1} = z_{j+1}\right] + \mathop{\mathbb{E}}_{z\sim \mathsf{NW}^f(s),\mathcal{D},\hat{z}} \left[\mathcal{D}(\hat{z})\neq 1 | \hat{z}_{j+1} \neq z_{j+1}\right] - 1 \right) \\
&\geq \gamma/t.
\end{aligned}$$

This concludes the construction of a collection of deterministic oracle circuits $\mathcal{P}^{\mathcal{D}}_{r,d}$ satisfying Eq. (10).

We now construct an algorithm $\mathcal{A}^{\mathcal{D}}_{j,r,r',d}(x)$ that has hardwired on its code the $(t-j)$-bit (random) string r from above and another $(m-n)$-bit (random) string r' introduced next. We will show that, on average over the choices of r and r', the algorithm computes $f(x)$ on a random x with noticeable probability. For each $i \neq j+1$, we now assume (and subsequently prove) the existence of a deterministic circuit $C_{i,r'}(x)$ of size $O(t)$ that computes $\mathsf{NW}^f(s)_i$, where $s \in \{0,1\}^m$ is defined as $s|_{S_{j+1}} = x$ and $s|_{\overline{S_{j+1}}} = r'$.

The circuit $\mathcal{A}^{\mathcal{D}}_{j,r,r',d}$ on an input x invokes $\mathcal{P}^{\mathcal{D}}_{r,d}$ as follows. $\mathcal{A}^{\mathcal{D}}_{j,r,r',d}(x)$ fixes seed s consisting of $s|_{S_{j+1}} = x$ and $s|_{\overline{S_{j+1}}} = r'$, computes $\mathsf{NW}^f(s)_1, \ldots, \mathsf{NW}^f(s)_j$ using the corresponding circuits $C_{i,r'}(x)$, and outputs $\mathcal{P}^{\mathcal{D}}_{r,d}\big(\mathsf{NW}^f(s)_1, \ldots, \mathsf{NW}^f(s)_j\big)$.

Notice that, averaging over the random choices of r and r', and writing s as the random string obtained from random choices of x and r', we have:

$$\begin{aligned}\mathop{\mathbb{E}}_{r,r'}\left[\Pr_{x,\mathcal{D}}[\mathcal{A}^{\mathcal{D}}_{j,r,r',d}(x) = f(x)]\right] &= \mathop{\mathbb{E}}_{r,r'}\left[\Pr_{x\in\{0,1\}^n,\mathcal{D}}\left[\mathcal{P}^{\mathcal{D}}_{r,d}\big(\mathsf{NW}^f(s)_1, \ldots, \mathsf{NW}^f(s)_j\big) = f(x)\right]\right] \\ &= \Pr_{z\sim\mathsf{NW}^f(s),r,\mathcal{D}}\left[\mathcal{P}^{\mathcal{D}}_{r,d}(z_1, \ldots, z_j) = z_{j+1}\right] \geq \frac{1}{2} + \gamma/t. \qquad (11)\end{aligned}$$

Consequently, we get that

$$\begin{aligned}\Pr_{r,r'}\left[\Pr_{x,\mathcal{D}}[\mathcal{A}^{\mathcal{D}}_{j,r,r',d}(x) \neq f(x)] \geq \frac{1}{2} - \gamma/2t\right] &\leq \frac{\mathbb{E}_{r,r'}\left[\Pr_{x,\mathcal{D}}[\mathcal{A}^{\mathcal{D}}_{j,r,r',d}(x) \neq f(x)]\right]}{\frac{1}{2} - \gamma/2t} \\ &\leq \frac{\frac{1}{2} - \gamma/t}{\frac{1}{2} - \gamma/2t} = \frac{1 - 2\gamma/t}{1 - \gamma/t} \leq 1 - \gamma/t, \qquad (12)\end{aligned}$$

where the first inequality comes from Markov's inequality, the second inequality comes from Equation (11), and the last inequality comes from the fact that $1 - 2x \leq (1-x)^2$ for every $x \in \mathbb{R}$.

Using the assumption about the size of $C_{i,r'}$, we have that the size of $\mathcal{A}^{\mathcal{O}}$ is trivially $O(t^2)$. We prove next the existence of circuits $C_{i,r}(x)$ of the claimed size. First notice that $\mathsf{NW}^f(s)_i$ depends on at most $\log t$ bits of $s|_{S_{j+1}} = x$, by Lemma 4.1. Therefore, for fixed i and r', we can define $C_{i,r'}(x)$ as a lookup-table of size $O(t)$ that stores the corresponding values of f for all possible choices of the $\leq \log t$ relevant bits of x.

Finally, we define the uniform algorithm $\mathcal{B}^f(1^t)$ as follows. First, it picks $j^* \in [t-1]$, $d^* \in \{0,1\}$, $r \in \{0,1\}^{t-j}$ and $r' \in \{0,1\}^{m-n}$ uniformly at random. Then $\mathcal{B}^f$ outputs the oracle circuit $\mathcal{A}^{\mathcal{O}}_{j^*,r,r',d^*}$, where the lookup table of each relevant circuit $C_{i,r'}$ can be computed by $\mathcal{B}$ using membership queries to its oracle f. Note that the computation of $\mathcal{B}^f$ and the description of $\mathcal{A}^{\mathcal{O}}$ are indeed oblivious to the particular choice of $\mathcal{D}$.

Since for any fixed $\mathcal{D}$ as in the hypothesis of the result good choices of j^* and d^* are made with probability at least $\frac{1}{2t}$, and in this case r and r' yield a circuit $\mathcal{A}^{\mathcal{D}}_{j^*,r,r',d^*}$ with the desired properties with probability at least γ/t by Equation (12), we have that for any admissible $\mathcal{D}$, with probability at least $\Omega(\gamma/t^2)$ over its internal randomness $\mathcal{B}^f$ outputs a deterministic oracle circuit $\mathcal{A}^{\mathcal{O}}_{j^*,r,r',d^*}$ that satisfies Equation (9). (Note that the analysis shows that the construction succeeds with the desired probability for any fixed choice of the inherently probabilistic circuit $\mathcal{D}$, though different random choices of the parameters d, r, and r' might be needed as a function of $\mathcal{D}$.) □

We now establish a quantum analogue of this result, stated in a way that is convenient for our application.

Lemma 4.4 (Uniform Nisan-Wigderson reconstruction for quantum circuits)**.** *Let $s_F, s_D, t\colon \mathbb{N} \to \mathbb{N}$ and $\gamma\colon \mathbb{N} \to [0,1]$ be constructive functions, where $n \leq t(n) \leq 2^n$ for every n. There exists a sequence $\{\mathcal{C}^{\mathsf{NW}}_n\}_{n\geq 1}$ of quantum circuits $\mathcal{C}^{\mathsf{NW}}_n$ such that:*

(*i*) Input: *Each circuit $\mathcal{C}^{\mathsf{NW}}_n$ gets as input strings 1^n, $1^{t(n)}$, $\mathsf{code}(D)$, and $\mathsf{code}(F)$, where $\mathsf{code}(D)$ encodes a quantum circuit D of size $\leq s_D(n)$ over $t(n)$ input bits, and $\mathsf{code}(F)$ encodes a quantum circuit F of size $\leq s_F(n)$ over n input bits.*

(*ii*) Uniformity and Size: *Each circuit $\mathcal{C}^{\mathsf{NW}}_n$ is of size $S(n) = \mathsf{poly}(t(n), s_F(n), s_D(n))$, and there is a uniform deterministic algorithm that given 1^n runs in time $\mathsf{poly}(S(n))$ and prints the string $\mathsf{code}(\mathcal{C}^{\mathsf{NW}}_n)$.*

(*iii*) Output and Correctness: *Suppose that the input circuit F computes a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, and consider the associated function $\mathsf{NW}^f\colon \{0,1\}^{m(n)} \to \{0,1\}^{t(n)}$, where $m(n) = cn^2$. In addition, assume that the input circuit D satisfies*

$$\left| \Pr_{s\in\{0,1\}^{m(n)},\, D} \left[D(\mathsf{NW}^f(s)) = 1\right] - \Pr_{y\in\{0,1\}^{t(n)},\, D} \left[D(y) = 1\right] \right| \;\geq\; \gamma(n). \tag{13}$$

Then with probability $\Omega(\gamma(n)/t(n)^2)$ over its output measurement, $\mathcal{C}^{\mathsf{NW}}_n$ produces a string $\mathsf{code}(A)$ encoding a quantum circuit A of size $O(t(n)^2 \cdot s_D(n))$ such that

$$\Pr_{x\in\{0,1\}^n,\, A} \left[A(x) = f(x)\right] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x\sim\{0,1\}^n,\, A} \left[\left\| \Pi_{f(x)} A \left|x\right\rangle \left|0^\ell\right\rangle \right\|\right] \;\geq\; \frac{1}{2} + \frac{\gamma(n)}{2t(n)}. \tag{14}$$

Note. We stress that the result is non-trivial because the size of A is independent of $s_F(n)$ (otherwise it would be sufficient for $\mathcal{C}^{\mathsf{NW}}_n$ to output the string $\mathsf{code}(F)$, which encodes a quantum circuit that computes $f(x)$ on every input x with probability $\geq 2/3$).

Proof of Lemma 4.4. We follow the argument in the proof of Lemma 4.3. Here the circuit $\mathcal{C}^{\mathsf{NW}}_n$ takes the role of $\mathcal{B}^f$, the oracle f is replaced by the input circuit F, and the input circuit D behaves as the inherently probabilistic circuit $\mathcal{D}$.

Under the assumption that the quantum circuit D distinguishes the output of $\mathsf{NW}^f(s)$ on a random seed s from a random string y (Equation 13), and proceeding as in the proof of Lemma 4.4 while replacing $\mathcal{D}$ with D, it follows from Lemma 2.17 that the same probability analysis holds. Consequently, there is a quantum circuit A that employs D as a sub-routine and computes f with advantage $1/2+\gamma/2t$. The only relevant difference here is that in order to generate with the desired probability a correct description of A, it is necessary to evaluate the function f on different input strings (recall that we hardwire the corresponding answers in the circuits $C_{i,r'}$, which appear as sub-circuits of A). To achieve that, $\mathcal{C}^{\mathsf{NW}}_n$ simulates the input quantum circuit $F(x)$ (using a universal quantum circuit), amplifying its success probability of computing $f(x)$ via standard techniques in a way that reduces the probability that F errs on *any* string employed during these simulations to less than $1/2$. Whenever the correct values of f needed in the circuits $C_{i,r'}$ are produced, we get that with probability $\Omega(\gamma(n)/t(n)^2)$ over its output measurement, $\mathcal{C}^{\mathsf{NW}}_n$ successfully generates a "good" quantum circuit A from circuits F and D and its internal random choices, i.e., Equation 14 holds for A.

While in the proof of Lemma 4.3 the size of each deterministic oracle circuit $\mathcal{A}^{\mathcal{O}}$ is $O(t^2)$, here the number of gates in the corresponding quantum circuit A is $O(t(n)^2 \cdot s_D(n))$, since A explicitly incorporates the computation of circuit D, which by assumption has size at most $s_D(n)$.

We now discuss the uniformity and size of each quantum circuit $\mathcal{C}^{\mathsf{NW}}_n$. Recall that $\mathcal{C}^{\mathsf{NW}}_n$ operates in a way that is analogous to algorithm B^f from Lemma 4.3. A bit more formally, $\mathcal{C}^{\mathsf{NW}}_n$ must output a string $\mathsf{code}(A)$ from strings $\mathsf{code}(F)$ and $\mathsf{code}(D)$ (and the auxiliary parameters 1^n and $1^{t(n)}$). The computation of $\mathcal{C}^{\mathsf{NW}}_n$ involves manipulating the codes of F and D to produce the code of A, and includes the simulation of the quantum circuit F on at most $t(n) \cdot t(n)$ distinct input strings (we have at most $t(n)$ circuits $C_{i,r'}$, and each of them stores the value of f on at most t inputs), with an overhead for the amplification of the success probability. Since the descriptions of F and D have length $\mathsf{poly}(s_F(n))$ and $\mathsf{poly}(s_D(n))$, respectively, and each quantum simulation can be done using $\mathsf{poly}(s_F(n))$ gates, it follows that each quantum circuit $\mathcal{C}^{\mathsf{NW}}_n$ can be implemented with $S(n) = \mathsf{poly}(t(n), s_F(n), s_D(n))$ gates. Finally, it is not hard to see that the code of $\mathcal{C}^{\mathsf{NW}}_n$ is explicit and can be deterministically generated from 1^n in time $\mathsf{poly}(S(n))$, since the description of B^f in the proof of Lemma 4.3 is also explicit. □

4.2 Goldreich-Levin lemma in the quantum setting

Lemma 4.5. *Let $f\colon \{0,1\}^{kn} \to \{0,1\}^k$. Suppose there is a quantum circuit $\mathcal{A}$ (that uses $m \geq 1$ workspace qubits) satisfying*

$$\mathop{\mathbb{E}}_{x\in\{0,1\}^{kn}} \mathop{\mathbb{E}}_{r\in\{0,1\}^k} \left[\|\Pi_{f(x)\cdot r}\mathcal{A}\,|x,r,0^m\rangle\|^2\right] \geq \frac{1}{2} + \gamma.$$

Then there is a quantum oracle circuit $\mathcal{B}^{\mathcal{A}}$ of size $O(kn)$ that has oracle access to $\mathcal{A}$ (and to its inverse $\mathcal{A}^\dagger$) such that

$$\mathop{\mathbb{E}}_{x,\mathcal{B}^{\mathcal{A}}} \left[\|\Pi_{f(x)}\mathcal{B}^{\mathcal{A}}\,|x,0^{k+m+1}\rangle\|^2\right] \geq \frac{\gamma^3}{2}.$$

Moreover, a quantum circuit $\mathcal{B}$ of this form can be constructed from a quantum circuit $\mathcal{A}$ as above by a uniform sequence of circuits, which we formalise as follows. Let $k, s_{\mathcal{A}}\colon \mathbb{N} \to \mathbb{N}$ and $\gamma\colon \mathbb{N} \to [0,1/2]$ be constructive functions. In addition, let $f_n\colon \{0,1\}^{kn} \to \{0,1\}^k$ be a sequence of functions, where $k = k(n)$. Then there is a sequence $\{\mathcal{C}^{\mathsf{GL}}_n\}_{n\geq 1}$ of deterministic circuits $\mathcal{C}^{\mathsf{GL}}_n$ of size $\mathsf{poly}(n, k(n), s_{\mathcal{A}}(n))$ for which the following holds. If $\mathcal{C}^{\mathsf{GL}}_n$ is given as input strings 1^n and a description $\mathsf{code}(\mathcal{A})$ of a quantum circuit $\mathcal{A}$ of size $\leq s_{\mathcal{A}}(n)$ with the property above, then it outputs a string $\mathsf{code}(\mathcal{B})$ describing a quantum circuit $\mathcal{B}$ of size $O(n \cdot k(n) \cdot s_{\mathcal{A}}(n))$ such that

$$\mathop{\mathbb{E}}_{x,\mathcal{B}} \left[\|\Pi_{f_n(x)}\mathcal{B}\,|x,0^{k+m+1}\rangle\|^2\right] \geq \frac{\gamma(n)^3}{2}.$$

Proof. Without loss of generality, let us assume that $\mathcal{A}$ measures the first qubit of $\mathcal{A}\,|x,r,0^m\rangle$ and produces it as an output. We define the algorithm $\mathcal{B}^{\mathcal{A}}$ on input x as follows:

1. Start with $\frac{1}{\sqrt{2^k}} \sum_r |x,r,0^m,1\rangle$
2. Apply $\mathcal{A}$ on the first $kn + k + m$ qubits.
3. Apply a control-Z operation with the "output of $\mathcal{A}$" (i.e., first qubit) as the control qubit and the last qubit as the target qubit.
4. Apply $\mathcal{A}^\dagger$ on the first $kn + k + m$ qubits.
5. Apply Hadamard transform on the second register.
6. Measure all qubits in the computational basis.

7. If the outcome is of the form $|x, a, 0^{k+m}, 1\rangle$, output a.

To analyze its correctness, we first define $G = \{x : \mathbb{E}_{r\in\{0,1\}^k}\left[\|\Pi_{f(x)\cdot r}\mathcal{A}\,|x, r, 0^m\rangle\|^2\right] \geq \frac{1}{2} + \frac{\gamma}{2}\}$. We will show that

$$|G| \geq \frac{\gamma}{2} \cdot 2^n, \tag{15}$$

and that for every $x \in G$, we have

$$\mathop{\mathbb{E}}_{\mathcal{B}^{\mathcal{A}}}\left[\|\Pi_{f(x)}\mathcal{B}^{\mathcal{A}}\,|x, 0^{k+m+1}\rangle\|^2\right] \geq \gamma^2. \tag{16}$$

Before proving Equations (15) and (16), notice that they directly imply our statement, since

$$\begin{aligned}\mathop{\mathbb{E}}_{x,\mathcal{B}^{\mathcal{A}}}\left[\|\Pi_{f(x)}\mathcal{B}^{\mathcal{A}}\,|x, 0^{k+m+1}\rangle\|^2\right] &\geq \frac{|G|}{2^n}\mathop{\mathbb{E}}_{x\in G,\mathcal{B}^{\mathcal{A}}}\left[\|\Pi_{f(x)}\mathcal{B}^{\mathcal{A}}\,|x, 0^{k+m+1}\rangle\|^2\right]\\ &\geq \frac{\gamma}{2}\mathop{\mathbb{E}}_{x\in G,\mathcal{B}^{\mathcal{A}}}\left[\|\Pi_{f(x)}\mathcal{B}^{\mathcal{A}}\,|x, 0^{k+m+1}\rangle\|^2\right] \geq \frac{\gamma^3}{2},\end{aligned}$$

where in the first inequality we removed some non-negative values, in the second inequality we use Equation (15) and in the third inequality we use Equation (16).

Let us now show Equation (15). Suppose toward a contradiction that $|G| < \frac{\gamma}{2} \cdot 2^n$. Then we have

$$\begin{aligned}\frac{1}{2} + \gamma &\leq \mathop{\mathbb{E}}_{x\in\{0,1\}^{kn}}\mathop{\mathbb{E}}_{r\in\{0,1\}^k}\left[\|\Pi_{f(x)\cdot r}\mathcal{A}\,|x, r, 0^m\rangle\|^2\right]\\ &= \frac{|G|}{2^n}\mathop{\mathbb{E}}_{x\in G}\mathop{\mathbb{E}}_{r\in\{0,1\}^k}\left[\|\Pi_{f(x)\cdot r}\mathcal{A}\,|x, r, 0^m\rangle\|^2\right] + \left(1 - \frac{|G|}{2^n}\right)\mathop{\mathbb{E}}_{x\notin G}\mathop{\mathbb{E}}_{r\in\{0,1\}^k}\left[\|\Pi_{f(x)\cdot r}\mathcal{A}\,|x, r, 0^m\rangle\|^2\right]\\ &< \frac{|G|}{2^n} + \left(1 - \frac{|G|}{2^n}\right)\left(\frac{1}{2} + \frac{\gamma}{2}\right)\\ &< \frac{1}{2} + \gamma/2,\end{aligned}$$

which is a contradiction. Therefore, Equation (15) must be true.

Now we prove Equation (16) and for that let us fix an arbitrary $x \in G$. This part of the proof closely follows the proof by Cleve and Adcock [AC02]. Without loss of generality, we can assume that $\mathcal{A}$ on the input state acts as

$$\mathcal{A}\,|x, r, 0^m\rangle = |x, r\rangle \otimes \left(\alpha_{x,r,0}\,|\psi_{x,r,0}\rangle\,|f(x)\cdot r\rangle + \alpha_{x,r,1}\,|\psi_{x,r,1}\rangle\,|1 \oplus f(x)\cdot r\rangle\right).$$

From the assumption that $x \in G$, we have that

$$\mathop{\mathbb{E}}_{r}\left[|\alpha_{x,r,0}|^2\right] \geq \frac{1}{2} + \frac{\gamma}{2} \quad \text{and} \quad \mathop{\mathbb{E}}_{r}\left[|\alpha_{x,r,1}|^2\right] \leq \frac{1}{2} - \frac{\gamma}{2}.$$

We define two quantum states. First, we start with $\frac{1}{\sqrt{2^k}}\sum_r |x, r, 0^m, 1\rangle$ and apply steps 2 and 3 of $\mathcal{B}^{\mathcal{A}}$ to obtain

$$|\phi_x\rangle = |x\rangle \otimes \frac{1}{\sqrt{2^k}}\sum_r (-1)^{f(x)\cdot r}\left(\alpha_{x,r,0}\,|r\rangle\,|\psi_{x,r,0}\rangle\,|f(x)\cdot r, 1\rangle - \alpha_{x,r,1}\,|r\rangle\,|\psi_{x,r,1}\rangle\,|1 \oplus f(x)\cdot r, 1\rangle\right).$$

In order to analyze the probability of the measurement outcome a (the output of $\mathcal{B}^{\mathcal{A}}$ in step (7)) equalling $f(x)$, consider the state where we start with $|x, f(x), 0^{k+m}, 1\rangle$, apply the Hadamard transform on the second register and then apply $\mathcal{A}$ on the first $kn + k + m$ qubits to obtain

$$|\sigma_x\rangle = \frac{1}{\sqrt{2^k}} |x\rangle \sum_r (-1)^{f(x)\cdot r} \big(\alpha_{x,r,0} |r\rangle |\psi_{x,r,0}\rangle |f(x)\cdot r, 1\rangle + \alpha_{x,r,1} |r\rangle |\psi_{x,r,1}\rangle |1 \oplus f(x)\cdot r, 1\rangle\big). \quad (17)$$

The probability that $\mathcal{B}^{\mathcal{A}}(x)$ outputs $f(x)$ is then given by

$$|\langle \sigma_x \mid \phi_x \rangle|^2 = \left| \mathop{\mathbb{E}}_r \left[|\alpha_{x,r,0}|^2 - |\alpha_{x,r,1}|^2 \right] \right|^2 \geq \gamma^2, \quad (18)$$

where we use the fact that for $x \in G$, we have $\mathbb{E}_r \left[|\alpha_{x,r,0}|^2 - |\alpha_{x,r,1}|^2 \right] \geq \gamma$. This concludes the proof of Equation (16). Notice that the size of $\mathcal{B}^{\mathcal{A}}$ can be simply checked by inspection from its description.

We discuss now the moreover part of our lemma. The circuit $\mathcal{C}_n^{\mathsf{GL}}$ receives the input $\mathsf{code}(\mathcal{A})$ (and the auxiliary parameter 1^n) and outputs the circuit $\mathcal{B}$ that executes the steps (1) – (7) described above, with an oracle call to $\mathcal{A}$ replaced by the execution of $\mathsf{code}(\mathcal{A})$ (or its inverse). It is straightforward from the previous calculations that $\mathcal{B}$ has the desired properties and that it has size at most $O(k(n) \cdot n \cdot s_{\mathcal{A}}(n))$. Notice that the circuit $\mathcal{C}_n^{\mathsf{GL}}$ that prints $\mathcal{B}$ is deterministic and it can be implemented using $\mathsf{poly}(n, k(n), s_A(n))$ gates. We also notice that code of $\mathcal{C}_n^{\mathsf{GL}}$ is explicit and it can be deterministically generated from 1^n in time $\mathsf{poly}(n, k(n), s_{\mathcal{A}}(n))$. □

4.3 Local list decoding and uniform hardness amplification for quantum circuits

In this section, we start with some arbitrary function $g : \{0,1\}^n \to \{0,1\}$, which we assume to be mildly hard, and our goal is to amplify its hardness using techniques from local list decoding of (classical) error-correcting codes. Specifically, we prove a quantum analogue of the direct product theorem of Impagliazzo, Jaiswal, Kabanets, and Wigderson [IJKW10]. For simplicity of notation, we will fix $n \in \mathbb{N}$ and denote $\mathfrak{U} = \{0,1\}^n$. Roughly speaking, we will prove that computation of the concatenation of k independent copies of g amplifies its hardness exponentially as a function of k. For that, we first define the domain of such concatenation.

Definition 4.6 (k-sets)**.** *Let $\mathcal{S}_{n,k} = \{S \subseteq \mathfrak{U} : |S| = k\}$ be the set of all subsets of $\{0,1\}^n$ of size k. When n is fixed, we denote $\mathcal{S}_k = \mathcal{S}_{n,k}$.*

Remark 4.7. We consider the n-bit strings in a set $B \in \mathcal{S}_{n,k}$ in *lexicographic order* when B is given as input to a Boolean circuit.

We consider then the *k-direct product* of g, denoted $g^k : \mathcal{S}_k \to \{0,1\}^k$, where $g^k(B)$ is the concatenation of $g(x)$ for every $x \in B$ in a canonical order, i.e,

$$g^k(x_1, \ldots, x_k) = \big(g(x_1), \ldots, g(x_k)\big).$$

Whenever it is clear from the context, we abuse the notation with $g(B) = g^k(B)$, for $B \in \mathcal{S}_k$. There are well known connections between hardness amplification and classical error-correcting (for example see [AB09, Chapter 19]) and part of our terminology reflects them. For instance, the code can be viewed as corresponding to g^k and decoding will correspond to computing $g(y)$ for some input $y \in \mathfrak{U}$ when given appropriate access to g^k.

As a preliminary step, we consider hardness amplification for inherently probabilistic computations, which we introduced in Section 2.7 as an intermediate model between classical and quantum circuits.

4.3.1 Inherently probabilistic circuits

Recall that an *inherently probabilistic circuit* $\mathcal{G}$ for computing a function $g\colon \{0,1\}^m \to \{0,1\}^\ell$ with probability at least ε is a circuit that assigns to each input $z \in \{0,1\}^m$ a distribution $\mathcal{G}(z)$ supported over $\{0,1\}^\ell$ such that

$$\Pr_{\substack{z\sim\{0,1\}^m,\\ v\sim\mathcal{G}(z)}}[v = g(z)] \geq \varepsilon\,.$$

Our goal in this section is to show that an inherently probabilistic circuit $\mathcal{G}$ that computes g^k with small probability $\varepsilon > 0$ can be used as a subroutine in a randomized circuit with oracle access to $\mathcal{G}$ to compute g with high probability $1-\delta$. Moreover, we aim to design a uniform randomized algorithm that is able to generate, with non-trivial probability ζ as a function of ε, a description of this oracle circuit from a description of $\mathcal{G}$.

We state the main theorem of this section. In the following, it might be instructive to think of the following setting of parameters as a function of the input size: $\varepsilon = 2^{-\sqrt{n}}$, $\delta = 1/\operatorname{poly}(n)$, and $k = \operatorname{poly}(n)$.

Theorem 4.8 (Local list decoding for inherently probabilistic circuits)**.** *There exists a universal constant $C \geq 1$ for which the following holds. Let $n \geq 1$ be a positive integer, k be an even integer, and let $\varepsilon, \delta > 0$ satisfy*

$$k \geq C \cdot \frac{1}{\delta} \cdot \left[\log\left(\frac{1}{\delta}\right) + \log\left(\frac{1}{\varepsilon}\right)\right]. \tag{19}$$

If $\mathcal{G}$ is an inherently probabilistic circuit defined over $\mathcal{S}_{n,k}$ with k output bits such that

$$\mathop{\mathbb{E}}_{B\sim\mathcal{S}_{n,k},\,\mathcal{G}}\left[\mathcal{G}(B) = g^k(B)\right] \geq \varepsilon, \tag{20}$$

then there is a randomized oracle circuit $\mathcal{B}^{\mathcal{O}}$ of size $\mathsf{poly}(n, k, \log(1/\delta), 1/\varepsilon)$ such that

$$\mathop{\mathbb{E}}_{\substack{x\sim\mathfrak{U},\\ y\sim\{0,1\}^*,\,\mathcal{G}}}\left[\mathcal{B}^{\mathcal{G}}(x,y) = g(x)\right] \geq 1-\delta. \tag{21}$$

Moreover, there is a uniform randomized algorithm $\mathcal{D}$ that, given n, k, ε, and δ satisfying the conditions of the theorem, access to a description of $\mathcal{G}$, and the ability to run $\mathcal{G}$ on a given input $B \in \mathcal{S}_{n,k}$, computes in time $\mathsf{poly}(n, k, \log(1/\delta), 1/\varepsilon)$ and outputs with probability $\zeta = \Omega(\varepsilon^2)$ over its internal randomness and the randomness of $\mathcal{G}$ a description of a circuit $\mathcal{B}^{\mathcal{O}}$ with this property.

Remark 4.9 (Amplification)**.** We observe that the generating probability ζ can be amplified using standard techniques (repetition and hypothesis testing) if the uniform randomized algorithm $\mathcal{D}$ is also given oracle access to the function g.

Remark 4.10 (On the correlation between $\mathcal{G}$ and g^k)**.** Before we proceed with the proof of Theorem 4.8, it is worth pointing out different ways in which the inherently probabilistic circuit $\mathcal{G}$ can be correlated with g^k. First, $\mathcal{G}$ might behave as a *deterministic* circuit, being correct on an arbitrary set $V \subseteq \mathcal{S}_k$ of relative size ε (i.e., on each $B \in V$ we have $\mathcal{G}(B) = g^k(B)$ with probability 1), and being incorrect elsewhere. At the other extreme, it is also possible for $\mathcal{G}$ to agree with $g^k(B)$ on each $B \in \mathcal{S}_k$ with probability about ε over its internal randomness, spreading out its correlation with g^k across all inputs. Since $\mathcal{G}$ is inherently probabilistic, there is no simple way of reducing this case to the preceding case (e.g., by fixing $\mathcal{G}$'s internal randomness). Finally, it is possible that $\mathcal{G}$'s behavior combines in an arbitrary way the two aforementioned cases, while maintaining an overall advantage ε with respect to g^k. A proof of Theorem 4.8 needs to address all possible scenarios.

To present the local decoding algorithm with which we will prove Theorem 4.8, it will be convenient to refer to the bipartite graph (the incidence graph of the Johnson scheme) that contains all $k/2$-sets on the left and all k-sets on the right, where the edges correspond to incidence of the sets.

Definition 4.11 (Edges and Neighbors)**.** *We define the set of* edges $\mathcal{I} = \{(A, B) \in \mathcal{S}_{k/2} \times \mathcal{S}_k : A \subseteq B\}$. *We define the* neighbors *of a set A by $N_{\mathcal{I}}(A) = \{B \in \mathcal{S}_k : (A, B) \in \mathcal{I}\}$ and the neighbors of $A \in \mathcal{S}_{k/2}$ and $x \in \mathfrak{U} \setminus A$ as $N_{\mathcal{I}}(A, x) = \{B \in \mathcal{S}_k : A \cup \{x\} \subseteq B\}$.*

We proceed to present the key sub-procedure for the list-decoding algorithm, which is a circuit that, given a $k/2$-set A and an assignment w to A, attempts to compute $g(x)$ for a given x by choosing a random neighbour B' of A and x, running $\mathcal{G}$ on B' and outputting the corresponding value for x if the output of $\mathcal{G}(B')$ is consistent with w on A.

Construction 4.12 (The decoding circuit $C_{A,w}$)**.** *For a fixed $A \in \mathcal{S}_{k/2}$, $w \in \{0,1\}^{k/2}$, and a parameter T, we define the randomized circuit $C_{A,w}$ with oracle access to $\mathcal{G}$ that, on input $x \in \mathfrak{U}$, works as follows:*

1. *If $x \in A$, output $w|_x$.*[12]
2. *Repeat for T steps:*
   - 2.1. *Pick a uniformly random $B' \in N_{\mathcal{I}}(A, x)$.*
   - 2.2. *Sample $v' \sim \mathcal{G}(B')$.*
   - 2.3. *If $v'|_A = w$, output $v'|_x$.*
3. *Output $\perp$.*

Using Construction 4.12, the list-decoding algorithm will decode by evaluating the input circuit $\mathcal{G}$ on a random k-set B and outputting the circuit $C_{A,w}$ with respect to a random $k/2$-set $A \subset B$.

Construction 4.13 (The list-decoding algorithm $\mathcal{D}$)**.** *The uniform randomized algorithm $\mathcal{D}$ is given n, k, ε, and δ satisfying the conditions of Theorem 4.8, access to a description of $\mathcal{G}$, and a parameter T. The algorithm $\mathcal{D}$ operates as follows:*

1. *Pick a random k-set $B \in \mathcal{S}_{n,k}$ and a random $k/2$-subset $A \subseteq B$.*
2. *Sample $\mathcal{G}(B)$ and use it to set $w = \mathcal{G}(B)|_A$.*
3. *Output the description of the circuit $C_{A,w}$ (defined in Construction 4.12) with respect to the given parameter T.*

Similar to [IJKW10], we prove Theorem 4.8 in two steps. First, we show that if A and w, which were randomly chosen by the list-decoder, have certain desired properties and the repetition parameter T is sufficiently large, the circuit $C_{A,w}$ computes g with high probability over a random input $x \in \mathfrak{U}$, its internal randomness, and the inherent randomness of $\mathcal{G}$. (Therefore, the oracle circuit $\mathcal{B}^{\mathcal{O}}$ in the statement of the lemma will be $C_{A,w}$ for a good choice of A and w, with its oracle $\mathcal{O}$ computing as $\mathcal{G}$, and the input string y used as a source of random bits.) Then, we argue that with probability at least $\zeta = \Omega(\varepsilon^2)$ the uniform randomized algorithm $\mathcal{D}$ is able to generate good parameters A and w.

Throughout this section, we define $\widetilde{C}_A = C_{A,g^{k/2}(A)}$, for simplicity.

[12] By $w|_x$, we mean the following: suppose $g(y_1, \ldots, y_{k/2}) = (w_1, \ldots, w_{k/2})$ for $y_i \in \{0,1\}^n$ and $w_i \in \{0,1\}$, and $y_\ell = x$ for some $\ell \in [k/2]$, then $w|_x = w_\ell$.

Remark 4.14 (Well-defined conditional probabilities)**.** In order to simplify our exposition, we will assume without loss of generality throughout this section that for every $B \in \mathcal{S}_k$, we have $\Pr_{\mathcal{G}}[\mathcal{G}(B) = g^k(B)] > 0$. Indeed, this can be easily obtained by defining a modified circuit $\mathcal{G}'$ from $\mathcal{G}$ that, say, with probability 2^{-n} outputs a uniformly random value in $\{0,1\}^k$, and otherwise runs $\mathcal{G}$ on B. We have this assumption so that definitions and proofs become simpler when considering certain conditional probabilities. We stress that our argument would still work without this trick, with a slightly more complicated exposition.

We will need a few definitions to prove the theorem. Observe that some of them implicitly refer to $\mathcal{G}$ and g, which are fixed for the remainder of this section. We start with the definition of correctness of the algorithm $\mathcal{G}$ on a k-set.

Definition 4.15. *For $B \in \mathcal{S}_k$, we define*

$$\mathsf{Corr}_{\mathcal{G}}(B) = \mathop{\mathbb{E}}_{\mathcal{G}}[\mathcal{G}(B) = g^k(B)].$$

We say that B is η-correct if $\mathsf{Corr}_{\mathcal{G}}(B) \geq \eta$.

Notice that, using this notation, we can rewrite the assumption of Equation (20) in Theorem 4.8 as

$$\mathop{\mathbb{E}}_{B \in \mathcal{S}_k}[\mathsf{Corr}_{\mathcal{G}}(B)] \geq \varepsilon.$$

Since $\mathcal{G}$ is fixed throughout this section, we might write $\mathsf{Corr}(B)$ instead of $\mathsf{Corr}_{\mathcal{G}}(B)$.

Next, we define good edges as (mostly) correct sets for which many of their neighbours are also (mostly) correct.

Definition 4.16 (Good edges)**.** *We say an edge $(A, B) \in \mathcal{I}$ is (γ, η)-*good *if*

(i) B is η-correct; and

(ii) A γ fraction of the neighbors of A are η-correct, i.e.,

$$\mathop{\mathbb{E}}_{B' \in N_{\mathcal{I}}(A)}[\mathsf{Corr}(B') \geq \eta] \geq \gamma.$$

We remark that with the above definition we already start deviating from [IJKW10]. Since their version of $\mathcal{G}$ is deterministic, its answers are either correct or wrong and therefore they can afford to have $\eta = 1$. In our case, we need to take into account the *randomized* aspect of $\mathcal{G}$ and therefore we need to be more flexible with the definition of goodness.

As in [IJKW10], in order to prove the correctness of $C_{A,w}$, we need edges that satisfy a stronger property than the above, referred to as "excellence". For that, we first define the function that computes the probability that an edge (A, B') leads to a wrong answer on a random $x \in B' \setminus A$ conditioned on a correct answer on A.

Definition 4.17. *For an edge $(A, B') \in \mathcal{I}$, we define*

$$\mathsf{ErrCons}(A, B') = \mathop{\mathbb{E}}_{x \in B' \setminus A}\left[p_{\mathsf{err}}(x, B')\right], \tag{22}$$

where $p_{\mathsf{err}}(x, B') = \Pr_{y \sim \mathcal{G}(B')}[y|_x \neq g(x) \mid y|_A = g^{k/2}(A)]$.

Using the foregoing definition, we define an excellent edge (A, B) as a good edge for which the expected fraction of errors in the neighbors of A is small, where this expectation gives weight to the edges based on their probability of being correct on A.

Definition 4.18 (Excellent edges)**.** *We call an edge* (A, B) (η, γ, α)-excellent *if it is* (η, γ)*-good and*

$$\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_{\mathcal{I}}(A)} \left[\mathsf{ErrCons}(A, B') \right] \leq \alpha, \tag{23}$$

where the distribution $\mathcal{W}_{\mathcal{I}}(A)$*, supported over* $N_{\mathcal{I}}(A)$*, is defined as follows: for each* $B' \in N_{\mathcal{I}}(A)$*, let* $p_{\mathsf{cons}}(B') = \Pr_{\mathcal{G}} \left[\mathcal{G}(B')|_A = g^{k/2}(A) \right]$*. Moreover, let* $p_{\mathsf{tot}}(A) = \sum_{B' \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B')$*. Then each* B' *is assigned probability* $p_{\mathsf{cons}}(B')/p_{\mathsf{tot}}(A)$*.*

For the reader familiar with [IJKW10], we mention that the definition above is really crucial. It refers to a more general class of distributions $\mathcal{W}_{\mathcal{I}}(A)$ when contrasted with the analogous definition from their paper. (This distribution naturally appears in our error analysis when we condition on $C_{A,w}$ not outputting the error symbol $\perp$.) Jumping ahead, it ties the two main lemmas stated below and proved in the subsequent sections, and introduces significant difficulties when establishing each of them.

Given these definitions, we can now state the main two lemmas needed for the proof of Theorem 4.8. The first lemma shows that if the decoding algorithm hits an excellent edge, then it will decode correctly with high probability. The second lemma shows that there are many excellent edges, and so the decoding algorithm will hit one with non-trivial probability (this is close to 0).

Lemma 4.19 (Excellence implies correctness)**.** *Fix some* $0 \leq \beta \leq 1$ *and let* $\lambda = 2e^{-\beta k/24}$*. Moreover, assume that*

$$\gamma, \eta < 1/10 \quad \text{and} \quad 4e^{-k\alpha/12} \leq (\gamma \cdot \eta)^5.$$

If (A, B) *is a* (γ, η, α)*-excellent edge, then*

$$\mathop{\mathbb{E}}_{\substack{x \sim \mathfrak{U} \setminus A, \\ \widetilde{C}_A, \mathcal{G}}} [\widetilde{C}_A(x) = g(x)] \geq 1 - \beta - (1 - \eta(\gamma - \lambda)/2)^T - 16\alpha,$$

where T *is the number of iterations in* $\widetilde{C}_A$*.*

Lemma 4.20 (Excellent edges are abundant)**.** *For any* $\alpha < \frac{1}{2}$*, if* $\mathbb{E}_{B \in \mathcal{S}_k} [\mathsf{Corr}(B)] \geq \varepsilon$ *then at least an* $(\varepsilon/3 - \frac{62208}{\alpha^3 \cdot \varepsilon^5} \cdot e^{-\frac{\alpha}{96} \cdot k})$*-fraction of the edges* $(A, B) \in \mathcal{I}$ *are* $(\varepsilon/3, \varepsilon/3, \alpha)$*-excellent.*

We prove Lemmas 4.19 and 4.20 in Sections 4.3.2 and 4.3.3, respectively. Assuming these results, we are ready to prove Theorem 4.8.

Proof of Theorem 4.8. We begin with the first part of the result; namely, showing that the algorithm $C_{A,w}$ defined in Construction 4.12, parameterized as we specify below, is of size $\mathsf{poly}(n, k, \log(1/\delta), 1/\varepsilon)$ and satisfies

$$\mathop{\mathbb{E}}_{\substack{x \sim \mathfrak{U}, \\ y \sim \{0,1\}^*, \mathcal{G}}} \left[C^{\mathcal{G}}_{A,w}(x, y) = g(x) \right] \geq 1 - \delta.$$

First, note that we can assume that $\varepsilon < 1/10$ without loss of generality. For this it is sufficient to redefine $\mathcal{G}$ to output a random value with probability $1 - 10^{-3}$, and to compute as before otherwise. Its overall success probability drops at most by a constant factor, and now $\varepsilon < 1/100$. We also assume that $\delta \leq 1/2$, by taking a smaller δ if necessary.

Let C be a large positive constant independent of the remaining parameters. Let

$$\gamma, \eta \stackrel{\text{def}}{=} \varepsilon/3 < 1/10 \quad \text{and} \quad \beta \stackrel{\text{def}}{=} \delta/3 \quad \text{and} \quad \alpha \stackrel{\text{def}}{=} \delta/48 \quad \text{and} \quad T = C \cdot \log(1/\delta) \cdot (1/\varepsilon^2),$$

where T is the repetition parameter in Construction 4.13. Note that

$$k \;\geq\; C \cdot \frac{1}{\delta} \cdot \left[\log\left(\frac{1}{\delta}\right) + \log\left(\frac{1}{\varepsilon}\right)\right],$$

and so $4e^{-k\alpha/12} \;\leq\; (\gamma \cdot \eta)^5$, for a sufficiently large choice of C.

By applying Lemma 4.20 with respect to these α, ε, and k we have that at least an $(\varepsilon/3 - \frac{62208}{\alpha^3 \cdot \varepsilon^5} \cdot e^{-\frac{\alpha}{96} \cdot k})$-fraction of the edges $(A, B) \in \mathcal{I}$ are $(\varepsilon/3, \varepsilon/3, \alpha)$-excellent. Let $\lambda = 2e^{-\beta k/24} < \varepsilon/6$. Note that our choice of parameters satisfies all the conditions of Lemma 4.19, and thus by invoking it we obtain that if (A, B) is a (γ, η, α)-excellent edge, then

$$\mathop{\mathbb{E}}_{\substack{x \sim \mathfrak{U} \setminus A, \\ \widetilde{C}_A,\, \mathcal{G}}} [\widetilde{C}_A = g(x)] \geq 1 - \beta - (1 - \eta(\gamma - \lambda)/2)^T - 16\alpha,$$

where $\widetilde{C}_A = C_{A, g^{k/2}}(x)$ as previously defined.

Now, observe that

$$(1 - \eta(\gamma - \lambda)/2)^T \leq \left(1 - \frac{\varepsilon}{3}\left(\frac{\varepsilon}{3} - \frac{\varepsilon}{6}\right) \cdot \frac{1}{2}\right)^T = \left(1 - \frac{\varepsilon^2}{36}\right)^T \leq e^{-(\varepsilon^2/36) \cdot T} \leq \frac{\delta}{3},$$

where we have used that $\varepsilon < 1/100$ and that the constant C appearing in T is large. As a consequence, if the randomized algorithm $\mathcal{D}$ succeeds in sampling a pair (A, B) that is $(\varepsilon/3, \varepsilon/3, \delta/48)$-excellent and it also samples a value $w = \mathcal{G}(B)|_A$ which agrees with $g^{k/2}(A)$, we get the randomised circuit $\widetilde{C}_A$ with oracle access to $\mathcal{G}$ such that

$$\mathop{\mathbb{E}}_{\substack{x \sim \mathfrak{U} \setminus A, \\ \widetilde{C}_A,\, \mathcal{G}}} [\widetilde{C}_A(x) = g(x)] \;\geq\; 1 - \delta/3 - \delta/3 - \delta/3 = 1 - \delta.$$

It is not hard to see that the inequality above also holds when x is sampled from the (marginally) larger set $\mathfrak{U}$, since $\widetilde{C}_A$ is always correct for $x \in A$.

On the other hand, note that, due to our choice of parameters in Lemma 4.20 we have

$$\frac{62208}{\alpha^3 \cdot \varepsilon^5} \cdot e^{-\frac{\alpha}{96} \cdot k} \;\leq\; \varepsilon/6.$$

It follows from Lemma 4.20 that at least an $(\varepsilon/6)$-fraction of the pairs $(A, B) \in \mathcal{I}$ are $(\varepsilon/3, \varepsilon/3, \delta/48)$-excellent. In particular, excellent edges of this form exist.

To sum up, the first part of the lemma follows by picking $\mathcal{B}^{\mathcal{O}} = \widetilde{C}_A$ with $\mathcal{O} = \mathcal{G}$ for any choice of (A, B) as above. Note that, by our choice of T, the number of gates in the circuit $\mathcal{B}^{\mathcal{O}}$ satisfies the parameters of the lemma.

We now prove that the uniform randomized algorithm $\mathcal{D}$ outputs a circuit with the desired properties with probability at least $\Omega(\varepsilon^2)$ over its internal randomness and the randomness of $\mathcal{G}$. It follows from our discussion in the paragraph above that, in order for the circuit output by $\mathcal{D}$ to have the desired accuracy, we need two properties from the values produced by $\mathcal{D}$:

(*i*) the randomly selected edge (A, B) is $(\varepsilon/3, \varepsilon/3, \delta/48)$-excellent; and

(*ii*) the sampled value $w = \mathcal{G}(B)|_A$ coincides with $g^{k/2}(A)$.

As we have already established, an $\Omega(\varepsilon)$ fraction of edges (A, B) are $(\varepsilon/3, \varepsilon/3, \delta/48)$-excellent, so by picking a random k-set B and a random $k/2$-subset A of B, which produces a uniformly random edge (A, B), we have that (A, B) is an excellent edge with probability $\Omega(\varepsilon)$. Then, assuming that (A, B) is $(\varepsilon/3, \varepsilon/3, \delta/48)$-excellent, it follows that $\mathcal{G}(B) = g^k(B)$ with probability at least $\varepsilon/3$, which implies that the probability that $\mathcal{G}(B)|_A = g^{k/2}(A)$ is also at least $\varepsilon/3$. Therefore, the probability that $\mathcal{D}$ outputs a circuit with the desired properties is at least $\Omega(\varepsilon^2)$. □

4.3.2 Excellence implies correctness

In this section, we prove Lemma 4.19, which shows that if an excellent edge (A, B) is picked, then $\tilde{C}_A$ computes $g(x)$ with high probability, on average over the x's. Since this proof is fairly technical, we first give an overview of its structure, before diving into the details. Recall that the goal is to upper bound the quantity

$$\mathop{\mathbb{E}}_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A,\, \mathcal{G}}} [\widetilde{C}_A(x) \neq g(x)]. \tag{24}$$

It is easy to see that the event $\widetilde{C}_A(x) \neq g(x)$ occurs if either (1) the decoder outputs $\perp$ and aborts or (2) the decoder does not output $g(x)$ conditioned on not outputting $\perp$.

We first show that the probability of (1) happening is small. More precisely, we show that given a fixed excellent edge (A, B) and a sufficiently large number of iterations T, for most $x \in \mathfrak{U} \setminus A$ the probability that $\widetilde{C}_A$ outputs $\perp$ on x is negligibly small.

Then, we follow up showing that (2) happens with low probability, which turns out to be much more cumbersome than [IJKW10]. Here, we need to upper bound

$$\mu \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x \in \mathfrak{U} \setminus A} \underbrace{\mathop{\mathbb{E}}_{y \sim \widetilde{C}_A(x),\, \mathcal{G}} [y \neq g(x) \mid y \neq \perp]}_{:= h(x)}. \tag{25}$$

If we dive into the definition of $\tilde{C}_A(x)$, Equation (25) computes, for a uniformly random $x \in \mathfrak{U} \setminus A$, the probability that at some iteration $\tilde{C}_A$ picks some $B' \sim N_{\mathcal{I}}(A, x)$ and then $\mathcal{G}(B')$ outputs some value y that is incorrect on x given that it is correct on A. Alternatively (as a thought experiment), we could consider a different sampling procedure that picks $B' \sim N_{\mathcal{I}}(A)$ and then picks a uniformly random $x \in B' \setminus A$ and analyze the probability of being incorrect to x conditioned on the fact of being correct to A. This latter procedure would give an expression similar to that for determining the excellence of the edge (A, B) and could then be related to α as required. In fact, this is the approach taken by [IJKW10] where the sampling behaviour satisfied by A, B and x (in their case), allows them to switch between both scenarios in a fairly direct way. This already becomes more complicated in our case.

For one thing, in our definition of excellence (see Definition 4.18), we sample $B' \sim \mathcal{W}_{\mathcal{I}}(A)$ and not uniformly at random. However, we succeed in indirectly relating the α-excellence of (A, B) to Eq. (25) in two steps: (i) show that α upper bounds the expression

$$\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \mathop{\mathbb{E}}_{x \sim B' \setminus A} [h(x)]; \tag{26}$$

(ii) lower bound Eq. (26) in terms of μ. In order to relate these two expressions, we still need to contend with the fact that $B' \in N_{\mathcal{I}}(A)$ is sampled with probability proportional to $p_{\mathsf{cons}}(B)$ in Eq. (26). To this end, we first partition the set $N_{\mathcal{I}}(A)$ as follows: let Γ_i be the set of all $B' \in N_{\mathcal{I}}(A)$ such that $p_{\mathsf{cons}}(B')$ lies in the interval $(2^{-i}, 2^{-i+1}]$. Observe that in the scenario of [IJKW10], if we were to do a similar partitioning, there would only be *two* sets, B's that are consistent with A and the B's that aren't consistent. For us, since each B is associated with a different weight, each B has a varying "degree of consistency". We now decompose Eq. (26) into all the distinct buckets and write it as

$$\sum_i \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \mathop{\mathbb{E}}_{x \sim B' \setminus A} \left[h(x) \mid B' \in \Gamma_i\right] \cdot \Pr_{B' \sim \mathcal{W}_I(A)} \left[B' \in \Gamma_i\right]. \tag{27}$$

At this point, we show with some straightforward calculations that most of the contribution in Eq. (27) comes from the buckets Γ_i that are heavy (i.e., contain many $B' \in N_{\mathcal{I}}(A)$ inside them) and for such heavy buckets, we show that

$$\mathop{\mathbb{E}}_{x \sim B' \setminus A}\left[h(x) \mid B' \in \Gamma_i\right] \cdot \Pr_{B' \sim \mathcal{W}_I(A)}\left[B' \in \Gamma_i\right] \geq \sigma \cdot \mathop{\mathbb{E}}_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A,\, \mathcal{G}}}[\widetilde{C}_A(x) \neq g(x)], \tag{28}$$

for some universal constant $\sigma < 1$. At this point, observe that the RHS of Eq. (28) is exactly the quantity we wanted to upper bound, i.e., Eq. (24) and we have shown that the LHS is at most α. This shows that $\mu \leq O(\alpha)$ and concludes the proof of the theorem. We now make this sketch more rigorous below.

We start with the first point of showing that the probability of outputting $\perp$ is small. In order to show this, we will need to use that if (A, B) is a (γ, η)-good edge (which is implied by (γ, η, α)-excellence), we can bound the number of $x \in \mathfrak{U} \setminus A$ for which less than about a $\gamma/2$-fraction of the sets $B' \in N_{\mathcal{I}}(A, x)$ have $\mathbb{E}\left[\mathcal{G}(B')\big|_A = g^{k/2}(A)\right] \geq \eta$ (these are the x's on which $\widetilde{C}_A$ is likely to output $\perp$). This claim is formalized as follows.

Lemma 4.21 (Analogue of [IJKW10, Lemma 3.9])**.** *Let $\beta \in [0,1]$ and $\lambda = 2e^{-\beta k/24}$. Fix a (γ, η)-good edge $(A, B) \in \mathcal{I}$. Let $F \subseteq \mathfrak{U} \setminus A$ be the set of elements x such that strictly less than a $(\gamma - \lambda)/2$ fraction of $B' \in N_{\mathcal{I}}(A, x)$ satisfy $\Pr_{\mathcal{G}}\left[\mathcal{G}(B')\big|_A = g^{k/2}(A)\right] \geq \eta$. Then the density $\rho(F) \stackrel{\text{def}}{=} |F|/|\mathfrak{U} \setminus A| < \beta$.*

Proof. Assume toward a contradiction that $\rho(F) = \beta$ (the case where $\rho(F) > \beta$ follows by fixing some $F' \subseteq F$ with $\rho(F') = \beta$). Define the set

$$W = \left\{B' \in N_I(A) : \mathbb{E}\left[\mathcal{G}(B')\big|_A = g^{k/2}(A)\right] \geq \eta\right\}.$$

Observe that

$$\Pr_{\substack{x \in \mathfrak{U} \setminus A \\ B' \supseteq \{x\} \cup A}}\left[x \in F \wedge B' \in W\right] = \Pr_{x \in \mathfrak{U} \setminus A}\left[x \in F\right] \cdot \Pr_{\substack{x \in \mathfrak{U} \setminus A \\ B' \supseteq \{x\} \cup A}}\left[B' \in W \mid x \in F\right] < \beta \cdot (\gamma - \lambda)/2. \tag{29}$$

On the other hand, let

$$W' = \{B' \in W : \frac{|(B' \setminus A) \cap F|}{|B' \setminus A|} \geq \beta/2\},$$

and observe that since $W' \subseteq W$, we have

$$\begin{aligned}\Pr_{\substack{x \in \mathfrak{U} \setminus A \\ B' \supseteq \{x\} \cup A}}\left[x \in F \wedge B' \in W\right] &\geq \Pr_{\substack{x \in \mathfrak{U} \setminus A \\ B' \supseteq \{x\} \cup A}}\left[x \in F \wedge B' \in W'\right] \\ &= \Pr_{B' \in N_I(A)}[B' \in W'] \cdot \Pr_{\substack{B' \in N_{\mathcal{I}}(A) \\ x \in B' \setminus A}}\left[x \in F \mid B' \in W'\right], \end{aligned} \tag{30}$$

where the equality uses the following simple fact: selecting x uniformly from $\mathfrak{U} \setminus A$ then picking a uniform $B' \in N_{\mathcal{I}}(A)$ that contains $\{x\} \cup A$ is equivalent to selecting B' uniformly from $N_{\mathcal{I}}(A)$ then picking a uniform $x \in B' \setminus A$. To complete the proof, we argue next that the last expression is larger than the bound in Eq. (29).

Since (A, B) is (γ, η)-good, we get that that $\frac{|W|}{|N_I(A)|} \geq \gamma$. Moreover, using that $\rho(F) = \beta$ and our choice of λ, by the Hoeffding bound (Lemma 2.3) the density of $W \setminus W'$ inside $N_{\mathcal{I}}(A)$ is at

most λ. Consequently, we have that $\frac{|W'|}{|N_I(A)|} \geq \gamma - \lambda$. Using this estimate along with Eq. (30),

$$\Pr_{\substack{x \in \mathfrak{U} \setminus A \\ B' \supseteq \{x\} \cup A}} \left[x \in F \wedge B' \in W\right] \geq \Pr_{B' \in N_I(A)}[B' \in W'] \cdot \Pr_{\substack{B' \in N_{\mathcal{I}}(A) \\ x \in B' \setminus A}} \left[x \in F \mid B' \in W'\right] \geq (\gamma - \lambda) \cdot \beta/2, \tag{31}$$

which stands in contradiction to Eq. (29). □

We now prove Lemma 4.19, which shows that selecting an excellence edge leads to correctness.

Proof of Lemma 4.19. Let β, λ, and T be as in the statement of the lemma, and suppose that (A, B) is a (γ, η, α)-excellent edge. We show that

$$\mathop{\mathbb{E}}_{\substack{x \sim \mathfrak{U} \setminus A, \\ \widetilde{C}_A, \mathcal{G}}} [\widetilde{C}_A = g(x)] \geq 1 - \beta - (1 - \eta(\gamma - \lambda)/2)^T - 16\alpha.$$

We first decouple the errors that stem from failing to find a consistent neighbour (in which case thee algorithm outputs $\perp$ and aborts) and those in which a consistent neighbour was found yet still the algorithm output incorrectly. To this end, by a union bound, we have

$$\Pr_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A, \mathcal{G}}} [\widetilde{C}_A(x) \neq g(x)] \leq \Pr_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A, \mathcal{G}}} [\widetilde{C}_A(x) = \perp] + \Pr_{\substack{x \in \mathfrak{U} \setminus A \\ y \sim \widetilde{C}_A(x), \mathcal{G}}} [y \neq g(x) \mid y \neq \perp]. \tag{32}$$

We will first bound the first term of the RHS of Eq. (32).

Let F be the subset of $\mathfrak{U} \setminus A$ such that $x \in F$ iff strictly less than a $(\gamma - \lambda)/2$ fraction of $B' \in N_{\mathcal{I}}(A, x)$ satisfy $\mathbb{E}[\mathcal{G}(B')|_A = g^{k/2}(A)] \geq \eta$. We have from Lemma 4.21 that $\rho(F) < \beta$, where $\rho(F)$ denotes the measure of F inside $\mathfrak{U} \setminus A$.

We now compute the probability that $\widetilde{C}_A(x) = \perp$ for an arbitrary but fixed $x \notin F$. Let W be the subset of $N_{\mathcal{I}}(A, x)$, such that $B' \in W$ iff $\mathbb{E}[\mathcal{G}(B')|_A = g^{k/2}(A)] \geq \eta$. From the assumption that $x \notin F$, we have that $\Pr_{B' \in N_{\mathcal{I}}(A,x)}[B' \in W] \geq (\gamma - \lambda)/2$. Let E_i be the event that on the i-th iteration of $\widetilde{C}_A$, $v'|_A \neq g^{k/2}(A)$, where v' is sampled in Step 2.2 from the definition of the circuit $\widetilde{C}_A$. For a fixed $i \in \{1, \ldots, T\}$, and a fixed x, we have

$$\begin{aligned}
\Pr[E_i] &= \Pr_{\widetilde{C}_A, \mathcal{G}}[\mathcal{G}(B')|_A \neq g^{k/2}(A) \mid B' \in N_{\mathcal{I}}(A, x)] \\
&= 1 - \Pr_{\widetilde{C}_A, \mathcal{G}}[\mathcal{G}(B')|_A = g^{k/2}(A) \mid B' \in N_{\mathcal{I}}(A, x)] \\
&\leq 1 - \Pr_{\widetilde{C}_A, \mathcal{G}}[\mathcal{G}(B')|_A = g^{k/2}(A) \mid B' \in W'] \cdot \Pr_{B' \in N_{\mathcal{I}}(A,x)}[B' \in W] \\
&\leq 1 - \eta \cdot \frac{\gamma - \lambda}{2}.
\end{aligned}$$

Using that in each iteration of $\widetilde{C}_A$ a fresh selection of $B' \sim N_{\mathcal{I}}(A, x)$ and $v' \sim \mathcal{G}(B')$ is made, we get that

$$\Pr_{\widetilde{C}_A, \mathcal{G}}[\widetilde{C}_A(x) = \perp \mid x \notin F] \leq \Pr_{\widetilde{C}_A, \mathcal{G}}[E_1 \wedge \cdots \wedge E_T] \leq (1 - \eta(\gamma - \lambda)/2)^T.$$

Putting together the previous estimates,

$$\begin{aligned}
\Pr_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A, \mathcal{G}}} [\widetilde{C}_A(x) = \perp] &\leq \Pr_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A, \mathcal{G}}} [x \in F] + \Pr_{\substack{x \in \mathfrak{U} \setminus A \\ \widetilde{C}_A, \mathcal{G}}} [\widetilde{C}_A(x) = \perp \mid x \notin F] \\
&\leq \beta + (1 - \eta(\gamma - \lambda)/2)^T.
\end{aligned}$$

We now bound the second term of the RHS of Eq. (32). Assuming that the output of the circuit $\widetilde{C}_A(x)$ is not $\perp$, we have that at some iteration $\widetilde{C}_A(x)$ picks $B' \sim N_{\mathcal{I}}(A, x)$ and $v' \sim \mathcal{G}(B')$ such that $v'|_A = g^{k/2}(A)$.

For convenience, for $x \in \mathfrak{U} \setminus A$ let us define $h(x)$ to be the conditional probability that $\widetilde{C}_A$ produces an incorrect answer on x (over the randomness of $\widetilde{C}_A$ and $\mathcal{G}$), given that it does not output $\perp$. Our goal is to show that

$$\Pr_{\substack{x \in \mathfrak{U} \setminus A \\ y \sim \widetilde{C}_A(x),\, \mathcal{G}}} [y \neq g(x) \mid y \neq \perp] = \mathop{\mathbb{E}}_{x \sim \mathfrak{U} \setminus A}[h(x)] \tag{33}$$

is small. To show this, it will be convenient to use the following notation. For $B \in N_{\mathcal{I}}(A)$,

$p_{\mathsf{cons}}(B) = \Pr_{\mathcal{G}}\left[\mathcal{G}(B)|_A = g^{k/2}(A)\right]$.
$p_{\mathsf{tot}}(A) = \sum_{B \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B)$.
$p_{\mathsf{tot}}(x) = \sum_{B \in N_{\mathcal{I}}(A,x)} p_{\mathsf{cons}}(B)$.
$p_{\mathsf{used}}(x, B) = p_{\mathsf{cons}}(B)/p_{\mathsf{tot}}(x)$.
$p_{\mathsf{err}}(x, B) = \Pr_{v \sim \mathcal{G}(B)}[v|_x \neq g(x) \mid v|_A = g^{k/2}(A)]$.

Note that all these values depend on A. It is not hard to see that

$$h(x) = \sum_{B \in N_{\mathcal{I}}(A,x)} p_{\mathsf{err}}(x, B) \cdot p_{\mathsf{used}}(x, B). \tag{34}$$

Moreover, since

$$p_{\mathsf{err}}(x, B) = \frac{\Pr_{v \sim \mathcal{G}(B)}[v|_x \neq g(x) \wedge v|_A = g^{k/2}(A)]}{\Pr_{\mathcal{G}}[\mathcal{G}(B)|_A = g^{k/2}(A)]} = \frac{\Pr_{v \sim \mathcal{G}(B)}[v|_x \neq g(x) \wedge v|_A = g^{k/2}(A)]}{p_{\mathsf{cons}}(B)},$$

we have

$$h(x) = \frac{\sum_{B'' \in N_{\mathcal{I}}(A,x)} \Pr_{\mathcal{G}}[\mathcal{G}(B'')|_x \neq g(x) \wedge \mathcal{G}(B'')|_A = g^{k/2}(A)]}{\sum_{B'' \in N_{\mathcal{I}}(A,x)} P_{\mathcal{G}}[\mathcal{G}(B'')|_A = g^{k/2}(A)]},$$

or equivalently,

$$h(x) = \frac{\sum_{B'' \in N_{\mathcal{I}}(A,x)} \Pr_{v'' \sim \mathcal{G}(B'')}[v''|_x \neq g(x) \wedge v''|_A = g^{k/2}(A)]}{p_{\mathsf{tot}}(x)}. \tag{35}$$

Next, we upper bound the quantity on the RHS of Equation (33). For convenience, let

$$\mu = \mathop{\mathbb{E}}_{x \sim \mathfrak{U} \setminus A}[h(x)].$$

As we are interested in bounding the probability that $\widetilde{C}_A$ outputs an erroneous value when it samples an excellent edge, we would like to relate μ to the excellence parameter α. We do this by first relating α and then μ to the following intermediate expression:

$$\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)}\left[\mathop{\mathbb{E}}_{x \sim B' \setminus A}[h(x)]\right] = \frac{1}{p_{\mathsf{tot}(A)}} \cdot \sum_{B' \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B') \mathop{\mathbb{E}}_{x \sim B' \setminus A}[h(x)], \tag{36}$$

where $\mathcal{W}_I(A)$ is the distribution supported on $N_{\mathcal{I}}(A)$ with each $B' \in N_{\mathcal{I}}(A)$ being sampled with probability $p_{\mathsf{cons}}(B')/p_{\mathsf{tot}}(A)$.

On the one hand, using Equation (35), we can rewrite the RHS of Equation (36) as:

$$\frac{1}{p_{\mathsf{tot}}(A)} \cdot \sum_{B' \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B') \cdot \frac{1}{k/2} \cdot \sum_{x \in B' \setminus A} \frac{1}{p_{\mathsf{tot}}(x)} \cdot \sum_{B'' \in N_{\mathcal{I}}(A,x)} \Pr_{\mathcal{G}}[\mathcal{G}(B'')|_x \neq g(x) \wedge \mathcal{G}(B'')|_A = g^{k/2}(A)].$$

Note that in the expression above, every fixed (x, B''), where $x \in \mathfrak{U} \setminus A$ and $B'' \in N_{\mathcal{I}}(A, x)$, contributes a value

$$\frac{1}{p_{\mathsf{tot}}(A)} \cdot \left(\sum_{B' \in N_{\mathcal{I}}(A,x)} p_{\mathsf{cons}}(B') \right) \cdot \frac{1}{k/2} \cdot \frac{1}{p_{\mathsf{tot}}(x)} \cdot \Pr_{\mathcal{G}}[\mathcal{G}(B'')|_x \neq g(x) \wedge \mathcal{G}(B'')|_A = g^{k/2}(A)],$$

since it appears in the sum with a corresponding factor $p_{\mathsf{cons}}(B')$ for each B' in $N_{\mathcal{I}}(A, x)$. Since the corresponding sum of $p_{\mathsf{cons}}(B')$ is precisely $p_{\mathsf{tot}}(x)$, it follows that every (x, B'') appears in Equation (36) with a contribution of

$$\frac{1}{p_{\mathsf{tot}}(A)} \cdot \frac{1}{k/2} \cdot \Pr_{\mathcal{G}}[\mathcal{G}(B'')|_x \neq g(x) \wedge \mathcal{G}(B'')|_A = g^{k/2}(A)], \tag{37}$$

and Equation (36) is precisely the sum of these contributions over all (x, B'').

On the other hand, using that the edge (A, B) from the statement of the lemma is α-excellent, we know that

$$\begin{aligned}
\alpha &\geq \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mathop{\mathbb{E}}_{x \sim B' \setminus A} [p_{\mathsf{err}}(x, B')] \right] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \cdot \sum_{B' \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B') \mathop{\mathbb{E}}_{x \sim B' \setminus A} [p_{\mathsf{err}}(x, B')] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \cdot \sum_{B' \in N_{\mathcal{I}}(A)} \frac{1}{k/2} \sum_{x \in B' \setminus A} \Pr_{\mathcal{G}}[\mathcal{G}(B')|_x \neq g(x) \wedge \mathcal{G}(B')|_A = g^{k/2}(A)].
\end{aligned}$$

In the expression above, each edge (x, B') also contributes a value equal to that in Equation (37). Consequently, we get from the discussion above that

$$\alpha \geq \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mathop{\mathbb{E}}_{x \sim B' \setminus A} [h(x)] \right]. \tag{38}$$

Next, we move on to relate $\mu = \mathbb{E}_{x \sim \mathfrak{U} \setminus A}[h(x)]$ to the RHS of this inequality. Unlike in the classical case, this calculation is more involved as the B' in Equation (38) is not sampled uniformly from $N_{\mathcal{I}}(A)$ but from $\mathcal{W}_{\mathcal{I}}(A)$ (where sampling B' is proportional to $p_{\mathsf{cons}}(B')$). To deal with this, we partition the elements $B' \in N_{\mathcal{I}}(A)$ into buckets based on their value $p_{\mathsf{cons}}(B') \in [0, 1]$. Let $\ell = 5 \cdot \log(1/(\gamma \cdot \eta))$, and for $i \in \{0, 1, \ldots, \ell\}$, set

$$\Gamma_i = \{ B' \in N_{\mathcal{I}}(A) \mid p_{\mathsf{cons}}(B') \in (2^{-i-1}, 2^{-i}] \}.$$

We also define the exceptional bucket $\Gamma_{\ell+1}$ as follows:

$$\Gamma_{\ell+1} = \{ B' \in N_{\mathcal{I}}(A) \mid p_{\mathsf{cons}}(B') \leq 2^{-\ell-1} \}.$$

Note that the buckets are disjoint, and that $N_{\mathcal{I}}(A) = \bigcup_{i=0}^{\ell+1} \Gamma_i$. For $B' \in N_{\mathcal{I}}(A)$, let $\mu(B') = \mathbb{E}_{x \sim B' \setminus A}[h(x)]$. Then

$$\begin{aligned}
\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mathop{\mathbb{E}}_{x \sim B' \setminus A} [h(x)] \right] &= \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mu(B') \right] \\
&= \sum_{i=0}^{\ell+1} \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mu(B') \mid B' \in \Gamma_i \right] \cdot \Pr_{B' \sim \mathcal{W}_I(A)} \left[B' \in \Gamma_i \right].
\end{aligned} \tag{39}$$

Our goal is to lower bound the expression in Equation (39). Towards that, we aim to bound each term in the expression individually. The following simple claim will be useful. It shows that, in our analysis, for $0 \leq i \leq \ell$, we can replace sampling a B' from Γ_i according to $\mathcal{W}_{\mathcal{I}}(A)$ with sampling a uniformly random $B' \sim \Gamma_i$.

Claim 4.22. *For every $0 \leq i \leq \ell$,*

$$\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mu(B') \mid B' \in \Gamma_i\right] \;\geq\; \frac{1}{2} \cdot \mathop{\mathbb{E}}_{B' \sim \Gamma_i} \left[\mu(B')\right].$$

Proof of Claim 4.22. Indeed,

$$\begin{aligned}
\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mu(B') \mid B' \in \Gamma_i\right] &= \frac{\sum_{B' \in \Gamma_i} \frac{p_{\mathsf{cons}}(B')}{p_{\mathsf{tot}}(A)} \cdot \mu(B')}{(1/p_{\mathsf{tot}}(A)) \cdot \sum_{B' \in \Gamma_i} p_{\mathsf{cons}}(B')} \\
&= \frac{\sum_{B' \in \Gamma_i} p_{\mathsf{cons}}(B') \cdot \mu(B')}{\sum_{B' \in \Gamma_i} p_{\mathsf{cons}}(B')} \\
&\geq \frac{\sum_{B' \in \Gamma_i} 2^{-i-1} \cdot \mu(B')}{|\Gamma_i| \cdot 2^{-i}} \\
&= \frac{1}{2} \cdot \left(\frac{1}{|\Gamma_i|} \cdot \sum_{B' \in \Gamma_i} \mu(B') \right) \\
&= \frac{1}{2} \cdot \mathop{\mathbb{E}}_{B' \sim \Gamma_i} \left[\mu(B')\right].
\end{aligned}$$

□

Notice that this bound does not work for $\Gamma_{\ell+1}$. However, it suffices for us to show that $\Pr_{B' \sim \mathcal{W}_{\mathcal{I}}(A)}[B' \in \Gamma_{\ell+1}]$ is "small" and can be omitted while determining the lower bound for Equation (39). In fact, we are able to claim something slightly stronger as shown next. For $0 \leq i \leq \ell$, we say that bucket Γ_i is *large* if $|\Gamma_i| \geq (\gamma \cdot \eta)^5 \cdot |N_{\mathcal{I}}(A)|$. Otherwise, we say that it is *small*. For convenience, let $w_i = \sum_{B' \in \Gamma_i} p_{\mathsf{cons}}(B')$. Recall that (A, B) is a (γ, η)-good edge, and therefore

$$\sum_{i=0}^{\ell+1} w_i \;=\; p_{\mathsf{tot}}(A) \;\geq\; \gamma \cdot \eta \cdot |N_{\mathcal{I}}(A)|. \tag{40}$$

Claim 4.23. *The following upper bounds hold.*

(i) For $0 \leq i \leq \ell$, if Γ_i is small then

$$\Pr_{B' \sim \mathcal{W}_I(A)}[B' \in \Gamma_i] \leq (\gamma \cdot \eta)^4.$$

(ii) Moreover, in the special case where $i = \ell + 1$, we have

$$\Pr_{B' \sim \mathcal{W}_I(A)}[B' \in \Gamma_{\ell+1}] \leq (\gamma \cdot \eta)^4.$$

Proof of Claim 4.23. For the proof of Item (i), we rely on Equation (40) and on the smallness of Γ_i:

$$\Pr_{B' \sim \mathcal{W}_I(A)}[B' \in \Gamma_i] = \frac{w_i}{p_{\mathsf{tot}}(A)} \leq \frac{|\Gamma_i| \cdot 2^{-i}}{p_{\mathsf{tot}}(A)} \leq \frac{|\Gamma_i|}{p_{\mathsf{tot}}(A)} \leq \frac{(\gamma \cdot \eta)^5 \cdot |N_{\mathcal{I}}(A)|}{\gamma \cdot \eta \cdot |N_{\mathcal{I}}(A)}} = (\gamma \cdot \eta)^4.$$

For the proof of Item (ii), we rely on Equation (40) an on the upper bound on $p_{\mathsf{cons}}(B')$ for $B' \in \Gamma_{\ell+1}$:

$$\Pr_{B' \sim \mathcal{W}_I(A)}[B' \in \Gamma_{\ell+1}] = \frac{w_{\ell+1}}{p_{\mathsf{tot}}(A)} \leq \frac{|\Gamma_{\ell+1}| \cdot 2^{-\ell-1}}{p_{\mathsf{tot}}(A)} \leq \frac{|N_{\mathcal{I}}(A)| \cdot 2^{-\ell-1}}{p_{\mathsf{tot}}(A)} \leq \frac{2^{-\ell-1}}{\gamma \cdot \eta} \leq (\gamma \cdot \eta)^4,$$

where the last inequality uses $\ell = 5 \cdot (\log(1/\gamma \cdot \eta))$. □

Intuitively, Claim 4.23 shows that the only significant terms in Equation (39) are the ones coming from buckets Γ_i with $0 \leq i \leq \ell$ that are large. Moreover, since there are $O(\log(1/\gamma\eta))$ terms, the combined probability weight of the insignificant terms is also small.

Claim 4.24. *A random $B' \sim \mathcal{W}_I(A)$ is likely to belong to a large bucket Γ_i with $0 \leq i \leq \ell$, i.e.,*

$$\Pr_{B' \sim \mathcal{W}_I(A)}[B' \textit{ is in a large bucket } \Gamma_i \textit{ for some } 0 \leq i \leq \ell\,] \geq 1/2.$$

Proof of Claim 4.24. Using Claim 4.23 and a union bound over buckets,

$$\Pr_{B' \sim \mathcal{W}_I(A)}[B' \text{ is in a small bucket or } B' \in \Gamma_{\ell+1}] \leq (\ell+2) \cdot (\gamma\eta)^4 \leq 6 \cdot \log(1/(\gamma\eta)) \cdot (\gamma\eta)^4 \leq 1/2,$$

where the last inequality uses the assumption of the lemma that $\gamma, \eta \leq 1/10$. □

Finally, the next claim establishes that when $0 \leq i \leq \ell$ and Γ_i is large, $\mathbb{E}_{B' \sim \Gamma_i}\left[\mu(B')\right] = \Omega(\mu)$, provided that μ is not too small and k is large enough. (Recall that if μ is small, specifically less than the excellence parameter α, we are already done.)

Claim 4.25. *Suppose that $\mu \geq \alpha$. Let $0 \leq i \leq \ell$ and Γ_i be a large bucket. Then*

$$\mathop{\mathbb{E}}_{B' \sim \Gamma_i}\left[\mu(B')\right] \;\geq\; \frac{\mu}{2} \cdot \left(1 - \frac{2e^{-k\alpha/12}}{(\gamma\eta)^5}\right) \;\geq\; \frac{\mu}{4}.$$

Proof of Claim 4.25. Let

$$\mathsf{Bad} = \{B' \in N_{\mathcal{I}}(A) \mid \mu(B') < \mu/2\}.$$

Then, by the Hoeffding bound, $\Pr_{B' \sim N_{\mathcal{I}}(A)}[B' \in \mathsf{Bad}] \leq 2e^{-k\mu/12} \leq 2e^{-k\alpha/12}$, using that $\mu \geq \alpha$. Note that, since Γ_i is large, $|\Gamma_i| \geq (\gamma\eta)^5 \cdot |N_{\mathcal{I}}(A)|$. For convenience, let $\lambda = 2e^{-k\alpha/12}$. Then

$$\frac{|\mathsf{Bad}|}{|\Gamma_i|} \leq \frac{\lambda \cdot |N_{\mathcal{I}}(A)|}{(\gamma\eta)^5 |N_{\mathcal{I}}(A)|} = \frac{\lambda}{(\gamma\eta)^5}.$$

Consequently,

$$\begin{aligned}
\mathop{\mathbb{E}}_{B' \sim \Gamma_i}\left[\mu(B')\right] \;&\geq\; \mathop{\mathbb{E}}_{B' \sim \Gamma_i}\left[\mu(B') \mid B' \notin \mathsf{Bad}\right] \cdot \Pr_{B' \sim \Gamma_i}\left[B' \notin \mathsf{Bad}\right] \\
&=\; \mathop{\mathbb{E}}_{B' \sim \Gamma_i \setminus \mathsf{Bad}}\left[\mu(B')\right] \cdot \frac{|\Gamma_i| - |\mathsf{Bad}|}{|\Gamma_i|} \\
&\geq\; \frac{\mu}{2} \cdot \left(1 - \frac{|\mathsf{Bad}|}{|\Gamma_i|}\right) \;\geq\; \frac{\mu}{2} \cdot \left(1 - \frac{\lambda}{(\gamma\eta)^5}\right),
\end{aligned}$$

where the second inequality used the definition of Bad. Claim 4.25 follows using the value of λ and the hypothesis of the lemma. □

We are ready to conclude the proof of Lemma 4.19. If $\mu \le \alpha$, we are done. So we assume from now on that $\mu \ge \alpha$. From Equations (38) and (39), we have

$$\begin{aligned}
\alpha &\ge \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mathop{\mathbb{E}}_{x \sim B' \setminus A} [h(x)] \right] \\
&= \sum_{i=0}^{\ell+1} \mathop{\mathbb{E}}_{B' \sim \mathcal{W}_I(A)} \left[\mu(B') \mid B' \in \Gamma_i \right] \cdot \Pr_{B' \in \mathcal{W}_I(A)} \left[B' \in \Gamma_i \right] \\
\text{(Omitting terms + Claim 4.22)} &\ge \sum_{\substack{0 \le i \le \ell \\ \Gamma_i \text{ is large}}} \frac{1}{2} \cdot \mathop{\mathbb{E}}_{B' \sim \Gamma_i} \left[\mu(B') \right] \cdot \Pr_{B' \sim \mathcal{W}_I(A)} \left[B' \in \Gamma_i \right] \\
\text{(By Claim 4.25)} &\ge \sum_{\substack{0 \le i \le \ell \\ \Gamma_i \text{ is large}}} \frac{\mu}{8} \cdot \Pr_{B' \sim \mathcal{W}_I(A)} \left[B' \in \Gamma_i \right] \\
&= \frac{\mu}{8} \cdot \Pr_{B' \sim \mathcal{W}_I(A)} \left[B' \text{ is in a large bucket } \Gamma_i \text{ for some } 0 \le i \le \ell \right] \\
\text{(By Claim 4.24)} &\ge \frac{\mu}{16}.
\end{aligned}$$

This shows that $\mu \le 16\alpha$, which completes the proof of Lemma 4.19. □

4.3.3 Excellent edges are abundant

In this section, we prove Lemma 4.20, which loosely speaking, shows that if the algorithm $\mathcal{G}$ non-trivially agrees with g^k, then there are many edges that satisfy the excellence condition. Recall that an edge is said to be (η, γ, α)-*excellent* if it is: (1) (η, γ)-good, and (2) satisfies

$$\mathop{\mathbb{E}}_{B' \sim \mathcal{W}_{\mathcal{I}}(A)} \left[\mathsf{ErrCons}(A, B') \right] \le \alpha,$$

where $\mathcal{W}_{\mathcal{I}}(A)$ gives each edge (A, B') weight according to its probability of being correct on A (see Definition 4.18). Our first lemma shows that the first condition holds, i.e., that under the assumption that our algorithm $\mathcal{G}$ computes g^k with probability at least ε, there are many good edges.

Lemma 4.26. *If $\mathbb{E}_{B \in \mathcal{S}_k} [\mathsf{Corr}(B)] \ge \varepsilon$, for any $0 < \eta, \gamma, \xi$ such that $\eta + \gamma + \xi = \varepsilon$, we have that at least a ξ-fraction of $(A, B) \in \mathcal{I}$ are (γ, η)-good.*

Proof. We prove the contrapositive statement in the lemma. In this direction, let

$$\left| \left\{ (A, B) \in \mathcal{I} : B \in N_{\mathcal{I}}(A) \text{ and } (A, B) \text{ is } (\gamma, \eta)\text{-good} \right\} \right| < \xi \cdot |I|. \tag{41}$$

We define the set

$$G = \left\{ A \in \mathcal{S}_{k/2} : \mathop{\mathbb{E}}_{B' \in N_{\mathcal{I}}(A)} [\mathsf{Corr}(B') \ge \eta] \ge \gamma \right\}$$

of As who have at least an γ-fraction of η-correct neighbors.

By our assumption in Equation (41), we have that $|G| < \xi |\mathcal{S}_{k/2}|$ and thus

$$\mathop{\mathbb{E}}_{A \in \mathcal{S}_{k/2}} [A \in G] \cdot \mathop{\mathbb{E}}_{\substack{A \in \mathcal{S}_{k/2}, \\ B \in N_I(A)}} [\mathsf{Corr}(B) | A \in G] < \xi \cdot 1 = \xi. \tag{42}$$

We also have that

$$\mathop{\mathbb{E}}_{\substack{A\in\mathcal{S}_{k/2}\\B\in N_I(A)}}[A\notin G\wedge \mathsf{Corr}(B)<\eta]\cdot \mathop{\mathbb{E}}_{\substack{A\in\mathcal{S}_{k/2}\\B\in N_I(A)}}\left[\mathsf{Corr}(B)\Big|^{A\notin G}_{\mathsf{Corr}(B)<\eta}\right]\le 1\cdot\eta=\eta, \tag{43}$$

and

$$\mathop{\mathbb{E}}_{\substack{A\in\mathcal{S}_{k/2}\\B\in N_I(A)}}[A\notin G\wedge \mathsf{Corr}(B)\ge\eta]\cdot \mathop{\mathbb{E}}_{\substack{A\in\mathcal{S}_{k/2}\\B\in N_I(A)}}\left[\mathsf{Corr}(B)\Big|^{A\notin G}_{\mathsf{Corr}(B)\ge\eta}\right]<\gamma\cdot 1=\gamma, \tag{44}$$

where the first inequality above used the fact that conditioned on $A\notin G$, then the probability of a uniformly random $B'\in N_{\mathcal{I}}(A)$ being η-correct is at most γ (by definition of G).

We have that

$$\mathop{\mathbb{E}}_{B\in\mathcal{S}_k}[\mathsf{Corr}(B)]=\mathop{\mathbb{E}}_{\substack{A\in\mathcal{S}_{k/2},\\B\in N_I(A)}}[\mathsf{Corr}(B)]<\ \text{Eq. (42)}+\text{Eq. (43)}+\text{Eq. (44)}<\xi+\eta+\gamma=\varepsilon,$$

where the first equality above used that we can obtain the uniform distribution on $B\in S_k$ by uniformly picking $A\in S_{k/2}$ and then considering a random $B\in N_{\mathcal{I}}(A)$; and the last equality is by the assumption of the lemma. This concludes the proof of the statement. □

Next we will use Lemma 4.26 to strengthen the foregoing conclusion and show that if $\mathcal{G}$ computes g^k with probability at least ε, then not only is the number of good edges large, but rather the number of *excellent* edges is also large. Formally, we prove that if $\mathbb{E}_{B\in\mathcal{S}_k}[\mathsf{Corr}(B)]\ge\varepsilon$ then at least an $(\varepsilon/3-\frac{62208}{\alpha^3\cdot\varepsilon^5}\cdot e^{-\frac{\alpha}{96}\cdot k})$-fraction of the edges $(A,B)\in\mathcal{I}$ are $(\varepsilon/3,\varepsilon/3,\alpha)$-excellent. Note that the fraction of edges with this property is at least $\varepsilon/6$ if $k=\frac{100}{\alpha}\cdot(15+3\log(1/\alpha)+6\log(1/\varepsilon))$, so this does show that a noticeable fraction of edges are excellent as long as k is not too small.

Compared with its counterpart in [IJKW10], in the proof of Lemma 4.20 we face additional difficulties due to the asymmetries in the definition of excellent edges, which in this paper refer to the more involved distribution $\mathcal{W}_{\mathcal{I}}(A)$.

Proof of Lemma 4.20. Consider random choices of $A\sim\mathcal{S}_{k/2}$ and $B\sim N_{\mathcal{I}}(A)$. We would like to lower bound

$$\Pr_{A,B}[(A,B)\text{ is }(\varepsilon/3,\varepsilon/3,\alpha)\text{-excellent}].$$

In order to show that an $(\varepsilon/3,\varepsilon/3)$-good edge $(A,B)\in\mathcal{I}$ is also $(\varepsilon/3,\varepsilon/3,\alpha)$-excellent, we need to show that

$$\mathop{\mathbb{E}}_{B'\sim\mathcal{W}_I(A)}[\mathsf{ErrCons}(A,B')]\le\alpha.$$

It will be useful to introduce the following probability space and event. In addition to sampling $A\sim\mathcal{S}_{k/2}$ and $B\sim N_{\mathcal{I}}(A)$, independently sample $B'\sim N_{\mathcal{I}}(A)$, $v'=\mathcal{G}(B')$, and $x\sim B'\setminus A$. Let $\mathsf{Err}(B',v')=\{x\in B'\setminus A\mid v'|_x\neq g(x)\}$ be the set of xs in $B'\setminus A$ for which $v'|_x$ disagrees with $g(x)$, and $\mathsf{err}(B',v')=|\mathsf{Err}(B',v')|/(k/2)$. We introduce the following event.

$\mathcal{E}(A,B,B',v',x)$: The following conditions hold:

$v'|_A=g^{k/2}(A)$.
$\mathsf{err}(B',v')>\alpha/4$.
$v'|_x\neq g(x)$.

By symmetry, when analysing $\Pr[\mathcal{E}]$ we can select B' first (accordingly sampling v' and x depending on it), followed by choices of a random A contained in B' and a random B that contains A. Note that

$$\Pr[\mathcal{E}] = \frac{1}{|\mathcal{S}_k|} \sum_{B' \in \mathcal{S}_k} \Pr[\mathcal{E} \ \mid \ B' \text{ fixed}],$$

by "B' fixed" we mean the event that the random choice gives a fixed $B' \in \mathcal{S}_k$. We now decompose each term in the sum above according to the possible values of $\lambda = \mathsf{err}(B', v')$ for $v' \sim \mathcal{G}(B')$. Since for $\mathcal{E}$ to hold this value must be larger than $\alpha/4$, we get for each fixed B':

$$\Pr[\mathcal{E} \mid B' \text{ fixed}] = \sum_{\lambda > \alpha/4} \Pr[\mathcal{E} \mid B' \text{ is fixed} \wedge \mathsf{err}(B', v') = \lambda] \cdot \Pr_{v'}[\mathsf{err}(B', v') = \lambda].$$

For a fixed B' and any choice of v' with $\mathsf{err}(B', v') = \lambda$, if $\mathsf{Err}(B', v') \cap A \neq \emptyset$ then it cannot be the case that $v'|_A = g^{k/2}(A)$. Consequently, for $\mathcal{E}$ to hold A must avoid the set $\mathsf{Err}(B', v')$, and in addition, x must be selected from $\mathsf{Err}(B', v')$. It follows from the Hoeffding bound using $\lambda > \alpha/4$ that

$$\Pr[\mathcal{E} \mid B' \text{ is fixed} \wedge \mathsf{err}(B', v') = \lambda] \leq 2 \cdot e^{-\frac{\alpha}{96} \cdot k} \cdot \lambda.$$

Putting together these estimates, and using that $\lambda \leq 1$, we have

$$\begin{aligned}
\Pr[\mathcal{E}] &\leq \frac{1}{|\mathcal{T}|} \sum_{B' \in \mathcal{T}} \sum_{\lambda > \alpha/4} 2 \cdot e^{-\frac{\alpha}{96} \cdot k} \cdot \Pr_{v' \sim \mathcal{G}(B')}[\mathsf{err}(B', v') = \lambda] \\
&\leq \frac{1}{|\mathcal{T}|} \sum_{B' \in \mathcal{T}} 2 \cdot e^{-\frac{\alpha}{96} \cdot k} \cdot \sum_{\lambda > \alpha/4} \Pr_{v' \sim \mathcal{G}(B')}[\mathsf{err}(B', v') = \lambda] \\
&\leq \frac{1}{|\mathcal{T}|} \sum_{B' \in \mathcal{T}} 2 \cdot e^{-\frac{\alpha}{96} \cdot k} \leq 2 \cdot e^{-\frac{\alpha}{96} \cdot k}.
\end{aligned}$$

Next, notice that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{B'' \sim \mathcal{W}_I(A)}[\mathsf{ErrCons}(A, B'')] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \sum_{B'' \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B'') \mathop{\mathbb{E}}_{x \sim B'' \setminus A} \left[\Pr_{v'' \sim \mathcal{G}(B'')} \left[v''|_x \neq g(x) \ \mid \ v''|_A = g^{k/2}(A) \right] \right] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \sum_{B'' \in N_{\mathcal{I}}(A)} \mathop{\mathbb{E}}_{x \sim B'' \setminus A} \left[\Pr_{v'' \sim \mathcal{G}(B'')} \left[v''|_x \neq g(x) \wedge v''|_A = g^{k/2}(A) \right] \right] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \sum_{B'' \in N_{\mathcal{I}}(A)} \frac{1}{(k/2)} \sum_{x \in B'' \setminus A} \Pr_{v'' \sim \mathcal{G}(B'')} \left[v''|_x \neq g(x) \wedge v''|_A = g^{k/2}(A) \right] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \sum_{B'' \in N_{\mathcal{I}}(A)} \Pr_{\substack{v'' \sim \mathcal{G}(B''), \\ x \sim B'' \setminus A}} [v''|_A = g^{k/2}(A) \wedge x \in \mathsf{Err}(B'', v'')] \\
&= \frac{1}{p_{\mathsf{tot}}(A)} \sum_{B'' \in N_{\mathcal{I}}(A)} p_{\mathsf{cons}}(B'') \Pr_{v'' \sim \mathcal{G}(B''),\, x \sim B'' \setminus A} [x \in \mathsf{Err}(B'', v'') \mid v''|_A = g^{k/2}(A)]
\end{aligned}$$

$$= \mathop{\mathbb{E}}_{B'' \sim \mathcal{W}_I(A)} \left[\Pr_{\substack{v'' \sim \mathcal{G}(B''), \\ x \sim B'' \setminus A}} \left[x \in \mathsf{Err}(B'', v'') \mid v''|_A = g^{k/2}(A) \right] \right],$$

where in the second equality we use the fact that

$$p_{\mathsf{cons}}(B'') \Pr_{v'' \sim \mathcal{G}(B'')} \left[v''|_x \neq g(x) \;\mid\; v''|_A = g^{k/2}(A) \right] = \Pr_{v'' \sim \mathcal{G}(B'')} \left[v''|_x \neq g(x) \wedge v''|_A = g^{k/2}(A) \right].$$

This motivates the following definition. We say that a set B'' is A-heavy if

$$\Pr_{\substack{v'' \sim \mathcal{G}(B''), \\ x \sim B'' \setminus A}} \left[x \in \mathsf{Err}(B'', v'') \mid v''|_A = g^{k/2}(A) \right] > \alpha/2.$$

Then, in order to show that an $(\varepsilon/3, \varepsilon/3)$-good edge (A, B) is $(\varepsilon/3, \varepsilon/3, \alpha)$-excellent, it suffices to prove that

$$\Pr_{B'' \sim \mathcal{W}_I(A)} [B'' \text{ is } A\text{-heavy}] \leq \alpha/2.$$

Consider the following event, which only depends on A and B, but can also be considered over the probability space of event $\mathcal{E}$:

$\mathcal{E}_1(A, B)$: The following conditions hold:

(A, B) is $(\varepsilon/3, \varepsilon/3)$-good.
$\Pr_{B'' \sim \mathcal{W}_I(A)}[B'' \text{ is } A\text{-heavy}] > \alpha/2$. (Note that this is a property of A.)

We want to show that the event $\mathcal{E}_1$ happens with small probability. In other words, we show that $\Pr[\mathcal{E}_1] = \Pr[\mathcal{E}] / \Pr[\mathcal{E} \mid \mathcal{E}_1]$ is small by arguing that $\Pr[\mathcal{E} \mid \mathcal{E}_1]$ is not too small. We make use of the following claim, whose proof is deferred:

Claim 4.27. *For every $(\varepsilon/3, \varepsilon/3)$-good edge (A, B),*

if $\Pr_{B'' \sim \mathcal{W}_I(A)}[B''$ is A-heavy$] > \alpha/2$ then $\Pr_{B' \sim N_{\mathcal{I}}(A)}[B'$ is A-heavy $\wedge\, p_{\mathsf{cons}}(B') \geq \varepsilon^*] > \frac{\alpha}{108} \cdot \varepsilon^3$,

where $\varepsilon^* \stackrel{\text{def}}{=} (\alpha/8) \cdot (\varepsilon^2/9)$ and $p_{\mathsf{cons}}(B') = \Pr_{v' \sim \mathcal{G}(B')}[v'|_A = g^{k/2}(A)]$.

Assuming this claim, we proceed as follows. Under event $\mathcal{E}_1$, it follows from Claim 4.27 that

$$\Pr_{B' \sim N_{\mathcal{I}}(A)} [B' \text{ is } A\text{-heavy} \wedge p_{\mathsf{cons}}(B') \geq \varepsilon^*] \geq (\alpha/108) \cdot \varepsilon^3.$$

Therefore, conditioning on $\mathcal{E}_1$, with probability at least $(\alpha/108) \cdot \varepsilon^3$ over the choices of A, B and B' we get

$$p_{\mathsf{cons}}(B') \geq \varepsilon^* \quad \text{and} \quad \Pr_{v' \sim \mathcal{G}(B'),\, x \sim B' \setminus A} [x \in \mathsf{Err}(B', v') \mid v'|_A = g^{k/2}(A)] > \alpha/2. \tag{45}$$

We can write the latter probability as follows:

$$\Pr \left[x \in \mathsf{Err}(B', v') \wedge \mathsf{err}(B', v') \leq \alpha/4 \mid v'|_A = g^{k/2}(A) \right] + \Pr \left[x \in \mathsf{Err}(B', v') \wedge \mathsf{err}(B', v') > \alpha/4 \mid v'|_A = g^{k/2}(A) \right].$$

Since the leftmost probability is at most $\alpha/4$, it follows that for a B' of this form

$$\Pr_{\substack{v' \sim \mathcal{G}(B'), \\ x \sim B' \setminus A}} [x \in \mathsf{Err}(B', v') \wedge \mathsf{err}(B', v') > \alpha/4 \mid v'|_A = g^{k/2}(A)] \geq \alpha/4.$$

Note that $x \in \mathsf{Err}(B', v')$ is equivalent to $v'|_x \neq g(x)$. Using this and rewriting the probability inequality above using the expression $\Pr[F_1 \mid F_2] = \Pr[F_1 \wedge F_2]/\Pr[F_2]$,

$$\begin{aligned}
&\Pr_{\substack{v' \sim \mathcal{G}(B'),\\ x \sim B' \setminus A}}[v'|_x \neq g(x) \wedge \mathsf{err}(B', v') > \alpha/4 \wedge v'|_A = g^{k/2}(A)] \\
&= \Pr_{v' \sim \mathcal{G}(B')}[v'|_x \neq g(x) \wedge \mathsf{err}(B', v') > \alpha/4 \mid v'|_A = g^{k/2}(A)] \Pr_{v' \sim \mathcal{G}(B')}[v'|_A = g^{k/2}(A)] \\
&\geq (\alpha/4) \cdot p_{\mathsf{cons}}(B') \\
&\geq (\alpha/4) \cdot (\alpha/8) \cdot (\varepsilon^2/9),
\end{aligned}$$

where we have used $p_{\mathsf{cons}}(B') \geq \varepsilon^*$ and $p_{\mathsf{cons}}(B') = \Pr_{v' \sim \mathcal{G}(B')}[v'|_A = g^{k/2}(A)]$. Overall, combining this probability lower bound and the probability that the conditions in Equation (45) hold, it follows that

$$\Pr[\mathcal{E} \mid \mathcal{E}_1] \geq (\alpha/108) \cdot \varepsilon^3 \cdot (\alpha/4) \cdot (\alpha/8) \cdot (\varepsilon^2/9) = \frac{\alpha^3 \cdot \varepsilon^5}{31104}.$$

Since $\Pr[\mathcal{E}_1] \leq \Pr[\mathcal{E}]/\Pr[\mathcal{E} \mid \mathcal{E}_1]$,

$$\Pr[\mathcal{E}_1] \leq \frac{62208 \cdot e^{-\frac{\alpha}{96} \cdot k}}{\alpha^3 \cdot \varepsilon^5}.$$

Finally, from the definition of event $\mathcal{E}_1$ and the discussion above we get

$$\Pr_{\substack{A \sim \mathcal{S}_{k/2},\\ B \sim N_{\mathcal{I}}(A)}}[(A, B) \text{ is } (\varepsilon/3, \varepsilon/3, \alpha)\text{-excellent}] \geq \Pr_{A,\, B}[(A, B) \text{ is } (\varepsilon/3, \varepsilon/3)\text{-good}] - \Pr_{A,\, B}[\mathcal{E}_1].$$

This implies using Lemma 4.26 and our probability estimate for $\mathcal{E}_1$ that the probability that a random edge (A, B) is $(\varepsilon/3, \varepsilon/3, \alpha)$-excellent is at least

$$\varepsilon/3 - \frac{62208 \cdot e^{-\frac{\alpha}{96} \cdot k}}{\alpha^3 \cdot \varepsilon^5}.$$

In order to complete the argument, it remains to establish Claim 4.27.

Proof of Claim 4.27. Let (A, B) be an $(\varepsilon/3, \varepsilon/3)$-good edge, and assume that

$$\Pr_{B'' \sim \mathcal{W}_I(A)}[B'' \text{ is } A\text{-heavy}] > \alpha/2.$$

We need to prove that

$$\Pr_{B' \sim N_{\mathcal{I}}(A)}[B' \text{ is } A\text{-heavy} \wedge p_{\mathsf{cons}}(B') \geq \varepsilon^*] > (\alpha/108) \cdot \varepsilon^3,$$

where $\varepsilon^* = (\alpha/8) \cdot (\varepsilon^2/9)$.

Note that the probability of interest can be rewritten as

$$\Pr_{B' \sim N_{\mathcal{I}}(A)}[B' \text{ is } A\text{-heavy} \mid p_{\mathsf{cons}}(B') \geq \varepsilon^*] \cdot \Pr_{B' \sim N_{\mathcal{I}}(A)}[p_{\mathsf{cons}}(B') \geq \varepsilon^*]. \tag{46}$$

Since $\varepsilon^* \leq \varepsilon/3$ and (A, B) is $(\varepsilon/3, \varepsilon/3)$-good, the rightmost probability is at least $\varepsilon/3$. We lower bound the other probability next.

Let $N_{\mathcal{I}}(A, \geq \varepsilon^*) = \{B' \in N_{\mathcal{I}}(A) \mid p_{\mathsf{cons}}(B') \geq \varepsilon^*\}$, and define $N_{\mathcal{I}}(A, < \varepsilon^*)$ in a similar way. On the one hand,

$$\begin{aligned} \Pr_{B' \sim N_{\mathcal{I}}(A)}[B' \text{ is } A\text{-heavy} \mid p_{\mathsf{cons}}(B') \geq \varepsilon^*] &= \Pr_{B' \sim N_{\mathcal{I}}(A, \geq \varepsilon^*)}[B' \text{ is } A\text{-heavy}] \\ &= \frac{1}{|N_{\mathcal{I}}(A, \geq \varepsilon^*)|} \cdot \sum_{B' \in N_{\mathcal{I}}(A, \geq \varepsilon^*)} \mathbf{1}_{[B' \text{ is } A\text{-heavy}]}. \end{aligned} \tag{47}$$

On the other hand, using the assumption of the claim,

$$\begin{aligned} \alpha/2 &< \frac{1}{p_{\mathsf{tot}}(A)} \left[\sum_{B' \in N_{\mathcal{I}}(A, \geq \varepsilon^*)} p_{\mathsf{cons}}(B') \cdot \mathbf{1}_{[B' \text{ is } A\text{-heavy}]} + \sum_{B' \in N_{\mathcal{I}}(A, < \varepsilon^*)} p_{\mathsf{cons}}(B') \cdot \mathbf{1}_{[B' \text{ is } A\text{-heavy}]} \right] \\ &\leq \frac{1}{p_{\mathsf{tot}}(A)} \left[\sum_{B' \in N_{\mathcal{I}}(A, \geq \varepsilon^*)} \mathbf{1}_{[B' \text{ is } A\text{-heavy}]} + \sum_{B' \in N_{\mathcal{I}}(A, < \varepsilon^*)} p_{\mathsf{cons}}(B') \right]. \end{aligned}$$

This yields

$$\sum_{B' \in N_{\mathcal{I}}(A, \geq \varepsilon^*)} \mathbf{1}_{[B' \text{ is } A\text{-heavy}]} > (\alpha/2) \cdot p_{\mathsf{tot}}(A) - \sum_{B' \in N_{\mathcal{I}}(A, < \varepsilon^*)} p_{\mathsf{cons}}(B').$$

In turn, thanks to our choice of ε^*,

$$\sum_{B' \in N_{\mathcal{I}}(A, < \varepsilon^*)} p_{\mathsf{cons}}(B') \leq |N_{\mathcal{I}}(A, < \varepsilon^*)| \cdot \varepsilon^* \leq |N_{\mathcal{I}}(A)| \cdot \frac{\alpha}{8} \cdot \frac{\varepsilon^2}{9} < \frac{\alpha}{4} \cdot p_{\mathsf{tot}}(A),$$

where the last inequality uses that (A, B) is $(\varepsilon/3, \varepsilon/3)$-good, which yields $p_{\mathsf{tot}}(A) \geq |N_{\mathcal{I}}(A)|\varepsilon^2/9$. As a consequence,

$$\sum_{B' \in N_{\mathcal{I}}(A, \geq \varepsilon^*)} \mathbf{1}_{[B' \text{ is } A\text{-heavy}]} > (\alpha/4) \cdot p_{\mathsf{tot}}(A) \geq \frac{\alpha}{4} \cdot \frac{\varepsilon^2}{9} \cdot |N_{\mathcal{I}}(A)|.$$

Overall, the probability we want to lower bound in Eq. (47) is at least

$$\frac{1}{|N_{\mathcal{I}}(A, \geq \varepsilon^*)|} \cdot \frac{\alpha}{4} \cdot \frac{\varepsilon^2}{9} \cdot |N_{\mathcal{I}}(A)| \geq \frac{\alpha}{4} \cdot \frac{\varepsilon^2}{9}.$$

Combining our estimates, Eq. (46) can be lower bound by $\frac{\alpha}{4} \cdot \frac{\varepsilon^2}{9} \cdot \varepsilon/3 = \alpha/108 \cdot \varepsilon^3$, which proves the claim. □

This completes the proof of Lemma 4.20. □

4.3.4 Extension to quantum circuits

The next statement shares part of the terminology from Theorem 4.8, and we refer to the beginning of Section 4.3 for more details. The part of the statement about uniformity refers to a fixed sequence of functions $g\colon \{0,1\}^n \to \{0,1\}$ indexed by n.

Theorem 4.28 (Local list decoding for quantum circuits)**.** *There exists a universal constant $C \geq 1$ for which the following holds. Let $n \geq 1$ be a positive integer, k be an even integer, and let $\varepsilon, \delta > 0$ satisfy*

$$k \;\geq\; C \cdot \frac{1}{\delta} \cdot \left[\log\left(\frac{1}{\delta}\right) + \log\left(\frac{1}{\varepsilon}\right)\right]. \tag{48}$$

If $\mathcal{G}$ is a quantum circuit of size at most s defined over $\mathcal{S}_{n,k}$ with k output bits such that

$$\mathop{\mathbb{E}}_{B \sim \mathcal{S}_{n,k},\, \mathcal{G}} \Big[\mathcal{G}(B) = g^k(B)\Big] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{B \sim \mathcal{S}_{n,k},\, \mathcal{G}} \Big[\|\Pi_{g^k(B)} \mathcal{G}\,|B, 0^m\rangle\|^2\Big] \;\geq\; \varepsilon, \tag{49}$$

then there is a quantum circuit $\mathcal{B}$ of size $\mathsf{poly}(n, k, s, \log(1/\delta), 1/\varepsilon)$ such that

$$\mathop{\mathbb{E}}_{x \sim \{0,1\}^n,\, \mathcal{B}} \Big[\mathcal{B}(x) = g(x)\Big] \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x \sim \{0,1\}^n,\, \mathcal{B}} \Big[\|\Pi_{g(x)} \mathcal{B}\,|x, 0^{m'}\rangle\|\Big] \;\geq\; 1 - \delta. \tag{50}$$

Moreover, a quantum circuit $\mathcal{B}$ of this form can be constructed with noticeable probability from a quantum circuit $\mathcal{G}$ as above by a uniform sequence of quantum circuits, a statement which is formalised as follows. For any choice of constructive functions $\varepsilon = \varepsilon(n)$, $\delta = \delta(n)$, and $k = k(n)$ satisfying the conditions of the theorem, and for any constructive function $s = s(n)$, there is a uniform family $\{\mathcal{C}_n^{\mathsf{IJKW}}\}_{n \geq 1}$ of quantum circuits $\mathcal{C}_n^{\mathsf{IJKW}}$ of size $\mathsf{poly}(n, k, s, \log(1/\delta), 1/\varepsilon)$ for which the following holds. If $\mathcal{C}_n^{\mathsf{IJKW}}$ is given as input a string $\mathsf{code}(\mathcal{G})$ describing a quantum circuit $\mathcal{G}$ with the properties above, then when its output is measured it produces with probability $\zeta = \Omega(\varepsilon^2)$ a string $\mathsf{code}(\mathcal{B})$ describing a quantum circuit $\mathcal{B}$ of the desired size and success probability.

Proof. The proof is divided into two parts: *existence* and *uniformity*. First, we argue that if there is quantum circuit $\mathcal{G}$ of the above form, then a corresponding quantum circuit $\mathcal{B}$ exists. Then, for a fixed choice of constructive functions ε, δ, and k that depend on n and satisfy the conditions of the result, and for any constructive function $s = s(n)$, we provide a deterministic algorithm D that, given 1^n, runs in time $t = \mathsf{poly}(n, k, s, \log(1/\delta), \varepsilon)$ and outputs the description of a quantum circuit $\mathcal{C}_n^{\mathsf{IJKW}}$ as in the statement.

Let $\mathcal{G}$ be any quantum circuit of size s satisfying the hypothesis of the theorem. Note that in the proof of Theorem 4.8 the corresponding circuit $\mathcal{G}$ is accessed in a classical way. By Lemma 2.17, the analysis of the success probability of the circuit $\mathcal{B}^{\mathcal{O}}$ constructed in the proof of Theorem 4.8 remains valid when the oracle $\mathcal{O}$ is replaced by the circuit $\mathcal{G}$. However, in the quantum case, instead of getting an inherently random *oracle* circuit $\mathcal{B}^{\mathcal{O}}$ of size $\mathsf{poly}(n, k, \log(1/\delta), 1/\varepsilon)$, we obtain a quantum circuit $\mathcal{B}$ of size $\mathsf{poly}(n, k, s, \log(1/\delta), 1/\varepsilon)$, where the size overhead comes from replacing each oracle gate $\mathcal{O}$ by a copy of the quantum circuit $\mathcal{G}$ of size s. Finally, recall that $\mathfrak{U} = \{0,1\}^n$, and note that the random input y of $\mathcal{B}^{\mathcal{O}}$ appearing in Equation 21 can be assumed to be part of the quantum computation of $\mathcal{B}$ by standard techniques. Consequently, we obtain a quantum circuit $\mathcal{B}$ for which Equation 50 holds.

Next, we argue that after fixing functions $s(n)$, $k(n)$, $\varepsilon(n)$, and $\delta(n)$, there is a quantum circuit $\mathcal{C}_n^{\mathsf{IJKW}}$ that given the *code* of a good quantum circuit $\mathcal{G}$ outputs with probability $\Omega(\varepsilon^2)$ the *code* of a quantum circuit $\mathcal{B}$ with the desired properties. The circuit $\mathcal{C}_n^{\mathsf{IJKW}}$ is simply the quantum analogue of the algorithm $\mathcal{D}$ from Construction 4.13. Let $T(n) = \mathsf{poly}(n, k, s, \log(1/\delta), 1/\varepsilon)$ be fixed as in the proof of Theorem 4.8. Then, given the code of $\mathcal{G}$ and using that n, k, and $T(n)$ are fixed, $\mathcal{C}_n^{\mathsf{IJKW}}$ proceed as follows. It uses its internal randomness (simulated in a quantum way) to compute as $\mathcal{D}$ in Step 1, then it simulates $\mathcal{G}$ using a *universal quantum circuit* (see Section 2.2) of size $\mathsf{poly}(T)$ in order to sample $w \sim \mathcal{G}(B)|_A$,[13] and finally it outputs the *description* of a quantum circuit $\mathcal{B}$

[13] This is where we use that all parameters are fixed, meaning that we can instantiate a universal quantum circuit for quantum computations containing a fixed number of gates.

that computes as $C_{A,w}$, where $C_{A,w}$ incorporates the code of $\mathcal{G}$. Since by the proof of Theorem 4.8 algorithm $\mathcal{D}$ outputs a good circuit $\mathcal{B}^{\mathcal{O}}$ with probability $\Omega(\varepsilon^2)$ over its internal randomness and the randomness of $\mathcal{G}$ (whenever $\mathcal{G}$ satisfies the conditions of the theorem), it follows that this is also true for $\mathcal{C}_n^{\mathsf{IJKW}}$ and the description $\mathsf{code}(\mathcal{B})$ that it generates from $\mathsf{code}(\mathcal{G})$.

Observe that $\mathcal{C}_n^{\mathsf{IJKW}}$ is defined over inputs of length $\mathsf{poly}(s)$, which represent the string $\mathsf{code}(\mathcal{G})$. In addition, $\mathcal{C}_n^{\mathsf{IJKW}}$ has size $\mathsf{poly}(T)$. This includes the time it takes to simulate $\mathcal{G}$, and the time it requires to print an explicit description of $\mathcal{B}$, which contains $\mathsf{poly}(T)$ many gates.

Finally, note that the *code* of the quantum circuit $\mathcal{C}_n^{\mathsf{IJKW}}$ is fully explicit, given a choice of parameters. In other words, there is a *deterministic* algorithm that when given 1^n runs for at most $t = \mathsf{poly}(T) = \mathsf{poly}(n, k, s, \log(1/\delta), 1/\varepsilon)$ steps and prints $\mathcal{C}_n^{\mathsf{IJKW}}$. □

4.4 Self-reducibility in the quantum setting

In this section, we explain how to use the downward and random self-reducibility of a language L to help us to produce a sequence of circuits computing L. In more detail, we show how to design a small quantum circuit B_n to compute L on n bit inputs from a large quantum circuit P_{n-1} computing L on $n-1$ bit inputs and a collection of small quantum circuits $A_1, \ldots, A_t$ with the following guarantee: some A_i offers a good approximation of L over n bit inputs.

We will implement this idea with respect to the language $L^\star$ provided by Theorem 2.16. Note that the downward self-reducibility and random-self-reducibility of this language holds with respect to *classical* computation, while here we will rely on these structural properties in the context of *quantum* circuits. This is not an issue for the following reasons.

In the case of downward self-reducibility, given a quantum circuit P_{n-1} for $L^\star_{n-1}$ (i.e. $L^\star$ restricted to $n-1$ bit inputs), we observe that its success probability on every input can be amplified to $1 - \mathsf{negl}(n)$. As a consequence, the classical reduction, which makes $\mathsf{poly}(n)$ classical queries, will obtain correct and consistent answers with overwhelming probability, even if P_{n-1} is a quantum circuit.

On the other hand, in the case of random self-reducibility, on an input x for $L^\star$, the reduction is implemented by a classical algorithm that makes n^a queries to a classical oracle A, where each query is uniformly distributed over $\{0,1\}^n$ (but different queries can be correlated). If the oracle A answers all queries according to a fixed function $\widetilde{f}_n \colon \{0,1\}^n \to \{0,1\}$ that is $1/n^b$-close to $L^\star_n$, the reduction gives a correct answer on x with high probability. Now note that Definition 2.15 does not offer a guarantee when $\widetilde{f}_n$ is not a classical oracle, e.g., if it is defined from a quantum circuit that does not provide a deterministic output. Fortunately, in order to quantize this reduction, we can reduce the analysis to the classical case. This is only possible because here we are in the regime where the correlation of interest is of the form $1 - o(1)$, while for instance in hardness amplification (Section 4.3) we consider correlations that are $o(1)$.

In more detail, here is one possible way of doing this. In our proof, A is some quantum circuit that computes $L^\star_n$ with probability p, in the sense that the expected agreement between A and $L^n_\star$ on a random input x and the measurement of A's ouput is p. It turns out that in our analysis we can take $p \geq 1 - 1/n^c$ for a convenient constant $c > a + b$. This allows us to show that there is a large set $S \subseteq \{0,1\}^n$ with $\Pr_x[x \in S] \geq 1 - 1/n^{a+b+1}$ such that A is correct on each $x \in S$ with high probability. Moreover, by amplifying the success probability of A, we can get a quantum circuit A' that is correct on each $x \in S$ with probability at least, say, $1 - 2^{-n}$. Finally, since the reduction makes at most n^a queries and each of them is uniform, by a union bound, with high probability all queries land in S. This can be used to show that with high probability all (classical) queries answered by the quantum circuit A' agree with a classical function that is $1/n^b$-close to $L^\star_n$, which guarantees the correctness of the reduction also in our context. Using the ideas described above,

it is not hard to implement the result explained at the beginning of this section. We provide the details next.

We start with the following: suppose we have a quantum circuit U that computes a random self-reducible language L with high probability for a uniformly random input, then one can construct a quantum circuit U^* that, with high probability, computes L on every $x \in \{0,1\}^n$.

Lemma 4.29 (Random self-reducibility and quantum circuits)**.** *Let $L : \{0,1\}^* \to \{0,1\}$ be a random self-reducible language with parameters a, b, c (as described in Definition 2.15). For every n, suppose we have the description of a quantum circuit U such that*

$$\mathop{\mathbb{E}}_{x\in\{0,1\}^n}\left[\|\Pi_{L(x)}U\left|x,0^q\right\rangle\|^2\right] \geq 1 - \frac{1}{n^k}, \tag{51}$$

for some $k \geq 2b + a$.

There is a $O(|U| \cdot \mathrm{poly}(n))$-size quantum circuit U^* that satisfies

$$\|\tilde{\Pi}_x U^* \left|0, x, 0^{q^*}\right\rangle\|^2 \geq 1 - 2^{-2n+1} \qquad \text{for every } x \in \{0,1\}^n,$$

where $\tilde{\Pi}_x = |L(x)\rangle\langle L(x)| \otimes |x\rangle\langle x| \otimes |0^{q^*}\rangle\langle 0^{q^*}|$ and $q^* = \mathrm{poly}(n)$.

Proof. Let us define $Y = \{x \in \{0,1\}^n : \|\Pi_{L(x)}U\left|x,0^q\right\rangle\|^2 \geq \frac{2}{3}\}$. It follows from Equation (51), that $|Y| \geq (1 - \frac{3}{n^k})2^n$. In this case, we can consider the circuit U^{amp} that, on input x, computes U on the input $O(n)$ times in parallel and answers with the majority of the outputs. It follows that in this case, for every $x \in Y$, we have that $\|\Pi_{L(x)}U^{\mathrm{amp}}\left|x,0^{O(qn)}\right\rangle\|^2 \geq 1 - 2^{-n}$.

Let $g : \{0,1\}^* \to \{0,1\}^*$ be the polynomial-time computable *classical* function from Definition 2.15 in the definition of the random self-reducibility of L. Before defining U^*, we first define the unitary $\tilde{U}_1$

$$\widetilde{U}_1 : |x,0\rangle \to \frac{1}{\sqrt{2^{n^c}}} \underbrace{\sum_{r\in\{0,1\}^{n^c}} |x,r,0\rangle \bigotimes_{i\in[n^a]} |g(i,x,r),0\rangle}_{:=|\chi\rangle}. \tag{52}$$

We now use the circuit U^{amp} to compute L for every $g(i,x,r)$, i.e., we apply $\widetilde{U}_2 = \mathsf{Id}\otimes(U^{\mathrm{amp}})^{\otimes n^a}$ to the state $|\chi\rangle$ to obtain

$$|\psi\rangle = \frac{1}{\sqrt{2^{n^c}}} \sum_{r\in\{0,1\}^{n^c}} |x,r,0\rangle \bigotimes_{i\in[n^a]} \left(U^{\mathrm{amp}}\left|g(i,x,r),0\right\rangle\right). \tag{53}$$

Let us fix some random r. Observe that by our assumption on g (i.e. $g(i,x,r) \sim \mathcal{U}_n$), it follows from a union bound that there exists some $i \in [n^a]$ such that $g(i,x,r) \notin Y$ with probability at most $\frac{3}{n^{k-a}}$. Assuming that for every $i \in [n^a]$ we have $g(i,x,r) \in Y$, it follows that for every $i \in [n^a]$

$$\Pr_r\left[\left\|\Pi_{L(g(i,x,r))}U\left|g(i,x,r),0\right\rangle\right\|^2\right] \geq 1 - 2^{-n}.$$

Let $\widetilde{U}_3$ be a unitary that implements the classical circuit $h : \{0,1\}^* \to \{0,1\}$ from Definition 2.15. The action of $\widetilde{U}_3$ on $|\psi\rangle$ can be written as

$$|\phi\rangle = \frac{1}{\sqrt{2^{n^c}}} \sum_r \tilde{U}_3 \left(|x,r,0\rangle \bigotimes_{i\in[n^a]} U^{\mathrm{amp}}\left|g(i,x,r),0\right\rangle\right). \tag{54}$$

Notice that if we assume that for every $i \in [n^a]$ we have $g(i,x,r) \in Y$, and that U^{amp} is correct for every $g(i,x,r)$, we have from Definition 2.15 that

$$\|\Pi_{L(x)} |\phi\rangle\|^2 \geq 1 - 2^{-2n}.$$

It follows from a union bound that for every $x \in \{0,1\}^n$,

$$\|\Pi_{L(x)} |\phi\rangle\|^2 \geq 1 - \frac{3}{n^{k-a}} - \frac{n^a}{2^n} - \frac{1}{2^{2n}} \geq 1 - \frac{1}{\text{poly}(n)}.$$

We can pick U^* as the algorithm that runs $\tilde{U}_3\tilde{U}_2\tilde{U}_1$ in parallel $O(n)$ times and answers with the majority. Finally, to remove any garbage from the computation, we can copy the output register into a separate register and uncompute U^* and still compute $L(x)$ with overwhelming probability. □

We now show that if L is downward self-reducible, then we can construct a quantum circuit U^* that computes L on inputs of size n from a quantum circuit U_{n-1} that computes L on inputs with size $n-1$.

Theorem 4.30 (Downward-self-reducibility of $L^\star$ and quantum circuits)**.** *Let $s_P \colon \mathbb{N} \to \mathbb{N}$ be a constructive function. Let $L^\star$ be the language from Theorem 2.16. There is a sequence $\{\mathcal{C}_n^{\mathsf{DR}}\}_{n\geq 1}$ of deterministic circuits $\mathcal{C}_n^{\mathsf{DR}}$ for which the following holds:*

- *(i)* Input: *Each circuit $\mathcal{C}_n^{\mathsf{DR}}$ gets as input 1^n and a string $\mathsf{code}(P_{n-1})$ that describes a quantum circuit P_{n-1} of size $\leq s_P(n-1)$.*
- *(ii)* Uniformity and Size: *Each circuit $\mathcal{C}_n^{\mathsf{DR}}$ is of size $S(n) = \mathsf{poly}(n, s_P(n-1))$, and there is a deterministic algorithm that when given 1^n prints $\mathsf{code}(\mathcal{C}_n^{\mathsf{DR}})$ in time $\mathsf{poly}(S(n))$.*
- *(iii)* Output and Correctness: *If P_{n-1} computes $L^\star$ on inputs of length $n-1$ then $\mathcal{C}_n^{\mathsf{DR}}$ outputs the description of a quantum circuit P_n of size $\mathsf{poly}(n, s_P(n-1))$ that computes $L^\star$ on inputs of length n.*

Proof. By item (i), we have that

$$\left\| \Pi_{L^\star(x)} P_{n-1} |x, 0^q\rangle \right\|^2 \geq 2/3.$$

We first amplify the success probability of P_{n-1} from $2/3$ to $1 - \mathsf{negl}(n)$. For this, we perform a standard majority vote on the outputs of $O(n)$ parallel copies of P_{n-1} and construct P^*_{n-1} of size $O(n \cdot s_P(n-1))$ such that

$$\left\| \widetilde{\Pi}_x P^*_{n-1} |0, x, 0^q\rangle \right\|^2 \geq 1 - \frac{1}{\mathsf{negl}(n)}, \tag{55}$$

where $q = O(\text{poly}(n))$ and $\widetilde{\Pi}_x = |L^\star(x)\rangle\langle L^\star(x)| \otimes |x\rangle\langle x| \otimes |0^q\rangle\langle 0^q|$. Now, P_n simulates the *classical* circuit $A^{L^\star}$ from Definition 2.14 that computes $L^\star(x)$ and answers A's queries to $L^\star$ on instances of size $n-1$ by simulating P^*_{n-1}. To remove garbage, P_n copies the output of $A^{L^\star}$ onto the output qubit, and then *uncomputes* $A^{L^\star}$ (by also uncomputing the calls to P^*_{n-1}). Since each one of the polynomially many queries is correct with probability at least $1 - 1/\mathsf{negl}(n)$, we have that the output of $A^{L^\star}$ is correct with probability at least $1 - 1/\mathsf{negl}(n) \geq 2/3$. Additionally, as A is a $\text{poly}(n)$ time circuit and P^*_{n-1} is of size $O(n \cdot s_P(n-1))$, P_n is of size $\text{poly}(n, s_P(n-1))$, and therefore (iii) holds.

In order to show (ii), observe that the circuit $\mathcal{C}_n^{\mathsf{DR}}$ first generates $\mathsf{code}(P^*_{n-1})$ of size $O(n \cdot s_P(n-1))$ by using $\mathsf{code}(P_{n-1})$. It then converts the classical circuit $A^{L^\star}$ into it's reversible form and replaces these reversible gates with corresponding unitary descriptions. It also replaces the queries made by $A^{L^\star}$ with $\mathsf{code}(P^*_{n-1})$. All of this can be computed by $\mathcal{C}_n^{\mathsf{DR}}$ using $\text{poly}(n, |\mathsf{code}(P_{n-1})|) = \text{poly}(n, s_P(n-1))$ gates. □

Theorem 4.31 (Self-reducibility of $L^\star$ and quantum circuits)**.** *Let $s_A, s_P, t\colon \mathbb{N} \to \mathbb{N}$ be constructive functions. Moreover, let $L^\star$ be the language from Theorem 2.16, and $a_\star, b_\star$ be the associated constants. There is a sequence $\{\mathcal{C}_n^{\mathsf{SR}}\}_{n\geq 1}$ of quantum circuits $\mathcal{C}_n^{\mathsf{SR}}$ for which the following holds:*

(*i*) Input: *Each circuit $\mathcal{C}_n^{\mathsf{SR}}$ gets as input 1^n and strings $\mathsf{code}(P_{n-1})$, $\mathsf{code}(A_1), \ldots, \mathsf{code}(A_{t(n)})$, where P_{n-1} is a quantum circuit of size $\leq s_P(n-1)$ and each A_i is a quantum circuit of size $\leq s_A(n)$.*

(*ii*) Uniformity and Size: *Each circuit $\mathcal{C}_n^{\mathsf{SR}}$ is of size $S(n) = \mathsf{poly}(n, t(n), s_A(n), s_P(n-1))$, and there is a deterministic algorithm that when given 1^n prints $\mathsf{code}(\mathcal{C}_n^{\mathsf{SR}})$ in time $\mathsf{poly}(S(n))$.*

(*iii*) Output and Correctness: *Assume that P_{n-1} computes $L^\star$ on inputs of length $n-1$, and that there exists $i \in [t(n)]$ such that*

$$\Pr_{x\sim\{0,1\}^n,\, A_i}[A_i(x) = L^\star(x)] \;\geq\; 1 - n^{-2b_\star - a_\star}. \tag{56}$$

Then with probability at least $1 - 1/500n^2$ over its output measurement, $\mathcal{C}_n^{\mathsf{SR}}$ generates the description $\mathsf{code}(B_n)$ of a quantum circuit B_n of size $\mathsf{poly}(n, s_A(n))$ that correctly computes $L^\star$ on inputs of length n. In other words, for every $x \in \{0,1\}^n$,

$$\Pr_{B_n}[B_n(x) = L^\star(x)] \geq 2/3.$$

Note. Notice that from Theorem 4.30, we could achieve a circuit of size $\mathrm{poly}(n, s_P(n-1))$ that computes L^* on inputs of size n. The non-trivial aspect of Theorem 4.31 is to achieve a circuit P_n whose size depends only on n and $s_A(n)$ and is *independent* of $s_P(n-1)$.

Proof. First, given $\mathsf{code}(P_{n-1})$, we use Theorem 4.30 to construct a circuit P_n with $|\mathsf{code}(P_n)| = \mathrm{poly}(n, s_P(n-1))$ such that for every $x \in \{0,1\}^n$ and for $q' = \mathrm{poly}(n)$, we have

$$\left\| \widetilde{\Pi}_x P_n \left|0, x, 0^{q'}\right\rangle \right\|^2 \geq 1 - \mathsf{negl}(n) \tag{57}$$

where $\widetilde{\Pi}_x = |L^\star(x)\rangle\langle L^\star(x)| \otimes |x\rangle\langle x| \otimes |0^{q'}\rangle\langle 0^{q'}|$. In the remainder of this proof, the usage of $\widetilde{\Pi}_x$ will denote the process of checking if the output qubit is $|L^\star(x)\rangle$, the input qubits remain as $|x\rangle$ and all auxiliary qubits are set to 0. We proceed to construct B_n in three steps using the random and downward self-reducibility of $L^\star$.

Step 1: Notice that, the theorem statement provides no guarantees on how well A_ℓ performs for $\ell \neq i$. To identify the circuits (among $\{A_1, \ldots, A_{t(n)}\}$) which compute $L^\star$ correctly with probability at least $1 - 1/\mathrm{poly}(n)$, under the promise that there exists one, we carry out the following test. Let $R = O(\log(t(n)/\eta))$ where $\eta = 1/\mathrm{poly}(n)$. For every $\ell \in [t(n)]$, pick uniformly random $x_\ell^1, \ldots, x_\ell^R \in \{0,1\}^n$, and for every $r \in [R]$ run A_ℓ and P_n (constructed at the start of the proof) separately on two copies of $|0, x_\ell^r, 0^{q'}\rangle$, measure the first qubit and let the outputs be $b_\ell^r, c_\ell^r \in \{0,1\}$ respectively. Consider the set

$$\mathcal{J} = \left\{ \ell \in [t(n)] : \sum_{r=1}^{R} [b_\ell^r = c_\ell^r] \geq \frac{3R}{4} \right\}.$$

Note that it is possible that there is an $\ell \in \mathcal{J}$ that passes the test but A_ℓ does not compute $L^\star$ correctly. However, it suffices for us to show that, with high probability (over the randomness of

sampling x_j^is) if $\ell \in \mathcal{J}$, then the quantum circuit A_ℓ computes $L^\star$. Towards this end, we define $t_\star = 2b_\star + a_\star + 1$ and show that, supposing $\Pr[A_\ell(x) = P_n(x)] < 1 - n^{-t_\star}$ (where the probability is taken over uniformly random $x \in \{0,1\}^n$ and randomness in A_ℓ, P_n) then with high probability $\ell \notin \mathcal{J}$. To prove this, first notice that if $\Pr[A_\ell(x) = P_n(x)] \leq 1 - n^{-t_\star}$, then applying the Chernoff bound (Theorem 2.1) with $\mu < 1 - n^{-t_\star}$ and $\delta = O(1)$, we have

$$\Pr\left[\frac{1}{R}\sum_r [b_\ell^r = c_\ell^r] \geq \frac{3}{4}\right] \leq e^{-O(R)},$$

This implies that with probability $\geq 1 - e^{-O(R)}$ (over the random samples x_j^is), if $\ell \in \mathcal{J}$, then we have $\Pr_x[A_\ell(x) = P_n(x)] \geq 1 - 1/n^{t_\star}$. Moreover, along with Equation (57), this implies that with probability $\geq 1 - e^{-O(R)}$, if $\ell \in \mathcal{J}$ then

$$\begin{aligned}\Pr_x[A_\ell(x) = L^\star(x)] &\geq \Pr_x[A_\ell(x) = L^\star(x) | P_n(x) = L^\star(x)] \cdot \Pr_x[P_n(x) = L^\star(x)] \\ &\geq (1 - n^{-t_\star})\,(1 - \mathsf{negl}(n)) \geq 1 - n^{-2b_\star - a_\star}\end{aligned}$$

where the second inequality uses the fact that the conditional probability refers to the event that $A_\ell(x) = P_n(x)$. Hence, with probability at least $1 - e^{-O(R)}$, if $\ell \in \mathcal{J}$ then $\Pr[A_\ell(x) = L^\star(x)] \geq 1 - n^{-2b_\star - a_\star}$. By taking a union bound over all $\ell \in \mathcal{J}$, we now have

$$\Pr\left[\exists \ell \in \mathcal{J} : \Pr[A_\ell(x) = L^\star(x)] < 1 - n^{-2b_\star - a_\star}\right] \leq t(n) \cdot e^{-O(R)} \leq \eta.$$

Hence with probability at least $1 - \eta$, every $\ell \in \mathcal{J}$ satisfies $\Pr[A_\ell(x) = L^\star(x)] \geq 1 - n^{-2b_\star - a_\star}$.

Before proceeding to the next step, we argue that with probability $\geq 1 - e^{-O(R)}$, $\mathcal{J}$ is non-empty. By assumption, we have that for $i \in [t(n)]$ it follows that $\Pr_x[A_i(x) = L^\star(x)] \geq (1 - n^{-2b_\star - a_\star})$. Notice that since $\Pr_x[P_n(x) = L^\star(x)] \geq 1 - \mathsf{negl}(n)$, we have that $\Pr_{x_1,\ldots,x_R}[\exists r \in [R], c_i^r \neq L^\star(x_i^r)] \leq \mathsf{negl}(n)$ by a union bound. Conditioned that for all r we have $c_i^r = L^\star(x_i^r)$, we have that the probability that A_i fails the test is very small i.e.,

$$\Pr\left[\frac{1}{R}\sum_r [b_i^r = c_i^r] \leq \frac{3}{4}\right] = \Pr\left[\frac{1}{R}\sum_r [b_i^r = L^\star(x_i^r)] \leq \frac{3}{4}\right] \leq e^{-O(R)},$$

where the inequality follows by applying the Chernoff bound (Theorem 2.1) on Eq. (56), Hence, with probability at least $1 - e^{-O(R)} - \mathsf{negl}(n) = 1 - e^{-O(R)}$, $\mathcal{J}$ contains i.

Step 2: In Step 1, we showed that circuits A_ℓ for $\ell \in \mathcal{J}$ compute $L^\star$ on *average* x. Now we use the random self-reducibility property of $L^\star$ to obtain a circuit that performs well for *every* $x \in \{0,1\}^n$ and not just for a uniformly random x. In this direction, for every $\ell \in \mathcal{J}$, given $\mathsf{code}(A_\ell)$, we use Lemma 4.29 to construct the circuit A_ℓ^* (with $|\mathsf{code}(A_\ell^*)| = \mathrm{poly}(|\mathsf{code}(A_\ell)|, n) = \mathrm{poly}(s_A(n), n)$). Then, for every $\ell \in \mathcal{J}$ such that $\mathbb{E}_x[A_\ell(x) = L^\star(x)] \geq 1 - n^{-2b_\star - a_\star}$ we also have

$$\left\|\widetilde{\Pi}_x A_\ell^* \left|0, x, 0^{\bar{q}}\right\rangle\right\|^2 \geq 1 - 2^{-2n+1}, \quad \text{for every } x \in \{0,1\}^n. \tag{58}$$

We now prove item (iii). Pick $\eta = 1/500n^2$ and note that with probability at least $1 - \eta = 1 - 1/500n^2$, every $\ell \in \mathcal{J}$ satisfies Eq. (58). Hence, picking an arbitrary $\ell \in \mathcal{J}$ and setting $B_n = A_\ell^*$ gives us the desired quantum circuit with $|\mathsf{code}(B_n)| = |\mathsf{code}(A_\ell^*)| \leq \mathrm{poly}(s_A(n), n)$.

Finally, observe that Step 1 can be described with $\mathrm{poly}(t(n) \cdot |\mathsf{code}(A_\ell^*)| + |\mathsf{code}(P_n)|) = \mathrm{poly}(n, t(n), s_P(n-1), s_A(n))$ gates. Similarly, Step 2 can be described with $\mathrm{poly}(T, t(n), s_P(n-1), s_A(n)) = \mathrm{poly}(n, t(n), s_P(n-1), s_A(n))$ gates. Step 3 uses $\mathrm{poly}(n, s_A(n))$ gates. Putting these together, $|\mathsf{code}(\mathcal{C}_n^{\mathsf{SR}})| = \mathrm{poly}(n, t(n), s_P(n-1), s_A(n))$. This proves item (ii) and completes the proof of the theorem. □

5 A conditional PRG against uniform quantum circuits

In this section, we put together the results of Section 4, and show that if polynomial-space classical algorithms cannot be simulated in sub-exponential time by quantum algorithms, then there exists a pseudorandom generator secure against uniform quantum computations.

Theorem 5.1 (Conditional PRG against uniform quantum computations)**.** *Suppose that* $\mathsf{PSPACE} \not\subseteq \mathsf{BQSUBEXP}$. *In other words, there is a language* $L \in \mathsf{PSPACE}$ *and* $\gamma > 0$ *such that* $L \notin \mathsf{BQTIME}[2^{n^\gamma}]$. *Then, for some choice of constants* $\alpha \geq 1$ *and* $\lambda \in (0, 1/5)$, *there is an infinitely often* $(\ell, m, s, \varepsilon)$*-generator* $G = \{G_n\}_{n\geq 1}$, *where* $\ell(n) \leq n^\alpha$, $m(n) = \lfloor 2^{n^\lambda} \rfloor$, $s(m) = 2^{n^{2\lambda}} \geq \mathsf{poly}(m)$ (*for any polynomial*), *and* $\varepsilon(m) = 1/m$.

In the proof given below, one can even take a larger constant λ closer to 1. However, since we are not optimizing the choice of the constant α in the seed length, this is inessential.

Proof. Let $L^\star \subseteq \{0,1\}^*$ be the special language from Theorem 2.16. For each $n \geq 1$, let $f_n \colon \{0,1\}^n \to \{0,1\}$ be the indicator Boolean function which agrees with the set $L^\star \cap \{0,1\}^n$. We use these functions to define a set of candidate generators $\{G^{\alpha,\lambda}\}_{\alpha,\lambda}$, where $G^{\alpha,\lambda} = \{G_n^{\alpha,\lambda}\}_{n\geq 1}$ with parameters ℓ, m, s, and ε as in the statement of the result. We then argue that if none of them is an infinitely often $(\ell, m, s, \varepsilon)$-generator, then for every $\gamma > 0$ we have $L^\star \in \mathsf{BQTIME}[2^{n^\gamma}]$. Since this language is complete for PSPACE, as a consequence we get $\mathsf{PSPACE} \subseteq \mathsf{BQSUBEXP}$, contradicting the hypothesis of the theorem.

Each generator $G^{\alpha,\lambda} = \{G_n^{\alpha,\lambda}\}_{n\geq 1}$ for a large enough $\alpha \geq 1$ is defined as follows. Let $d_\star \geq 1$ be the constant appearing in Theorem 2.16, and $a_\star, b_\star, c_\star \geq 1$ be the constants from Definition 2.15 associated with the random-self-reducibility of $L^\star$. Next, for every $n \geq 1$ we set

$$n_1(n) \stackrel{\text{def}}{=} k{\cdot}n \ \text{ (for } k(n) \stackrel{\text{def}}{=} 2n^{2b_\star+a_\star+d_\star+2}\text{)}, \quad n_2(n) \stackrel{\text{def}}{=} kn{+}k, \quad \ell(n) \stackrel{\text{def}}{=} (kn{+}k)^2 \leq n^\alpha, \quad m(n) \stackrel{\text{def}}{=} \lfloor 2^{n^\lambda} \rfloor,$$

and introduce functions

$$g_n \colon \{0,1\}^{n_1} \to \{0,1\}^k, \quad h_n \colon \{0,1\}^{n_2} \to \{0,1\}, \quad G_n^{\alpha,\lambda} \colon \{0,1\}^\ell \to \{0,1\}^m,$$

which are defined as follows:

$$\begin{aligned} g_n \quad &\equiv \quad f_n^k, \text{ i.e., } g(x^1, \ldots, x^k) \stackrel{\text{def}}{=} (f_n(x^1), \ldots, f_n(x^k)) \text{ for every } \vec{x} \in \{0,1\}^{kn}, \\ h_n(\vec{x}, r) \quad &\stackrel{\text{def}}{=} \quad g_n(\vec{x}) \cdot r = \sum_{i=1}^{k} f(x^i) \cdot r \ \ (\mathsf{mod}\ 2), \text{ where } r \in \{0,1\}^k, \text{ and} \\ G_n^{\alpha,\lambda}(z) \quad &\stackrel{\text{def}}{=} \quad \mathsf{NW}^{h_n}(z) \text{ instantiated with } m(n) \text{ output bits.} \end{aligned}$$

This completes the definition of the generator $G^{\alpha,\lambda} \stackrel{\text{def}}{=} \{G_n^{\alpha,\lambda}\}_{n\geq 1}$.

We argue next that if $G^{\alpha,\lambda}$ is *not* an infinitely often $(\ell, m, s, \varepsilon)$-generator, then $L^\star \in \mathsf{BQTIME}[2^{n^{3\lambda}}]$. First, note that it has the correct *stretch*. Moreover, f_n can be computed in (deterministic) time $O(2^{n^{d_\star}})$, g_n and h_n can be computed in time $\mathsf{poly}(n) \cdot 2^{n^{d_\star}}$, and the sets in the combinatorial design can each be computed in time $\mathsf{poly}(n)$. Given these time bounds, it follows that $G_n^{\alpha,\lambda}$ can be computed in time $m(n) \cdot \mathsf{poly}(n) \cdot 2^{n^{d_\star}} = O(2^{\ell(n)})$, by our choice of $\ell(n)$. In other words, the generator also satisfies the *uniformity and running time* requirements. Thus, if $G^{\alpha,\lambda}$ is not an infinitely often $(\ell, m, s, \varepsilon)$-generator it must be the case that its output distributions violate the *pseudorandomness* condition. We explore this in what follows.

Let $\mathcal{D}_m \equiv G_n^{\alpha,\lambda}(\mathcal{U}_\ell)$ be the distribution induced by the output of the generator on a random input seed. Since the generator is *not* infinitely often pseudorandom for parameters $s(m) = 2^{n^{2\lambda}}$ and $\varepsilon(m) = 1/m$, there is a deterministic algorithm $A(1^m)$ that runs in time $s(m)$, outputs a "distinguisher" quantum circuit D_m over m input variables and of size at most $s(m)$, and for *every large enough* n (say, $n \geq \kappa \in \mathbb{N}$),

$$\left| \Pr_{x \sim \{0,1\}^m} [D_m(x) = 1] - \Pr_{y \sim \mathcal{D}_m} [D_m(y) = 1] \right| > \varepsilon(m).$$

Our next step is to argue that algorithm A can be used to define a *uniform* family of quantum circuits Q_n of size at most $O(2^{n^{3\lambda}})$ that correctly decide $L^\star$ on n-bit inputs. In other words, according to our notation, for every $x \in \{0,1\}^n$ we have $\Pr_{Q_n}[Q_n(x) = f_n(x)] \geq 2/3$. By the discussion above, this completes the proof of the theorem.

The quantum circuit $Q_n(x)$. We now describe each quantum circuit Q_n and argue about its correctness. This circuit computes in $1 + (n - \kappa + 1) + 1$ sequential stages, which are delegated to sub-circuits Q'_0 followed by $Q'_\kappa, Q'_{\kappa+1}, \ldots, Q'_n$ and a final sub-circuit E_n:

Initialization: Q'_0 on input $|0^{q_0}\rangle$ prints the description $\mathsf{code}(P_{\kappa-1})$ of a quantum circuit $P_{\kappa-1}$ of size $O(1)$ that computes the Boolean function $f_{\kappa-1}$ corresponding to $L^\star$ over $(\kappa-1)$-bit strings.

Core Stages: For every $j \in \{\kappa, \ldots, n\}$, the circuit Q'_j expects as input a description $\mathsf{code}(P_{j-1})$ of a quantum circuit P_{j-1} that computes f_{j-1}. The goal of Q'_j is to output with high probability the code of a circuit P_j that computes f_j.

Final Computation: Quantum circuit E_n expects input strings $\mathsf{code}(P_n)$ and $x \in \{0,1\}^n$, and outputs the simulation of P_n on x.

(We omit in this description the amplification of the success probability of Q_n from $1/2 + \Omega(1)$ to $\geq 2/3$ (see e.g. Section 4.4, where more elaborate amplifications are discussed.)

Next, we formalise this idea, analysing the size and uniformity of quantum circuits Q'_j and E_n, the size of the involved circuits P_j, and the success probability of each Q'_j. This will allow us to bound the size of Q_n and to show that it correctly computes f_n.

In order to define these circuits, we will make use of the uniform families of quantum circuits from Section 4 and of the uniform family of quantum circuits $\{D_{m(n)}\}_{n \geq 1}$ that violate the pseudorandomness of the generator whenever $n \geq \kappa$. Our main goal is to prove the following lemma.

Lemma 5.2. *There exist universal constants $C_U \geq C_Q \geq C_P \geq 1$ for which the following holds. Let $s_P(n) \stackrel{\text{def}}{=} 2^{C_P \cdot n^{2\lambda}}$ for every $n \geq 1$. For every $j \in \{\kappa, \ldots, n\}$, there is a quantum circuit Q'_j such that:*

(*i*) Input: *Q'_j expects as input the description of a quantum circuit P_{j-1} of size $\leq s_P(j-1)$.*

(*ii*) Size and Uniformity: *Q'_j is a circuit of size $\leq 2^{C_Q \cdot j^{2\lambda}}$. Moreover, there is a deterministic algorithm that when given 1^j prints $\mathsf{code}(Q'_j)$ in time $\leq 2^{C_U \cdot j^{2\lambda}}$.*

(*iii*) Output and Correctness: *If the input circuit P_{j-1} correctly computes f_{j-1}, then with probability at least $1 - 1/100j^2$ over its output measurement, Q'_j generates the description of a circuit P_j of size $\leq s_P(j)$ that computes f_j.*

Assuming Lemma 5.2, we can complete the proof of Theorem 5.1 as follows. By the definition of Q_n and its components, it follows from a union bound over all measurements in the core stages of Q_n that the probability that the string $\mathsf{code}(P_n)$ output by Q'_n does not describe a quantum circuit P_n that computes f_n is a most

$$\sum_{j=\kappa}^{n} 1/100j^2 \;\leq\; \frac{1}{100} \cdot \sum_{j \geq 1} \frac{1}{j^2} \;=\; \frac{1}{100} \cdot \frac{\pi^2}{6} \;\leq\; \frac{1}{50}.$$

In this case, for every fixed $x \in \{0,1\}^n$, $\Pr_{P_n}[P_n(x) = f_n(x)] \geq 2/3$. Since in the final computation stage of Q_n it uses E_n to simulate the input circuit P_n on x, we get by a union bound that on each input x,

$$\Pr_{Q_n}[Q_n(x) \neq f_n(x)] \;\leq\; 1/50 + 1/3 \;<\; 2/5.$$

In other words, Q_n computes f_n. The size of Q_n is given by the sum of the sizes of each component. First, we can assume that $\mathsf{size}(Q'_0) \leq 2^{C_P \cdot (\kappa-1)^{2\lambda}}$ provided that C_P is a large enough constant, given that κ is constant. In addition, we have

$$\sum_{j=\kappa}^{n} \mathsf{size}(Q'_j) \;\leq\; n \cdot \mathsf{size}(Q'_n) \;\leq\; n \cdot 2^{C_Q \cdot n^{2\lambda}}.$$

Finally, the size of E_n is at most polynomial in the size of the input circuit P_n. Overall, we get that $\mathsf{size}(Q_n) = 2^{O(n^{2\lambda})} = O(2^{n^{3\lambda}})$, as desired. The uniformity of Q_n follows from the uniformity of its components (the code of Q'_0 can be obtained using an exhaustive computation, since κ is constant.). It follows from this discussion that $L^\star \in \mathsf{BQTIME}[2^{n^{3\lambda}}]$.

We now proceed to prove Lemma 5.2, which finishes our proof.

Proof of Lemma 5.2. It will be evident from our proof that large enough constants C_P, C_Q, and C_U can be chosen so that the argument works. Furthermore, from the proof given below it will be clear that every sub-circuit of Q'_j can be uniformly constructed in deterministic time that is polynomial in the size of the sub-circuit. From this and using that it is easy to describe how these sub-circuits are connected, we get that the sequence $\{Q'_j\}_{j \geq \kappa}$ of quantum circuits Q'_j is uniform.

Let $j \in \{\kappa, \ldots, n\}$ be fixed, and assume that Q'_j is given as input a string $\mathsf{code}(P_{j-1})$ representing a quantum circuit P_{j-1} of size $\leq s_P(j-1)$ that computes f_{j-1}. First, Q'_j invokes the deterministic circuit $\mathcal{C}^{\mathsf{DR}}_j$ from Lemma 4.30 on $\mathsf{code}(P_{j-1})$, obtaining from it a string $\mathsf{code}(F_j)$ describing a quantum circuit F_j of size $\mathsf{poly}(s_P(j-1))$ that computes f_j. Note that F_j might contain more than $s_P(j)$ gates, so it cannot be used as the output circuit P_j. However, we show next that we can use F_j to uniformly construct the circuit P_j that computes f_j with size $s_P(j)$. For that we need four steps:

1. From a large circuit for f_j to a smaller approximate circuit for h_j (with non-trivial probability). Circuit Q'_j takes the code of F_j and produces from it a string $\mathsf{code}(H_j)$ describing a quantum circuit H_j that computes the function h_j defined above. Note that the definition of H_j from F_j is elementary, and that $\mathsf{size}(H_j) = \mathsf{poly}(\mathsf{size}(F_j)) = \mathsf{poly}(s_P(j-1))$. Recall that h_j is defined over $\ell(j) = \mathsf{poly}(j)$ input bits, and that the corresponding function NW^{h_j} from above produces $m(j) = \lfloor 2^{j^\lambda} \rfloor$ output bits. We now invoke Lemma 4.4 with parameters associated with the Nisan-Wigderson generator for index value j: function h_j and its corresponding quantum circuit H_j, stretch $m(j) = \lfloor 2^{j^\lambda} \rfloor$, and distinguisher circuit $D_{m(j)}$ of size $s(j) = 2^{j^{2\lambda}}$ and advantage $\gamma(j) = 1/m(j)$. From this lemma and our choice of parameters, it follows that Q'_j has access to a quantum

circuit $\mathcal{C}^{\mathsf{NW}}$ of size $\mathsf{poly}(m(j), s_P(j-1), s(j)) = \mathsf{poly}(s_P(j-1)) = 2^{O(j^{2\lambda})}$ such that, when given $\mathsf{code}(H_j)$, $\mathsf{code}(D_{m(j)})$, the input length of h_j and the stretch value $m(j)$, it outputs with probability at least $\Omega(\gamma(j)/m(j)^2) = \Omega(1/m(j)^3) = \Omega(2^{-3\cdot j^\lambda})$ the encoding $\mathsf{code}(A_j)$ of a quantum circuit A_j of size $O(m(j)^2 \cdot s(j)) = O(2^{1.01\cdot j^{2\lambda}})$ such that

$$\Pr_{x,\, r,\, A_j}[A_j(x,r) = h_j(x)] \;\geq\; \frac{1}{2} + \frac{\gamma(j)}{2m(j)} \;=\; \frac{1}{2} + \frac{1}{2m(j)^2} \;\geq\; \frac{1}{2} + 2^{-3\cdot j^\lambda}. \tag{59}$$

Crucially, note that the size of A_j does not depend on the sizes of F_j and P_{j-1}. In its next steps, Q'_j tries to compute from A_j a quantum circuit for f_j of size $\leq 2^{C_P \cdot j^{2\lambda}}$, while maintaining its total number of gates $\leq 2^{C_Q \cdot j^{2\lambda}}$ (we will boost the success probability of Q'_j later in the proof). Note that the constant C_Q might depend on C_P once we fix a large enough C_P, and indeed this is needed for this plan to work.

2. From a circuit approximating h_j to a non-trivial quantum circuit for g_j (with probability 1). Given the string $\mathsf{code}(A_j)$ produced by $\mathcal{C}^{\mathsf{NW}}$ in the step above, Q'_j proceeds as follows. It instantiates the corresponding deterministic circuit $\mathcal{C}^{\mathsf{GL}}$ from Lemma 4.5, which computes from $\mathsf{code}(A_j)$ a string $\mathsf{code}(B_j)$ describing a quantum circuit B_j, with $\mathsf{size}(B_j) = \mathsf{poly}(j, k(j)) \cdot \mathsf{size}(A_j) = O(2^{1.02\cdot j^{2\lambda}})$. If we assume that A_j satisfies Equation (59), it follows from Lemma 4.5 that

$$\Pr_{x,\, B_j}[B_j(x) = g_j(x)] \;\geq\; \frac{(2^{-3\cdot j^\lambda})^3}{2} \;\geq\; 2^{-10\cdot j^\lambda}. \tag{60}$$

Moreover, note that $\mathsf{size}(\mathcal{C}^{\mathsf{GL}}) = \mathsf{poly}(j, k(j), \mathsf{size}(A_j)) = 2^{O(j^{2\lambda})}$.

3. From a non-trivial circuit for g_j to an excellent circuit for f_j (with non-trivial probability). Given the string $\mathsf{code}(B_j)$ produced by $\mathcal{C}^{\mathsf{GL}}$ in the step above, and assuming for now that B_j satisfies Equation 60, Q'_j proceeds as follows.[14] Let $\varepsilon'(j) \stackrel{\text{def}}{=} 2^{-10\cdot j^\lambda}$ and $\delta(j) \stackrel{\text{def}}{=} j^{-2b^\star + a_\star}$, and consider $k(j)$ and the size bound for B_j established above. Then, for this choice of parameters as a function of j, it is not hard to check that Equation 48 in Theorem 4.28 holds. Indeed,

$$k(j) = 2j^{2b_\star + a_\star + d_\star + 2}, \quad \text{while} \quad \frac{1}{\delta(j)} \cdot \left[\log\left(\frac{1}{\delta(j)}\right) + \log\left(\frac{1}{\varepsilon'(j)}\right)\right] \;\ll\; j^{2b_\star + a_\star + \lambda + 1} \;\ll\; j^{2b_\star + a_\star + 2}.$$

Let $\mathcal{C}^{\mathsf{IJKW}}$ be the quantum circuit provided by Theorem 4.28 for our choice of parameters. We rewrite Equation 60 more explicitly as follows:

$$\Pr_{x^1,\ldots,x^k \sim (\{0,1\}^j)^k,\, B_j}\Big[B_j\big(x^1,\ldots,x^k\big) = \big(f_j(x^1),\ldots,f_j(x^k)\big)\Big] \;\geq\; 2^{-10\cdot j^\lambda}, \tag{61}$$

where by assumption $0 < \lambda < 1/5$. Now note that there is a mismatch between the expression above and the assumption in Equation 49, because there we sample a k-tuple of j-bit strings according to $\mathcal{S}_{j,k}$,[15] i.e., there is no repetition of strings and we assume a canonical order of the tuple when using it as an input string of length $k \cdot j$ bits. To remedy this situation, Q'_j will not invoke $\mathcal{C}^{\mathsf{IJKW}}$ directly on $\mathsf{code}(B_j)$, as we explain next.

Define the quantum circuit A'_j that attempts to compute g_j on $\mathcal{S}_{j,k}$ based on B_j as follows:

[14] To be more formal Q'_j proceeds as we describe independently of the assumption that B_j satisfies Equation (60), but we keep this assumption for simplicity of exposition.

[15] Remember from Definition 4.6 that we define $\mathcal{S}_{j,k} = \{S \subseteq \{0,1\}^j : |S| = k\}$.

1. On an input $\vec{x} \in \mathcal{S}_{j,k}$,
2. Sample a random permutation $\pi\colon [k] \to [k]$.
3. Permute the k strings in $\vec{x}$ according to π, and let $\vec{y} \stackrel{\text{def}}{=} \pi(\vec{x})$ be the corresponding kj-bit string.
4. Output $B_j(\vec{y})$.

Claim 5.3. *The following holds:*

$$\Pr_{\vec{x}\sim\mathcal{S}_{j,k},\, A'_j}\left[A'_j(\vec{x}) = g_j(\vec{x})\right] \;\geq\; \frac{2^{-10\cdot j^\lambda}}{2}.$$

Proof of Claim 5.3. Indeed, using the definition of A'_j, and recalling the definition of g_j,

$$\begin{aligned}
\Pr_{\vec{x}\sim\mathcal{S}_{j,k},\, A'_j}\left[A'_j(\vec{x}) = g_j(\vec{x})\right] &= \Pr_{\substack{\vec{x}\sim\mathcal{S}_{j,k}\\ \vec{y}=\pi(\vec{x}),\, B_j}}\left[B_j(\vec{y}) = g_j(\vec{y})\right] \\
&= \Pr_{\vec{x}\sim\{0,1\}^{jk},B_j}\left[B_j(\vec{x}) = g_j(\vec{x}) \mid \text{strings in } \vec{x} \text{ are distinct}\right] \\
&\geq \Pr_{\vec{x}\sim\{0,1\}^{jk},B_j}\left[B_j(\vec{x}) = g_j(\vec{x}) \;\wedge\; \text{strings in } \vec{x} \text{ are distinct}\right] \\
(\Pr[E_1 \wedge E_2] \geq \Pr[E_1] - \Pr[\neg E_2]) &\geq \Pr_{\vec{x}\sim\{0,1\}^{jk},B_j}\left[B_j(\vec{x}) = g_j(\vec{x})\right] - \Pr_{\vec{x}\sim\{0,1\}^{jk}}\left[\exists i_1 \neq i_2 \text{ s.t. } x^{i_1} = x^{i_2}\right] \\
&\geq 2^{-10\cdot j^\lambda} - k^2\cdot 2^{-j} \;\geq\; \frac{2^{-10\cdot j^\lambda}}{2},
\end{aligned}$$

where the last inequality used that $\lambda < 1$ and $k = k(j) = \mathsf{poly}(j)$. □

Circuit Q'_j constructs $\mathsf{code}(A'_j)$ from $\mathsf{code}(B_j)$, then invokes $\mathcal{C}^{\mathsf{IJKW}}$ on $\mathsf{code}(A'_j)$. Assuming Equation 61 holds, $\mathcal{C}^{\mathsf{IJKW}}$ outputs with probability $\Omega(\varepsilon'(j)^2) = \Omega(2^{-20\cdot j^\lambda})$ a string $\mathsf{code}(B'_j)$ describing a quantum circuit B'_j of size

$$\mathsf{size}(B'_j) = \mathsf{poly}(j, k(j), \mathsf{size}(B_j), \log(1/\delta(j)), 1/\varepsilon'(j)) = \mathsf{poly}(\mathsf{size}(B_j)) = 2^{O(j^{2\lambda})} \tag{62}$$

such that

$$\Pr_{x\sim\{0,1\}^j,B'_j}\left[B'_j(x) = f_j(x)\right] \;\geq\; 1 - \delta(j) \;=\; 1 - j^{-2b_\star - a_\star}.$$

Note that $\mathsf{size}(\mathcal{C}^{\mathsf{IJKW}}) = 2^{O(j^{2\lambda})}$.

Summary of Steps 1–3. By composing Steps 1–3 described above, we get that for every sufficiently large constant C_1 (such that Equation 62 holds) there is a constant $C_2 > C_1$ and a quantum circuit Q'_j of size $2^{C_2\cdot j^{2\lambda}}$ that when given a description $\mathsf{code}(P_{j-1})$ of a quantum circuit P_{j-1} of size $2^{C_1\cdot(j-1)^{2\lambda}}$ that computes f_{j-1}, outputs with probability $\zeta(j) = \Omega(2^{-23\cdot j^\lambda})$ the description of a quantum circuit $\widetilde{P_j}$ of size $\leq 2^{C_1\cdot j^{2\lambda}}$ such that

$$\Pr_{x\sim\{0,1\}^j,\widetilde{P_j}}\left[\widetilde{P_j}(x) = f_j(x)\right] \;\geq\; 1 - j^{-2b_\star - a_\star}.$$

(Crucially, the size of $\widetilde{P_j}$ does not depend on the exponent C_2 nor on the size of P_{j-1}, thanks to the results from Section 4 and the existence of distinguisher circuits $D_{m(j)}$ for every $j \geq \kappa$.) In

order to complete the proof of Lemma 5.2, it remains for us to (1) amplify the success probability that Q'_j generates an almost-correct circuit $\widetilde{P_j}$; and (2) convert an almost-correct circuit $\widetilde{P_j}$ into a quantum circuit P_j that computes f_j on each input with probability at least $2/3$. These two goals are achieved next.

4. Amplifying the success probability of $Q'_j(\mathsf{code}(P_{j-1}))$ and generating a correct circuit P_j for f_j. Our final version of Q'_j works as follows. This circuit takes its (classical) input $\mathsf{code}(P_{j-1})$ and runs Steps 1–3 for a total of $t(j) \stackrel{\text{def}}{=} \mathsf{poly}(j, 1/\zeta(j)) = 2^{O(j^\lambda)}$ times, obtaining from this a collection $\widehat{P}_1, \ldots, \widehat{P}_{t(j)}$ of candidate quantum circuits such that,

$$\text{with probability } \geq 1 - 1/500j^2, \text{ there is } i \in [t(j)] \text{ s.t. } \Pr_{x \sim \{0,1\}^j, \widehat{P_i}} \left[\widehat{P}_i(x) = f_j(x)\right] \geq 1 - j^{-2b_\star - a_\star}.$$

Now Q'_j instantiates the corresponding circuit $\mathcal{C}^{\mathsf{SR}}$ from Theorem 4.31 on inputs 1^j, $\mathsf{code}(\widehat{P}_i)_{i \in [t(j)]}$, and $\mathsf{code}(P_{j-1})$ in order to output with high probability a (single) circuit P_j that computes f_j. Note that despite the blowup in the size of Q'_j and P_j due to the amplification, our size requirements for them are maintained. In particular, we have from Theorem 4.31 that the size of the output circuit P_j does not depend on $\mathsf{size}(P_{j-1})$.

This finishes the construction of circuits Q'_j satisfying the conditions of Lemma 5.2, which completes the proof of Theorem 5.1. □

□

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. 17, 34, 39

[ABG06] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Machine learning in a quantum world. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 431–442. Springer, 2006. 3

[AC02] Mark Adcock and Richard Cleve. A quantum Goldreich-Levin theorem with cryptographic applications. In *Symposium on Theoretical Aspects of Computer Science* (STACS), pages 323–334. Springer, 2002. 13, 32, 38

[ACL+19] Srinivasan Arunachalam, Sourav Chakraborty, Troy Lee, Manaswi Paraashar, and Ronald de Wolf. Two new results about quantum exact learning. In *International Colloquium on Automata, Languages, and Programming* (ICALP), volume 132, pages 16:1–16:15, 2019. 3

[AGS21] Srinivasan Arunachalam, Alex Bredariol Grilo, and Aarthi Sundaram. Quantum hardness of learning shallow classical circuits. *SIAM J. Comput.*, 50(3):972–1013, 2021. 3, 10

[Aha03] Dorit Aharonov. A simple proof that Toffoli and Hadamard are quantum universal. *CoRR*, abs/0301040, 2003. 19

[AS05] Alp Atici and Rocco A. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005. 3

[AS07] Alp Atıcı and Rocco A. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2007. 3

[AS16] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John Wiley & Sons, 2016. 17

[AW17] Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *ACM SIGACT News*, 48(2):41–67, 2017. 3, 21

[AW18] Srinivasan Arunachalam and Ronald de Wolf. Optimal quantum sample complexity of learning algorithms. *The Journal of Machine Learning Research*, 19(1):2879–2878, 2018. 3

[BFGH10] Debajyoti Bera, Stephen Fenner, Frederic Green, and Steven Homer. Efficient universal quantum circuits. *Quantum Information & Computation*, 10(1):16–28, 2010. 19

[BHLR19] Abhishek Bhrushundi, Kaave Hosseini, Shachar Lovett, and Sankeerth Rao. Torus polynomials: An algebraic approach to ACC lower bounds. In *Innovations in Theoretical Computer Science Conference* (ITCS), pages 13:1–13:16, 2019. 14

[BJ98] Nader H. Bshouty and Jeffrey C Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136–1153, 1998. 3

[BT94] Richard Beigel and Jun Tarui. On ACC. *Comput. Complex.*, 4:350–366, 1994. 14

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Conference on Computational Complexity* (CCC), pages 10:1–10:24, 2016. 7, 15

[CLW20] Lijie Chen, Xin Lyu, and Ryan Williams. Almost-everywhere circuit lower bounds from non-trivial derandomization. In *Symposium on Foundations of Computer Science* (FOCS), 2020. 16

[COS18] Ruiwen Chen, Igor C. Oliveira, and Rahul Santhanam. An average-case lower bound against ACC^0. In *Latin American Symposium on Theoretical Informatics* (LATIN), pages 317–330, 2018. 16

[CR20] Lijie Chen and Hanlin Ren. Strong average-case lower bounds from non-trivial derandomization. In *Symposium on Theory of Computing* (STOC), pages 1327–1334, 2020. 16

[CRTY20] Lijie Chen, Ron Rothblum, Roei Tell, and Eylon Yogev. On exponential-time hypotheses, derandomization, and circuit lower bounds. In *Symposium on Foundations of Computer Science* (FOCS), 2020. 6, 8, 12, 16

[FK09] Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009. 6, 7, 11, 15

[GKZ19] Alex B. Grilo, Iordanis Kerenidis, and Timo Zijlstra. Learning with Errors is easy with quantum samples. *Physical Review Letters A*, 99:032314, 2019. arXiv: 1702.08255. 3

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Symposium on Theory of Computing* (STOC), pages 25–32, 1989. 13, 32

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Symposium on Theory of Computing* (STOC), pages 212–219, 1996. 3

[HH13] Ryan C. Harkins and John M. Hitchcock. Exact learning algorithms, betting games, and circuit lower bounds. *Transactions on Computation Theory* (TOCT), 5(4):18, 2013. 6, 15

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Staist. Assoc.*, pages 13–30, 1963. 17

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: simplified, optimized, and derandomized. *SIAM Journal on Computing*, 39(4):1637–1665, 2010. 6, 9, 12, 13, 32, 39, 41, 42, 43, 45, 46, 53

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Symposium on Foundations of Computer Science* (FOCS), pages 538–545, 1995. 13

[IW01] Russell Impagliazzo and Avi Wigderson. Randomness vs time: Derandomization under a uniform assumption. *J. Comput. Syst. Sci.*, 63(4):672–688, 2001. 8, 12, 13, 32

[JLR11] Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random Graphs*. John Wiley & Sons, 2011. 17

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*. Springer, 2012. 18

[Kha93] Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Symposium on Theory of Computing* (STOC), pages 372–381, 1993. 3

[KKO13] Adam Klivans, Pravesh Kothari, and Igor C. Oliveira. Constructing hard functions using learning algorithms. In *Conference on Computational Complexity* (CCC), pages 86–97, 2013. 6, 7, 11, 15

[KS94] Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. *J. Comput. Syst. Sci.*, 48(3):464–497, 1994. 26

[KV94] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994. 18

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993. 14, 15

[MW18] Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Symposium on Theory of Computing* (STOC), pages 890–901, 2018. 16

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. 19

[NR99] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *Journal of Computer and System Sciences*, 58(2):336–375, 1999. 3

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994. 8, 12, 13, 32, 33

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. 73

[Oli13] Igor C. Oliveira. Algorithms versus circuit lower bounds. *Electronic Colloquium on Computational Complexity*, 20:117, 2013. 10

[Oli19] Igor C. Oliveira. Randomness and intractability in Kolmogorov complexity. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 32:1–32:14, 2019. 6, 14, 15

[OS17] Igor C. Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Computational Complexity Conference* (CCC), pages 18:1–18:49, 2017. 4, 6, 7, 10, 11, 14, 15, 29, 32

[OS18] Igor C. Oliveira and Rahul Santhanam. Pseudo-derandomizing learning and approximation. In *International Workshop on Randomization and Computation* (RANDOM), pages 55:1–55:19, 2018. 6, 15

[OW16] Ryan O'Donnell and John Wright. Efficient quantum tomography. In *Symposium on Theory of Computing* (STOC), pages 899–912, 2016. 3

[OW17] Ryan O'Donnell and John Wright. Efficient quantum tomography II. In *Symposium on Theory of Computing* (STOC), pages 962–974, 2017. 3

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997. 10, 15, 22

[San09] Rahul Santhanam. Circuit lower bounds for merlin–arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009. 5

[SG04] Rocco A. Servedio and Steven J. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing*, 33(5):1067–1092, 2004. 3, 21

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Symposium on Foundations of Computer Science* (FOCS), pages 124–134, 1994. 3

[ST17] Rocco A. Servedio and Li-Yang Tan. What circuit classes can be learned with non-trivial savings? In *Innovations in Theoretical Computer Science Conference* (ITCS), pages 30:1–30:21, 2017. 14, 15

[TV07] Luca Trevisan and Salil P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007. 7, 8, 12, 23

[Vol14] Ilya Volkovich. On learning, lower bounds and (un)keeping promises. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 1027–1038, 2014. 6, 10, 15

[Vol16] Ilya Volkovich. A guide to learning arithmetic circuits. In *Conference on Learning Theory* (COLT), pages 1540–1561, 2016. 6, 15

[Wil13] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013. 16

[Wil14] Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, 2014. 3, 5, 14, 16

[Wil18] R. Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. *Theory Comput.*, 14(1):1–25, 2018. 14

[Yam92] Kenji Yamanishi. A learning criterion for stochastic rules. *Mach. Learn.*, 9:165–203, 1992. 26

A On trivial quantum learning algorithms

In this section, we explain in more detail that there are two quantum learners with different parameters that are "trivial", in the sense that they do not really exploit the structure of a concept class $\mathfrak{C}$. First observe that, even classically, there is always a "brute-force" learner that works for *all* possible functions: query the input function $f : \{0,1\}^n \to \{0,1\}$ on all inputs, then store the outcomes in a lookup table which is used as the exponentially large output hypothesis.

The learner above also works in the quantum setting. However, notice that quantumly there exists a second "trivial" learner coming from *Fourier sampling*. Before we describe this learner, we briefly discuss the basics of Fourier analysis on the Boolean cube (and refer the reader to [O'D14] for more details).

Given the space of functions $f : \{0,1\}^n \to \mathbb{R}$, define the inner product between two functions in this space as $\langle f, g\rangle = \mathbb{E}_x[f(x) \cdot g(x)]$ where the expectation is taken uniformly from $x \in \{0,1\}^n$. In this space, one can define a set of *orthonormal basis functions* as follows: for $S \in \{0,1\}^n$, define $\chi_S(x) = (-1)^{S \cdot x}$ where $S \cdot x = \sum_i S_i \cdot x_i$. Hence, every function $f : \{0,1\}^n \to \mathbb{R}$ can be written *uniquely* as

$$f(x) = \sum_S \widehat{f}(S)\chi_S(x),$$

where $\widehat{f}(S) = \mathbb{E}_x[f(x) \cdot \chi_S(x)]$ is called a *Fourier coeffient* of f. Moreover, it is not hard to see that by Parseval's identity, for every Boolean function $f : \{0,1\}^n \to \{-1,1\}$,

$$\sum_S \widehat{f}(S)^2 = \mathop{\mathbb{E}}_x[f(x)^2] = 1,$$

hence the set of of squared Fourier coefficients $\{\widehat{f}(S)^2\}_S$ of a Boolean function f forms a probability distribution.

It is well known that in the quantum learning model, given one uniform quantum example $\frac{1}{\sqrt{2^n}} \sum_x |x, f(x)\rangle$ for $f : \{0,1\}^n \to \{-1,1\}$, with probability $1/2$ we can *Fourier sample*, i.e., sample from the distribution $\{\widehat{f}(S)^2\}_S$. Indeed, given $\frac{1}{\sqrt{2^n}} \sum_x |x, f(x)\rangle$, apply the one-qubit Hadamard gate on the last register and measure the last qubit: with probability $1/2$ we get the 1 outcome, in which case the resulting state is $\frac{1}{\sqrt{2^n}} \sum_x f(x) |x\rangle |1\rangle$, then apply the n-qubit Hadamard transform on the first n qubits to obtain the state $\sum_S \widehat{f}(S) |S\rangle |1\rangle$. Measuring this state produces a sample S from the distribution $\{\widehat{f}(S)^2\}_S$. Let $\boldsymbol{S}$ be the random variable that outputs $S \subseteq [n]$ with probability $\widehat{f}(S)^2$.

Claim A.1. *For every $0 \leq \gamma \leq 1$ and $f : \{0,1\}^n \to \{-1,1\}$, we have*

$$\Pr_{\boldsymbol{S}}\left[\, |\hat{f}(\boldsymbol{S})| \geq \gamma \cdot 2^{-n/2} \,\right] \;\geq\; 1 - \gamma^2.$$

Proof. It is enough to show that $p \stackrel{\text{def}}{=} \Pr_{\boldsymbol{S}}[\, \hat{f}(\boldsymbol{S})^2 \geq \varepsilon \cdot 2^{-n} \,] \geq 1 - \varepsilon$. Note that with probability $1 - p$ over the choice of $\boldsymbol{S}$, we have $\hat{f}(\boldsymbol{S})^2 < \varepsilon \cdot 2^{-n}$. But then

$$\sum_{S\colon\, \hat{f}(S)^2 < \varepsilon \cdot 2^{-n}} \hat{f}(S)^2 \;=\; 1 - p.$$

This in turn implies that

$$2^n \cdot (\varepsilon \cdot 2^{-n}) \;\geq \sum_{S\colon\, \hat{f}(S)^2 < \varepsilon \cdot 2^{-n}} \varepsilon \cdot 2^{-n} \;\geq\; 1 - p,$$

which completes the proof. □

Let $\mathfrak{F}_n$ be the class of all Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$. Claim A.1 implies that $\mathfrak{F}_n$ admits a quantum learning algorithm $\mathcal{A}$ under the uniform distribution with the following property: for every $f \in \mathfrak{F}_n$, $\mathcal{A}$ runs in polynomial time and outputs with probability at least 0.249 a Boolean circuit C that computes f with probability $\frac{1}{2} + \Omega(2^{-n/2})$. In order to see this, we use Fourier sampling to obtain a set $\boldsymbol{S}$, and then with probability $\frac{1}{2}$ we let $C = \chi_S$ and with probability $\frac{1}{2}$ we let $C = \neg\chi_S$. By Claim A.1, we are guaranteed that with probability at least 0.999, we pick some S such that χ_S or $\neg\chi_S$ computes f with probability $\frac{1}{2} + \Omega(2^{-n/2})$, and we pick the correct one with probability $\frac{1}{2}$. Since the Fourier sampling routine described above samples from the correct distribution with probability $1/2$, we are done.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il