

# From estimation of quantum probabilities to simulation of quantum circuits

Hakop Pashayan,<sup>1</sup> Stephen D. Bartlett,<sup>1</sup> and David Gross<sup>2</sup>

<sup>1</sup>*Centre for Engineered Quantum Systems, School of Physics,  
The University of Sydney, Sydney, NSW 2006, Australia*

<sup>2</sup>*Institute for Theoretical Physics, University of Cologne, Germany*  
(Dated: 7 December 2017)

We show that there are two distinct aspects of a general quantum circuit that can make it hard to efficiently simulate with a classical computer. The first aspect, which has been well-studied, is that it can be hard to efficiently estimate the probability associated with a particular measurement outcome. However, we show that this aspect alone does not determine whether a quantum circuit can be efficiently simulated. The second aspect is that, in general, there can be an exponential number of ‘relevant’ outcomes that are needed for an accurate simulation, and so efficient simulation may not be possible even in situations where the probabilities of individual outcomes can be efficiently estimated. We show that these two aspects are distinct, the former being necessary but not sufficient for simulability whilst the pair is jointly sufficient for simulability. Specifically, we prove that a family of quantum circuits is efficiently simulable if it satisfies two properties: one related to the efficiency of Born rule probability estimation, and the other related to the sparsity of the outcome distribution. We then prove a pair of hardness results (using standard complexity assumptions and a variant of a commonly-used average case hardness conjecture), where we identify families of quantum circuits that satisfy one property but not the other, and for which efficient simulation is not possible. To prove our results, we consider a notion of simulation of quantum circuits that we call  $\epsilon$ -simulation. This notion is less stringent than exact sampling and is now in common use in quantum hardness proofs.

## I. INTRODUCTION AND SUMMARY OF MAIN RESULTS

Which quantum processes can be efficiently simulated using classical resources is a fundamental and longstanding problem [1–6]. Research in this area can be split into two broad classes: results showing the hardness of efficient classical simulation for certain quantum processes, and the development of efficient classical algorithms for simulating other quantum processes. Recently, there has been substantial activity on both sides of this subject. Works on boson sampling [7], instantaneous quantum polynomial (IQP) circuits [8, 9], various translationally invariant spin models [10, 11], quantum Fourier sampling [12], one clean qubit (also known as DQC1) circuits [13, 14], chaotic quantum circuits [15] and conjugated Clifford circuits [16] have focused on showing the difficulty of classically simulating these quantum circuits. On the other hand, there has been substantial recent progress in classically simulating various elements of quantum systems including matchgate circuits with generalized inputs and measurements [17], circuits with positive quasi-probabilistic representations [18, 19], stabilizer circuits supplemented with a small number of  $T$  gates [20], stabilizer circuits with small coherent local errors [21], low negativity magic state injection in the fault tolerant circuit model [22], quantum circuits with polynomial bounded negativity [23] and abelian-group normalizer circuits [24, 25]. In addition, there has been some work on using small quantum systems to simulate larger quantum systems [26] as well as using noisy quantum systems to simulate ideal ones [27].

The most commonly used mathematical model for the concept of a “computational problem” is a *decision problem*, where the task is to decide whether a given input bit string belongs one of two classes. In quantum language, the result of the computation will thus be determined by the outcome of a binary measurement. A classical simulation of a quantum process that solves a decision problem then corresponds to estimating the Born rule probability of this binary outcome. However, it has been increasingly appreciated that this model may be too narrow: local measurements on many-body quantum systems produce probabilistic results that lie in an outcome space that grows exponentially with the number of particles. This situation is modelled more naturally as a *sampling problem*, and indeed, this problem class has received growing attention recently (see, for example, Refs. [11, 15, 16, 28]) and have been argued to be a promising avenue for first experimental demonstrations of a quantum advantage. Analyzing the notion of “classical simulation” for sampling problems turns out to be much more subtle than for decision problems, and a number different definitions are used in the literature. The purpose of this work is to describe a framework for analyzing classical simulations of quantum sampling problems. In particular, we show that the ability to estimate Born rule probabilities is no longer sufficient for simulability.

## A. Outline of our main results

### 1. Defining $\epsilon$ -simulation.

In Section II, we introduce and discuss a precise notion of efficient simulation, which we call  $\epsilon$ -simulation. Related notions of simulation have been used in prior works, e.g., in Refs. [7, 9, 20] in the context of hardness or existence of classical simulators. Essentially, we say that an algorithm can  $\epsilon$ -simulate a family of quantum circuits, if for any  $\epsilon > 0$ , it can sample from a distribution that is  $\epsilon$ -close in  $L_1$ -norm to the true output distribution of the circuit, and if the algorithm runs in time polynomial in  $1/\epsilon$  and in the number of qubits. We provide an operational meaning for this notion by showing that “possessing an  $\epsilon$ -simulator” for a family of circuits is equivalent to demanding that even a computationally omnipotent referee cannot efficiently distinguish the simulator’s outputs from the quantum ones.

**Theorem 1.** *Bob has an  $\epsilon$ -simulator of Alice’s quantum computer if and only if given the hypothesis testing scenario considered in Sec. II C there exists a strategy for Bob which jointly achieves indistinguishability and efficiency.*

### 2. Efficient outcome estimation: the poly-box.

In Section III, we consider the question of whether the quantum probability associated with a particular outcome of a quantum circuit can be efficiently estimated. We introduce the notion of a *poly-box*, which is a device that computes an additive polynomial precision estimate of the quantum probability (or marginal probability) associated with a specific outcome of a quantum circuit.

We give three examples of poly-boxes. The first one is an estimator based on Monte Carlo sampling techniques applied to a quasiprobability representation. This follows the work of Ref. [23], where it was found that the efficiency of this estimator depends on the amount of “negativity” in the quasiprobability description of the quantum circuit. We then consider the family of circuits  $\mathcal{C}_{\text{PROD}}$ , for which the  $n$ -qubit input state  $\rho$  is an arbitrary product state (with potentially exponential negativity), transformations consist of Clifford unitary gates, and measurements are of  $k \leq n$  qubits in the computational basis. We present an explicit classical poly-box for  $\mathcal{C}_{\text{PROD}}$  in Section III. As a third example, we also outline a construction of a classical poly-box for Instantaneous Quantum Polynomial-time (IQP) circuits  $\mathcal{C}_{\text{IQP}}$  based on the work of Ref. [29].

### 3. From estimation to simulation.

Previous work on simulation of quantum circuits has primarily focussed on decision problems, where estimating the probability of a single binary outcome is sufficient for simulation. When considering sampling problems, however, it was not previously known in general whether efficient Born rule probability estimation can allow for  $\epsilon$ -simulation. Being able to efficiently estimate the probability (or marginal probability) associated with any particular outcome does not necessarily allow for efficient simulation, because there may be an exponentially large number of outcomes that are relevant to the simulation. The second property of a quantum circuit we consider is whether the outcome distribution is sparse or not. The sparsity property measures “peakedness versus uniformness” of quantum distributions and is related to the scaling of the smooth max-entropy of the output distributions of quantum circuits. Loosely, a *poly-sparse* quantum circuit can have its outcome probability distribution well approximated by specifying the probabilities associated with polynomially many of the most likely outcomes. We formalise this notion in Section IV.

Also in Section IV, we prove that, if a family of quantum circuits possesses a poly-box (allowing for efficient estimation of outcome probabilities) and satisfies a poly-sparsity condition on the outcome distribution, then this family of quantum circuits can be efficiently simulated.

**Theorem 2.** *Let  $\mathcal{C}$  be a family of quantum circuits with a corresponding family of probability distributions  $\mathbb{P}$ . Suppose there exists an efficient poly-box over  $\mathcal{C}$ , and that  $\mathbb{P}$  is poly-sparse. Then, there exists a  $\epsilon$ -simulator of  $\mathcal{C}$ .*

We emphasize that the proof of this theorem is constructive, and allows for new simulation results for families of quantum circuits for which it was not previously known if they were efficiently simulable. As an example, our results can be straightforwardly used to show that Clifford circuits with sparse outcome distributions and with small amounts of local unitary (non-Clifford) noise, as described in Ref. [21], are  $\epsilon$ -simulable.

In Section IV, we also show that families of quantum circuits must admit a poly-box in order to be  $\epsilon$ -simulable.

**Theorem 3.** *If  $\mathcal{C}$  is a family of quantum circuits that does not admit a poly-box algorithm, then  $\mathcal{C}$  is not  $\epsilon$ -simulable.*

#### 4. Hardness results.

Finally, in Section V, we prove that the poly-box requirements of Theorem 2 is on its own not sufficient for  $\epsilon$ -simulability. The challenge to proving such a result is identifying a natural family of non-poly-sparse quantum circuits for which a poly-box exists but for which  $\epsilon$ -simulation is impossible.

We prove that the family  $\mathcal{C}_{\text{PROD}}$  described above, for which we have an explicit poly-box, violates the poly-sparsity requirement. Then, by making a now commonly used “average case hardness” conjecture [7, 9, 10, 12, 16, 30], we prove that the ability to perform  $\epsilon$ -simulation of  $\mathcal{C}_{\text{PROD}}$  would prove that the polynomial hierarchy collapses to the third level. Loosely, this result means that there exist quantum circuits where the probability of any individual outcome (and marginals) can be efficiently estimated, but due to the vast number of relevant outcomes the system cannot be efficiently simulated. Our hardness result closely follows the structure of several similar results, and in particular that of the IQP circuits result of Ref. [9].

Our proof relies on a conjecture regarding the hardness of estimating Born rule probabilities to within a small multiplicative factor for a substantial fraction of randomly chosen circuits from  $\mathcal{C}_{\text{PROD}}$ . This average case hardness conjecture (which we formulate explicitly as Conjecture 1) is a strengthening of the worst case hardness of multiplicative precision estimation of probabilities associated with circuits from  $\mathcal{C}_{\text{PROD}}$ . Worst case hardness follows from an analogous argument to Theorem 5.1 of Ref. [16].

**Theorem 4.** *If there exists an  $\epsilon$ -simulator of  $\mathcal{C}_{\text{PROD}}$  and Conjecture 1 holds, then the polynomial hierarchy collapses to the third level.*

## B. Discussion

We believe the results presented here can serve as a framework for a more structured and quantitative focus on the following question: “In terms of sampling problems (as opposed to decision problems), what makes quantum devices computationally more powerful than a classical probabilistic Turing machine?”. Previous results make it clear that quantum phenomena such as negativity and contextuality play key roles in the super-classical computational power of quantum devices with respect to decision problems. This is evidenced by the apparent difficulty in constructing poly-boxes for such families of quantum circuits. In contrast, we see that there exist families of quantum circuits (e.g.,  $\mathcal{C}_{\text{PROD}}$ ) that admit a poly-box (and as a corollary cannot solve any decision problem outside of BPP) but are nevertheless capable of demonstrating a quantum advantage over any classical device.

Our results show that any family of quantum circuits that both admits a classical poly-box and satisfies the poly-sparsity condition is in the complexity class SampP [31]. In light of the multiple known constructions of poly-boxes (see Refs. [2, 20, 21, 23, 32]) over restricted families of quantum circuits, and in particular Ref. [23], this is a substantial new contribution to the understanding of the computational power of these families of quantum circuits not only in terms of decision problems but more generally, i.e., in terms of sampling and search problems.

Our results may provide insight into related questions in BosonSampling. Interestingly, we note that due to an algorithm by Gurvits [6] (see also Ref. [33]), the family of linear optical quantum circuits considered in the BosonSampling scenario admit additive polynomial precision estimators of individual outcome probabilities. However, there is no known poly-box over this family since it is currently unclear how to produce such estimators for the marginal probabilities. It is plausible that the non-poly-sparse linear optical networks

admit poly-boxes despite the hardness of  $\epsilon$ -simulation for these circuits, placing them in a similar category to  $\mathcal{C}_{\text{IQP}}$  and the Clifford circuit family  $\mathcal{C}_{\text{PROD}}$ .

## II. DEFINING SIMULATION OF A QUANTUM COMPUTER

While there has been a breadth of recent results in the theory of simulation of quantum systems, this breadth has unfortunately been accompanied with a plethora of different notions of simulation. This variety brings with it challenges for comparing results. Consider the following results, which are all based on (often slightly) different notions of simulation. As a first example, the ability to perform *strong simulation* of certain classes of quantum circuits would imply a collapse of the polynomial hierarchy, while under a weaker (but arguably more useful) notion of simulation this collapse is only implied if additional mathematical conjectures hold true [7, 9]. As another example, Ref. [14] shows that the quantum complexity class BQP is contained in the second level of the polynomial hierarchy if there exist efficient classical probabilistic algorithms for sampling a particular outcome (from the quantum circuits considered) at a frequency that is exponentially close to the true quantum probability in terms of additive error (or polynomially close in terms of multiplicative error). As additional examples, Refs. [21, 23] present efficient classical algorithms for additive polynomial precision estimates of Born rule probabilities. While many such technical results are crucially sensitive to these distinctions in the meaning of simulation, there is a growing need to connect the choice of simulation definition used in a proof against (or for) efficient classical simulability to a statement about proofs of quantum advantage (or ability to practically classically solve a quantumly solvable problem). In particular, to the non-expert it can be unclear what the complexity of classical simulation (in each of the above mentioned notions of simulation) of a given quantum device says about the hardness of building a classical device that can efficiently solve the computational problems that are solvable by the quantum device.

In this section, we will introduce and discuss a meaningful notion of efficient simulation, which we call  $\epsilon$ -simulation. Related notions of simulation have been used in prior works, e.g., in Refs. [7, 9, 20]. Further, this notion of simulation corresponds to the class of problems in complexity theory known as sampling problems. Here, we define  $\epsilon$ -simulation and prove some of its properties. In particular, we will show that for a broad class of tests,  $\epsilon$ -simulators are statistically indistinguishable from the quantum process that they simulate.

### A. Strong and weak simulation

We note that every quantum circuit has an associated probability distribution that describes the statistics of the measurement outcomes. We will refer to this as the *circuit's quantum probability distribution*. As an example, Fig. 1 below depicts a quantum circuit. The output of running this circuit is a classical random variable  $X = (X_1, \dots, X_k)$  that is distributed according to the quantum probability distribution.

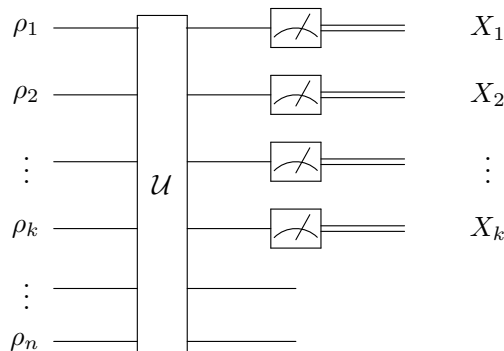


FIG. 1: An example of a quantum circuit. This circuit acts on  $n$  qubits (or, in general, qudits). The initial state is a product state. The unitary operation  $\mathcal{U}$  must be constructed out of a sequence of local unitary gates. The first  $k$  qubits in this example are each measured in a fixed basis, yielding outcome  $(X_1, X_2, \dots, X_k)$ . Qubits  $i > k$ , shown without a measurement, are traced over (marginalized).

Two commonly used notions of simulation are *strong simulation* and *weak simulation*. A weak simulator of a quantum circuit generates samples from the circuit’s quantum probability distribution. In the strict sense of the term, a weak simulator generates samples from the exact quantum probability distribution. Loosely, having a weak simulator for a quantum system is an equivalent resource to using the quantum system itself.

A strong simulator, in contrast, outputs probabilities or marginal probabilities associated with the quantum distributions. More specifically, a strong simulator of a circuit is a device that outputs the quantum probability of observing any particular outcome or the quantum probability of an outcome marginalized over one or more of the measurements. Note that a strong simulator requires an input specifying the event for which the probability of occurrence is required. Taking Fig. 1 as an example, a strong simulator could be asked to return the probability of observing the event  $(X_1, X_2) = (1, 0)$ , marginalized over the measurements 3 to  $k$ . The requirement that a strong simulator can also output estimates of marginals is weaker than requiring them to estimate the quantum probability associated with any event (subset of the outcome space).

While the names ‘strong’ and ‘weak’ simulation suggest that they are in some sense different magnitudes of the same type of thing, we note that these two types of simulation produce different types of output. In particular, a strong simulator outputs probabilities. (More specifically, it outputs exponential additive precision estimates of Born rule probabilities and their marginals.) In contrast a weak simulator outputs samples (from the exact target probability distribution).

## B. $\epsilon$ -simulation

A weak simulator, which generates samples from the exact quantum probability distribution, is a very strict notion. Often, it would be sufficient to consider a simulator that generates samples from a distribution that is only sufficiently close to the quantum distribution, for some suitable measure of closeness. Such a relaxation of the requirement of weak simulation has been used by several authors, e.g., in Refs. [7, 9, 20]. Here, we define the notion of  $\epsilon$ -simulation, which is a particular relaxation of the notion of weak simulation, and motivate its use.

We first define a notion of sampling from a distribution that is only *close* to a given distribution. Consider a discrete probability distribution  $\mathcal{P}$ . Let  $B(\mathcal{P}, \epsilon)$  denote the  $\epsilon$  ball around the target  $\mathcal{P}$  according to the  $L_1$  distance (or equivalently, up to an irrelevant constant, the total variation distance). We define  $\epsilon$ -sampling of a probability distribution  $\mathcal{P}$  as follows:

**Definition 1.** *Let  $\mathcal{P}$  be a discrete probability distribution. We say that a classical device or algorithm can  $\epsilon$ -sample  $\mathcal{P}$  iff for any  $\epsilon > 0$ , it can sample from a probability distribution  $\mathcal{P}^\epsilon \in B(\mathcal{P}, \epsilon)$ . In addition, its run-time should scale at most polynomially in  $1/\epsilon$ .*

We note that the use of the  $L_1$ -norm in the above is motivated by the fact that the  $L_1$ -norm provides lower bounds on the errors in a hypothesis test for distinguishing distributions. More details can be found in the proof of Theorem 1 in Appendix A.

The definition above does not require the device to sample from precisely the quantum probability distribution  $\mathcal{P}$ , but rather allows it to sample from any probability distribution  $\mathcal{P}^\epsilon$  which is in the  $\epsilon$  ball around the target probability distribution,  $\mathcal{P}$ . We note that the device or algorithm will in general take time (or other resources) that depends on the desired precision  $\epsilon$  in order to output a sample, hence the efficiency requirement ensures that these resources scale at most polynomially in the precision  $1/\epsilon$ .

**Definition 2.** *We say that a classical device or algorithm can  $\epsilon$ -simulate a quantum circuit  $c$  if it can  $\epsilon$ -sample from the circuit’s associated output probability distribution  $\mathcal{P}$ .*

We note that each of the above mentioned notions of simulation refers to the simulation of a single quantum circuit. More generally, we may be interested in (strong, weak, or  $\epsilon$ ) simulators of uniform families of quantum circuits. In this setting we can discuss the efficiency of a simulator with respect to  $n$ , the number of qubits<sup>1</sup>. As an example, consider a family of circuits described by a mapping from  $\mathcal{A}^*$  (finite strings over some finite

---

<sup>1</sup> As a technical condition, we require the circuit size, run-time (or any other resources) as well as the length of the circuit’s description to be upper-bounded by  $\text{poly}(n)$ .

alphabet  $\mathcal{A}$ ) to some set of quantum circuits  $\mathcal{C} = \{c_a \mid a \in \mathcal{A}^*\}$  where for each  $a \in \mathcal{A}^*$ ,  $c_a$  is a quantum circuit with some efficient description<sup>2</sup> given by the index  $a$ . In the case of strong (weak) simulation, we say that a device can efficiently strong (weak) simulate the family of quantum circuits  $\mathcal{C}$  if the resources required by the device to strong (weak) simulate  $c_a \in \mathcal{C}$  are upper-bounded by a polynomial in  $n$ . In the case of  $\epsilon$ -simulation, we require that the simulator be able to sample a distribution within  $\epsilon$  distance of the quantum distribution efficiently in both  $n$  and  $1/\epsilon$ .

**Definition 3.** We say that a classical device or algorithm can  $\epsilon$ -simulate a uniform family of quantum circuit  $\mathcal{C}$  if for all  $\epsilon > 0$  and for any  $c \in \mathcal{C}$  (with number of qubits  $n$  and quantum distribution  $\mathcal{P}$ ) it can sample from a probability distribution  $\mathcal{P}^\epsilon \in B(\mathcal{P}, \epsilon)$  in run-time  $O(\text{poly}(n, \frac{1}{\epsilon}))$ .

### C. Motivating the definition of $\epsilon$ -simulation through hypothesis testing

As noted earlier, this definition ensures that  $\epsilon$ -simulation is a weaker form of simulation than exact weak simulation. However, we point out that the notion of exact sampling may be weakened in a number of ways, with the  $\epsilon$ -simulation approach being well suited to many applications related to quantum simulators. As an example, if the definition of simulation allowed for a fixed but small amount of deviation in  $L_1$  distance (as opposed to one that can be made arbitrarily small) then computational power of a simulator will immediately be detectably compromised. The above notion of  $\epsilon$ -simulation requires a polynomial scaling between the precision ( $1/\epsilon$ ) of the approximate sampling and the time taken to produce a sample. In the below (Theorem 1), we will use a statistical indistinguishability argument to show that a polynomial scaling is precisely what should be demanded from a simulator. In particular, we will show that a run-time which scales sub-polynomially in  $1/\epsilon$  puts unnecessarily strong demands on a simulator while a super-polynomial run-time would allow the simulator's output to be statistically distinguishable from the output of the device it simulates.

We now introduce the hypothesis testing scenario we consider.

**Hypothesis testing scenario.** Suppose Alice possesses a quantum computer capable of running a (possibly non-universal) family of quantum circuits  $\mathcal{C}$ , and Bob has some simulation scheme for  $\mathcal{C}$  (whether it's an  $\epsilon$ -simulator is to be decided). Further, suppose that a referee with unbounded computational power will request data from either Alice or Bob and run a test that aims to decide between the hypotheses:

$H_a$ : The requested data came from Alice's quantum computer or

$H_b$ : The requested data came from Bob's simulator.

The setup will be as follows: At the start of the test, one of Alice or Bob will be randomly appointed as "the candidate". Without knowing their identity, the referee will then enter into a finite length interactive protocol with the candidate (see Fig 2). Each round of the protocol will involve the referee sending a circuit description to the candidate requesting the candidate to run the circuit and return the outcome. The choice of requests by the referee may depend on all prior requests and data returned by the candidate. The rules by which the referee:

1. chooses the circuit requested in each round,
2. chooses to stop making further circuit requests and
3. decides on  $H_a$  versus  $H_b$  given the collected data

define the hypothesis test. The goal of the referee is as follows. For any given  $\delta > 0$  decide  $H_a$  versus  $H_b$  such that  $P_{\text{correct}} \geq \frac{1}{2} + \delta$  where  $P_{\text{correct}}$  is the probability of deciding correctly. Bob's goal is to come up with a ( $\delta$ -dependent) strategy for responding to the referee's requests such that it jointly achieves:

- indistinguishability: for any  $\delta > 0$  and for any test that the referee applies,  $P_{\text{correct}} < \frac{1}{2} + \delta$  and

---

<sup>2</sup> Such a description must satisfy the uniformity condition. This can be done by fixing a finite gate set, input state and measurement basis and explicitly defining an efficiently computable mapping between  $\mathcal{A}^*$  and the gate sequence.

- efficiency: for every choice of circuit request sequence  $\alpha$ , Bob must be able to execute his strategy using resources which are  $O(\text{poly}(N(\alpha), \frac{1}{\delta}))$  where  $N(\alpha)$  is the resource cost incurred by Alice for the same circuit request sequence.

We note that the referee can always achieve a success probability  $P_{\text{correct}} = \frac{1}{2}$  simply by randomly guessing  $H_a$  or  $H_b$ . We will show that if Bob has an  $\epsilon$ -simulator then there exists a strategy for Bob such that he jointly achieves indistinguishability (i.e. the referee cannot improve on a random guess by any fixed probability  $\delta > 0$ ) and efficiency.

We note that the efficiency requirement imposed on Bob's strategy is with respect to the resource cost incurred by Alice. Here we will define what this means and justify the rationale behind this requirement. Let us first note that for any circuit  $c_a \in \mathcal{C}$ , there are resource costs  $R(c_a)$  incurred by Alice in order to run this circuit. This may be defined by any quantity as long as this quantity is upper and lower-bounded by some polynomial in the number of qubits. For example,  $R(c_a)$  may be defined by run-time, number of qubits, number of elementary gates, number of qubits plus gates plus measurement, length of circuit description etc. Since this quantity is polynomially equivalent to the number of qubits, without loss of generality, we can treat  $n_a$  (the number of qubits used in circuit  $c_a$ ) as the measure of Alice's resource cost  $R(c_a)$ . We now note that for a given test, the referee may request outcome data from some string of circuits  $c_1, \dots, c_m \in \mathcal{C}$ . Thus we define the resource cost for Alice to meet this request by  $N := n_1 + \dots + n_m$ .

Bob's resource cost (run-time) with respect to each circuit  $c_a \in \mathcal{C}$  is polynomially dependent on both  $n_a$  and the inverse of his choice of accuracy parameter  $\epsilon$ . Thus, Bob's strategy is defined by the rules by which he chooses  $\epsilon_j$ , the accuracy parameter for his response in the  $j^{\text{th}}$  round<sup>3</sup>. Thus, for a given sequence of circuit requests  $a_1, \dots, a_m \in \mathcal{A}^*$ , Bob will incur a resource cost  $T = t_1 + \dots + t_m$  where  $t_j \sim \text{poly}(n_{a_j}, 1/\epsilon_j)$  is Bob's resource in the  $j^{\text{th}}$  round. Thus the efficiency condition requires that there exists some polynomial  $f(x, y)$  such that for all  $\delta > 0$  and for all possible request sequences  $\alpha = (a_1, \dots, a_m)$ ,  $T(\alpha) \leq f(N(\alpha), \frac{1}{\delta})$ . The efficiency requirement imposed on Bob's strategy thus naturally requires that the resource costs of Alice and Bob be polynomial equivalent for the family of tests that the referee can apply.

**Theorem 1.** *Bob has an  $\epsilon$ -simulator of Alice's quantum computer if and only if given the hypothesis testing scenario considered above, there exists a strategy for Bob which jointly achieves indistinguishability and efficiency.*

The proof for this theorem can be found in Appendix A.

We briefly highlight some conclusions for each direction of the “if and only if” statement. Loosely, the “only if” component of the theorem says that, up to polynomial equivalence, an  $\epsilon$ -simulator of a family of quantum circuits  $\mathcal{C}$  is no less useful than having access to  $\mathcal{C}$  itself. This is because any observable consequence of this difference could be used by the referee to decide between  $H_a$  and  $H_b$ . We note that this extends to implying that for all  $\mathcal{C}$  there is an equivalence in the computational power of Bob (given an  $\epsilon$ -simulator of  $\mathcal{C}$ ) and Alice (given  $\mathcal{C}$ ). This can be seen by noting that, despite having unbounded computational power, the referee cannot assign any computational task to the candidate to observe a consequence that will help decide between  $H_a$  and  $H_b$ .

We note the similarity between the efficient indistinguishability condition and the condition of meeting the  $\epsilon$ -simulation definition. We thus note the following interpretation to the “only if” component of Theorem 1: Given a uniform family of quantum circuits  $\mathcal{C}$  and a family of quantum circuits  $\mathcal{C}'$  which is generated from  $\mathcal{C}$  by any set of uniform rules (analogous to the referees rules for choosing the sequence of circuit requests), then a device or algorithm is an  $\epsilon$ -simulator of  $\mathcal{C}$  only if it is an  $\epsilon$ -simulator of  $\mathcal{C}'$ . Thus, the notion of  $\epsilon$ -simulation is robust to modifications of the family of quantum circuits by classical procedures.

In terms of relating the definition of simulation to efficient indistinguishability, the “only if” component of the theorem says that meeting the definition of  $\epsilon$ -simulator is sufficient for achieving efficient indistinguishability while the “if” component of the theorem says that meeting the definition of  $\epsilon$ -simulator is necessary for achieving efficient indistinguishability. Thus, the notion of simulation cannot be weakened any further without compromising efficient indistinguishability.

---

<sup>3</sup> Bob must possess some computational power in order to execute these rules. We will only require that Bob have some small amount of memory (to keep count of the rounds in the protocol) and compute simple arithmetic functions of this counter.

### III. PROBABILITY ESTIMATION

As described in the previous section, a weak simulator produces outcomes sampled from the exact Born rule probability distribution associated with a quantum circuit. The notion of  $\epsilon$ -simulation is a strictly weaker notion of simulation, and so one may not require the full power of a weak simulator to perform  $\epsilon$ -simulation of a quantum circuit. In this paper, we describe an approach to  $\epsilon$ -simulation of quantum circuits based on two components: first, estimating Born rule probabilities for specific outcomes of a quantum circuit to a specified precision, and then using such estimates to construct a simulator. In this section, we describe this first component, coined a *poly-box*. In the next section, we employ such an estimator to construct an  $\epsilon$ -simulator under certain conditions.

#### A. Estimation of Born rule probabilities

Consider the description  $c = \{\rho, U, \mathcal{M}\}$  of some ideal quantum circuit, with  $\rho$  an initial state,  $U = U_L U_{L-1} \cdots U_1$  a sequence of unitary gates, and  $\mathcal{M}$  a set of measurement operators (e.g., projectors). Associated with the measurement  $\mathcal{M} = \{E_x \mid x \in \{0, 1\}^k\}$  is a set of possible measurement outcomes  $x \in \{0, 1\}^k$ . The Born rule gives us the exact quantum predictions associated with observing any particular outcome  $x$ :

$$\mathcal{P}(x) := \text{Tr}(U \rho U^\dagger E_x). \quad (1)$$

Further, probabilities associated with events  $S \subseteq \{0, 1\}^k$  are given by:

$$\mathcal{P}(S) := \sum_{x \in S} \text{Tr}(U \rho U^\dagger E_x) \quad (2)$$

When discussing poly-boxes, we will be interested in the restricted set of events  $S \in \{0, 1, \bullet\}^k$ . We use this notation to indicate the set of all specific outcomes and marginals. Specifically,  $S \in \{0, 1, \bullet\}^k$  is a subset of  $\{0, 1\}^k$  where  $\bullet$  represents either a 0 or a 1 element. For example,  $S = (0, \bullet, 1) := \{(0, 0, 1), (0, 1, 1)\}$ . If  $S$  is a vector with bits  $x_1, \dots, x_{k-m}$  in positions  $i_1, \dots, i_{k-m}$  and a  $\bullet$  in each of the positions  $j_1, \dots, j_m$ ; then  $S$  represents the set of  $2^m$  outcomes where the qubits numbered  $i_1, \dots, i_{k-m}$  produced single qubit measurement outcomes corresponding to  $x_1, \dots, x_{k-m}$  while the remaining qubits (those numbered  $j_1, \dots, j_m$ ) produce either a 0 or a 1 measurement outcome. The probability corresponding to such an event  $S$  is the marginal probability associated with observing the outcome bitstring  $x_1, \dots, x_{k-m}$  on the qubits numbered  $i_1, \dots, i_{k-m}$  marginalised over the qubits  $j_1, \dots, j_m$ .

The task of calculating or even estimating these probabilities efficiently is of great practical interest, but is known to be hard for general quantum circuits even for rather inaccurate levels of estimation. For example, given a circuit  $c_a$  from a family of universal quantum circuits with a Pauli  $Z$  measurement of the first qubit only, deciding if  $\mathcal{P}_a(0) > \frac{2}{3}$  or  $< \frac{1}{3}$  is BQP-complete.

Monte Carlo methods are a common approach to estimating Born rule probabilities that are difficult to calculate directly. Let  $p$  be an unknown parameter we wish to estimate, e.g., a Born rule probability. In a Monte Carlo approach,  $p$  is estimated by observing a number of random variables  $X_1, \dots, X_s$  and computing some function of the outcomes  $\hat{p}_s(X_1, \dots, X_s)$ . In this case,  $\hat{p}_s$  is an estimator of  $p$ .

One way to generate such random variables is to imagine using the quantum system itself. Given access to a quantum circuit  $c_a$ , we may wish to estimate  $p = \mathcal{P}_a(x)$ . In this example, we can construct the estimator  $\hat{p}_s$  by independently running the circuit  $s$  times. On each of the runs  $i = 1, \dots, s$ , we observe if the outcome is  $x$  (in this case,  $X_i = 1$ ) or if the outcome is not  $x$  (in this case,  $X_i = 0$ ). We then define  $\hat{p}_s = \frac{1}{s} \sum_{i=1}^s X_i$ .

Rather than using the quantum system to generate samples, the more interesting case is when one has a classical algorithm that yields an estimator for a class of quantum circuits. There are many such examples; before considering them, we first articulate what properties make for a ‘good’ estimator. This leads us to the notion of a *poly-box*.



### B. The poly-box: generating an additive polynomial precision estimate

We first fix some terminology regarding the precision of an estimator, and how this precision scales with resources. We say that an estimator  $\hat{p}_s$  of  $p$  is *additive  $(\epsilon, \delta)$ -precision* if:

$$\Pr(|p - \hat{p}_s| \geq \epsilon) \leq \delta, \quad \text{additive } (\epsilon, \delta)\text{-precision.} \quad (3)$$

We say that  $\hat{p}_s$  is *multiplicative  $(\epsilon, \delta)$ -precision* if:

$$\Pr(|p - \hat{p}_s| \geq \epsilon p) \leq \delta, \quad \text{multiplicative } (\epsilon, \delta)\text{-precision.} \quad (4)$$

In the case where  $p \leq 1$  is a probability, a multiplicative precision estimator is more accurate than an additive precision estimator.

For any estimator based on the Monte Carlo type of approach described above, there is a polynomial (typically linear) resource cost associated with the number of samples  $s$ . For example, the time taken to compute  $\hat{p}_s$  will scale polynomially in  $s$ . For this reason, we may wish to classify additive/multiplicative  $(\epsilon, \delta)$ -precision estimators by how  $s$  scales with  $1/\epsilon$  and  $1/\delta$ . We say that  $\hat{p}_s$  is an additive *polynomial precision estimator* of  $p$  if there exists a polynomial  $f(x, y)$  such that for all  $\epsilon, \delta > 0$ ,  $\hat{p}_s$  is an additive  $(\epsilon, \delta)$ -precision estimator for all  $s \geq f(\epsilon^{-1}, \log \delta^{-1})$ . We say that  $\hat{p}_s$  is a multiplicative *polynomial precision estimator* of  $p$  if there exists a polynomial  $f(x, y)$  such that for all  $\epsilon, \delta > 0$ ,  $\hat{p}_s$  is a multiplicative  $(\epsilon, \delta)$ -precision estimator for all  $s \geq f(\epsilon^{-1}, \delta^{-1})$ .

A useful class of polynomial additive precision estimators is given by application of the Hoeffding inequality. Suppose  $\hat{p}_1$  resides in some interval  $[a, b]$  and is an unbiased estimator of  $p$ . Let  $\hat{p}_s$  be defined as the average of  $s$  independent observations of  $\hat{p}_1$ . Then, by the Hoeffding inequality, we have:

$$\Pr(|p - \hat{p}_s| \geq \epsilon) \leq 2 \exp\left(\frac{-2s\epsilon^2}{(b-a)^2}\right), \quad (5)$$

for all  $\epsilon > 0$ . We note that for  $s(\epsilon^{-1}, \log \delta^{-1}) \geq \frac{(b-a)^2}{2\epsilon^2} \log(2\delta^{-1})$ ,  $\hat{p}_s$  is an additive  $(\epsilon, \delta)$ -precision estimator of  $p$ . With this observation, we see that additive polynomial precision estimators can always be constructed from unbiased estimators residing in a bounded interval.

Using the Hoeffding inequality, it is easy to show that the Born rule probability estimator  $\hat{p}_s$  constructed in the example above is an additive polynomial precision estimator of  $p$ .

Given a family of quantum circuits  $\mathcal{C}$ , we will be interested in constructing an  $\epsilon$ -simulator of  $\mathcal{C}$  using estimates of Born rule probabilities associated with circuits in  $\mathcal{C}$ . For this purpose it is practical to define a poly-box over  $\mathcal{C}$ .

**Definition 4.** (poly-box). A *poly-box* over a family of quantum circuits  $\mathcal{C} = \{c_a \mid a \in \mathcal{A}^*\}$  with associated family of probability distributions  $\mathbb{P} = \{\mathcal{P}_a \mid a \in \mathcal{A}^*\}$  is a classical algorithm that can be used to compute an estimate from an additive polynomial precision estimator  $\hat{p}_s$  of  $\mathcal{P}_a(S)$  for all  $a \in \mathcal{A}^*$ ,  $s \in \mathbb{N}$ ,  $S \in \{0, 1, \bullet\}^{k_n}$  efficiently in  $s$  and  $n$ .

A poly-box over a family of quantum circuits allows one to estimate probabilities of any particular outcome (and also marginal probabilities) for any circuit in  $\mathcal{C}$  to additive  $(\epsilon, \delta)$ -precision efficiently in the number of qubits,  $\epsilon^{-1}$  and  $\log \delta^{-1}$ .

Whether or not a family of quantum circuits  $\mathcal{C}$  admits a poly-box has bearing on both the complexity of sampling problems and decision problems solvable by  $\mathcal{C}$ , and so we will find that the notion of a poly-box is useful concept. As we will show in Theorems 3 and 2 respectively, a poly-box over  $\mathcal{C}$  is necessary for the existence of an  $\epsilon$ -simulator over  $\mathcal{C}$  and, combined with an additional requirement, it is also sufficient. In addition, we note that if  $\mathcal{C}$  admits a poly-box then all “generalized decision problems” solvable by  $\mathcal{C}$  are solvable within BPP. As an illustrative but unlikely example, suppose there exists a classical poly-box over a universal quantum circuit family  $\mathcal{C}_{\text{univ}}$ . Then, for any instance  $x$  of a decision problem  $L$  in BQP, there is a quantum circuit  $c_a \in \mathcal{C}_{\text{univ}}$  that decides if  $x \in L$  (correctly on at least  $2/3$  of the runs), simply by outputting the decision “ $x \in L$ ” when the first qubit measurement outcome is 1 on a single run of  $c_a$  and conversely, outputting the decision “ $x \notin L$ ” when the first qubit measurement outcome is 0. We note that, in order to decide if  $x \in L$  one does not need the full power of an  $\epsilon$ -simulator over  $\mathcal{C}_{\text{univ}}$ . In fact it is sufficient to only have access to the poly-box over  $\mathcal{C}_{\text{univ}}$ . Given a poly-box over  $\mathcal{C}_{\text{univ}}$ , one can request

an  $(\epsilon, \delta)$ -precision estimate  $\hat{p}$  for the probability  $p$  that the sampled outcome from  $c_a$  is in  $S = (1, \bullet, \dots, \bullet)$ . For  $\epsilon < 1/6$  and  $\delta < 1/3$ , one may decide “ $x \in L$ ” if  $\hat{p} \geq 1/2$  and “ $x \notin L$ ” otherwise. This will result in the correct decision with probability  $\geq 2/3$  as required. A poly-box over  $\mathcal{C}$  offers the freedom to choose any  $S \in \{0, 1, \bullet\}^n$  which can in general be used to define a broader class of decision problems. Of course in the case of  $\mathcal{C}_{\text{univ}}$ , this freedom cannot be exploited because for every choice of  $a$  and  $S \neq (1, \bullet, \dots, \bullet)$ , there is a alternative easily computable choice of  $a'$  such that the probability that a run of  $c_{a'} \in \mathcal{C}_{\text{univ}}$  results in an outcome in  $(1, \bullet, \dots, \bullet)$  is identical to probability that a run of  $c_a \in \mathcal{C}_{\text{univ}}$  results in an outcome in  $S$ .

### C. Examples of poly-boxes

#### 1. Poly-boxes from quasiprobability representations

There are a number of known algorithms for constructing poly-boxes over certain non-universal families of quantum circuits [2, 20, 21, 23, 32]. In particular, we focus on the algorithm presented in Ref. [23], which can be used to construct a poly-box over any family of quantum circuits  $\mathcal{C}$  where the *negativity* of quantum circuits grows at most polynomially in the circuit size. We refer the interested reader to Ref. [23] for a definition of the negativity of a quantum circuit, but note that this quantity depends on the initial state, sequence of unitaries and the final POVM measurement that defines the circuit. For general quantum circuits, the negativity can grow exponentially in both the number of qudits and the depth of the circuit.

A key application of this approach is to Clifford circuits. In odd dimensions, stabilizer states, Clifford gates, and measurements in the computational basis do not contribute to the negativity of a Monte Carlo based estimator. Non-Clifford but efficiently describable operators, including product state preparations or measurements that are not stabilizer states, or non-Clifford gates such as the  $T$  gate, may contribute to the negativity of the circuit. Nonetheless, these non-Clifford operations can be accommodated within the poly-box provided that the total negativity is bounded polynomially. In addition, a poly-box exists for such circuits even in the case where the negativity of the initial state, *or* of the measurement, is exponential [23].

#### 2. A poly-box over $\mathcal{C}_{\text{PROD}}$

As a nontrivial example of a class of Clifford circuits for which there exists a poly-box, consider the family of circuits  $\mathcal{C}_{\text{PROD}}$  for which the  $n$ -qubit input state  $\rho$  is an arbitrary product state<sup>4</sup> (with potentially exponential negativity [23]), transformations are non-adaptive Clifford unitary gates, and measurements are of  $k \leq n$  qubits in the computational basis. Such a circuit family has been considered by Jozsa and Van den Nest [34], where it was referred to as INPROD, OUTMANY, NON-ADAPT. This circuit family will be useful in our hardness results presented in Section V. Aaronson and Gottesmann [2] provide the essential details of a poly-box for this family of circuits; for completeness, we present an explicit poly-box for  $\mathcal{C}_{\text{PROD}}$  in the following Lemma.

**Lemma 1.** *A classical poly-box exists for the Clifford circuit family  $\mathcal{C}_{\text{PROD}}$ .*

*Proof.* Give an arbitrary circuit  $c = \{\rho, U, \mathcal{M}\} \in \mathcal{C}_{\text{PROD}}$  and an event  $S \in \{0, 1, \bullet\}^n$  we construct an estimator  $\hat{p}_s$  of the probability  $\mathcal{P}(S)$  as follows:

1. Let  $\Pi = \otimes_{i=1}^n \Pi_i$  be the projector corresponding to  $S$ . Here, we set:

$$\Pi_i = \begin{cases} \frac{I+Z}{2} & \text{if the } i^{\text{th}} \text{ entry of } S \text{ is } 0 \\ \frac{I-Z}{2} & \text{if the } i^{\text{th}} \text{ entry of } S \text{ is } 1 \\ I & \text{if the } i^{\text{th}} \text{ entry of } S \text{ is } \bullet \end{cases} \quad (6)$$

---

<sup>4</sup> As an additional technical requirement, we impose that the input product state is generated from  $|0\rangle^{\otimes n}$  by the application of polynomially many gates from a universal single qubit gate set with algebraic entries.

2. For each  $i$  where the  $i^{th}$  entry of  $S$  is not  $\bullet$ ,  $\Pi_i = \frac{I \pm Z}{2}$ . In these cases, define a local Pauli operator  $P_i$  by sampling either  $I$  or  $\pm Z$  with equal probability. For each  $i$  where the  $i^{th}$  entry of  $S$  is a  $\bullet$ , we deterministically set  $P_i = I$ .
3. We construct the  $n$ -qubit Pauli operator  $P := \otimes_{i=1}^n P_i$ , (including its sign  $\pm$ ).
4. Using the Gottesmann-Knill theorem [2], we compute the Pauli operator  $P' = \otimes_{i=1}^n P'_i := U^\dagger P U$ .
5. We compute the single sample estimate  $\hat{p}_1$  using the equation:

$$\hat{p}_1 := \text{tr}(\rho P') = \prod_{i=1}^n \text{tr}(\rho_i P'_i). \quad (7)$$

6. We compute the estimator  $\hat{p}_s$  by computing  $s$  independent single sample estimates and taking their average.

It is straightforward to show that the expectation value of  $\hat{p}_s$  is the target quantum probability  $p := \mathcal{P}(S)$ . Further, the single sample estimates are bounded in the interval  $[-1, 1]$ . Hence, by the Hoeffding inequality,

$$\Pr(|\hat{p}_s - p| \geq \epsilon) \leq 2e^{-\frac{s\epsilon^2}{2}}. \quad (8)$$

This algorithm can be executed efficiently in  $s$  and in  $n$  and produces additive polynomial precision estimates of  $\mathcal{P}(S)$  for any circuit  $c \in \mathcal{C}_{\text{PROD}}$  and any  $S \in \{0, 1, \bullet\}^n$  and is thus a poly-box.  $\square$

### 3. A poly-box over $\mathcal{C}_{\text{IQP}}$

As an additional example, we note that  $\mathcal{C}_{\text{IQP}}$ , the family of Instantaneous Quantum Polynomial-time (IQP) quantum circuits [8, 9, 35, 36] that consist of computational basis preparation and measurements with all gates diagonal in the  $X$  basis admits a poly-box. One can construct such a poly-box over  $\mathcal{C}_{\text{IQP}}$  by noting that Proposition 5 from Ref. [29] gives a closed form expression for all Born rule probabilities and marginals of these circuits [37]. This expression:

$$\Pr(X \in S) = \mathbb{E}_{r \in \text{span}\{\vec{e}_i \mid i \in \{i_1, \dots, i_k\}\}} \left[ (-1)^{r \cdot s} \alpha\left(P_r, \frac{\pi}{4}\right) \right] \quad (9)$$

is an expectation value over  $2^k$  vectors in  $\mathbb{Z}_2^n$  where:

- $\{i_1, \dots, i_k\}$  are the set of indices where the entries of  $S$  are in  $\{0, 1\}$ ;
- $s \in \mathbb{Z}_2^n$  is defined by  $s_i = S_i$  when  $i \in \{i_1, \dots, i_k\}$  and  $s_i = 0$  otherwise;
- $P_r$  is the *affinification* of the  $m \times n$  binary matrix  $P$  which defines a Hamiltonian of the IQP circuit constructed from Pauli  $X$  operators according to  $H_P := \sum_{i=1}^m \otimes_{j=1}^n X^{P_{ij}}$ ;
- $\alpha(P, \theta)$  is the normalized version of the weight enumerator polynomial (evaluated at  $e^{-2i\theta}$ ) of the code generated by the columns of  $P$ .

We note that this is an expectation over exponentially many terms which have their real part bounded in the interval  $[-1, 1]$ . Further, for each  $r$ , the quantity  $\alpha\left(P_r, \frac{\pi}{4}\right)$  can be evaluated efficiently using Vertigan's algorithm [38] and Ref. [29]. As such, one can construct an additive polynomial precision estimator for all Born rule probabilities and marginals simply by evaluating the expression:

$$\hat{p}_1 = \text{Re} \left[ (-1)^{r \cdot s} \alpha\left(P_r, \frac{\pi}{4}\right) \right] \quad (10)$$

for polynomially many independent uniformly random chosen  $r \in \text{span}\{\vec{e}_i \mid i \in \{i_1, \dots, i_k\}\}$  and computing the average over all choices. This can be shown to produce a poly-box by application of the Hoeffding inequality.

#### IV. FROM ESTIMATION TO SIMULATION

With a poly-box, one can efficiently estimate Born rule probabilities of outcomes of a quantum circuit with additive precision. However, a poly-box alone is not a simulator. We illustrate this using a simple but somewhat contrived example, wherein an encoding into a large number of qubits is used to obscure the computational power of sampling from the poly-box [39].

##### A. A poly-box is not sufficient for $\epsilon$ -simulation

Define a family of quantum circuits  $\mathcal{C}_e$  based on using a universal quantum computer as an oracle as follows:

1. take as input a quantum circuit description  $a \in \mathcal{A}^*$  (this is a description of some quantum circuit with  $n$  qubits);
2. call the oracle to output a sample outcome from this quantum circuit. Label the first bit of the outcome by  $X$ ;
3. sample an  $n$ -bit string  $Y \in \{0, 1\}^n$  uniformly at random;
4. output  $Z = (X \oplus \text{Par}(Y), Y) \in \{0, 1\}^{n+1}$ , where  $\text{Par}(Y)$  is the parity function on the input bit-string  $Y$ .

We note that  $\mathcal{C}_e$  cannot admit an  $\epsilon$ -simulator unless  $\text{BQP} \subseteq \text{BPP}$ , since simple classical post processing reduces the  $\epsilon$ -simulator over  $\mathcal{C}_e$  to an  $\epsilon$ -simulator over universal quantum circuits restricted to a single qubit measurement.

We now show that  $\mathcal{C}_e$  admits a poly-box:

1. take as input  $a \in \mathcal{A}^*$ ,  $\epsilon, \delta > 0$  and  $S \in \{0, 1, \bullet\}^{n+1}$ . Our poly-box will output probability estimates that are deterministically within  $\epsilon$  of the target probabilities and hence we can set  $\delta = 0$ ;
2. if  $S$  specifies a marginal probability i.e.  $k < n+1$ , then the poly-box outputs the estimate  $2^{-k}$  (where  $k$  is the number of non-marginalized bits in  $S$ ); otherwise,
  - (a) small  $\epsilon$  case: if  $\epsilon < 1/2^n$ , explicitly compute the quantum probability  $p := \Pr(X = 1)$ ;
  - (b) large  $\epsilon$  case: if  $\epsilon \geq 1/2^n$ , output the probability  $2^{-(n+1)}$  as a guess.

This algorithm is not only a poly-box over  $\mathcal{C}_e$  but it in fact outputs probability estimates that have exponentially small precision.

**Lemma 2.** *For all  $a \in \mathcal{A}^*$ ,  $\epsilon > 0$  and  $S \in \{0, 1, \bullet\}^{n+1}$ , the above poly-box can output estimates within  $\epsilon$  additive error of the target probability using  $O(\text{poly}(n, 1/\epsilon))$  resources. Further, the absolute difference between estimate and target probabilities will be  $\leq \min\{2^{-(n+1)}, \epsilon\}$ .*

*Proof.* We note that the resource cost of this algorithm is  $O(\text{poly}(n, 1/\epsilon))$ . Since in the case of small  $\epsilon$  it is  $O(\text{poly}(2^n)) \subseteq O(\text{poly}(1/\epsilon))$  and in the case of large  $\epsilon$  it is  $O(n)$ .

We now consider the machine's precision by considering the case with no marginalization and the case with marginalization separately. We restrict the below discussion to the large  $\epsilon$  case as the estimates are exact in the alternate case.

Let  $z = (z_0, \dots, z_n) \in \{0, 1\}^{n+1}$  be fixed and define  $z' := (z_1, \dots, z_n)$ . Then,

$$\Pr(Z = z) = \Pr(Z_0 = z_0 \mid Y = z')\Pr(Y = z') = \Pr(X = z_0 \oplus \text{Par}(z'))2^{-n}. \quad (11)$$

So for  $S = z$  (i.e. no marginalization), we have an error given by  $\max_{r \in \{p, 1-p\}} |2^{-(n+1)} - \frac{r}{2^n}| \leq 2^{-(n+1)}$ .

For the case where  $S_i = \bullet$  (i.e. there is marginalization over the  $i^{\text{th}}$  bit only and  $k = n$ ), we note that the quantum marginal probability  $p(S)$  is given exactly by:

$$p(S) = \sum_{z_i=0}^1 \Pr(Z = z) = \sum_{z_i=0}^1 \Pr(X = z_0 \oplus \text{Par}(z'))2^{-n} = p2^{-n} + (1-p)2^{-n} = 2^{-k}, \quad (12)$$

where  $z_j := S_j$  for  $j \neq i$ . This argument can be seen to imply that for all  $k < n+1$ , the quantum probability is exactly  $2^{-k}$ . Thus, in the worst case (no marginalization and  $\epsilon \geq 2^{-n}$ ), the error is  $\leq 2^{-(n+1)}$ .  $\square$

This contrived example clearly demonstrates that the existence of a poly-box for a class of quantum circuits is not sufficient for  $\epsilon$ -simulation. In the following, we highlight the role of the *sparsity* of the output distribution in providing, together with a poly-box, a sufficient condition for  $\epsilon$ -simulation.

## B. Sparsity and sampling

Despite the fact that in general the existence of a poly-box for some family  $\mathcal{C}$  does not imply the existence of an  $\epsilon$ -simulator for  $\mathcal{C}$ , for some quantum circuit families, a poly-box suffices. Here, we show that one can construct an  $\epsilon$ -simulator for a family of quantum circuits  $\mathcal{C}$  provided that there exists a poly-box over  $\mathcal{C}$  and that the family of probability distributions corresponding to  $\mathcal{C}$  satisfy an additional constraint on the *sparsity* of possible outcomes. We begin by reviewing several results from Schwarz and Van den Nest [40] regarding sparse distributions. In Ref. [40], they define the following property of discrete probability distributions:

**Definition 5.** ( $\epsilon$ -approximately  $t$ -sparse). A discrete probability distribution is  $t$ -sparse if at most  $t$  outcomes have a non-zero probability of occurring. A discrete probability distribution is  $\epsilon$ -approximately  $t$ -sparse if it has a  $L_1$  distance less than or equal to  $\epsilon$  from some  $t$ -sparse distribution.

The lemma below is a (possibly slightly weakened) restatement of Theorem 11 from Ref. [40].

**Lemma 3.** (Theorem 11 of Ref. [40]). Let  $\mathcal{P}$  be a distribution on  $\{0,1\}^k$  that satisfies the following conditions:

1.  $\mathcal{P}$  is promised to be  $\epsilon$ -approximately  $t$ -sparse, where  $\epsilon \leq 1/6$ ;
2. For all  $S \in \{0,1,\bullet\}^k$ , there exists an  $(s,k)$ -efficient randomized classical algorithm for sampling from  $\hat{p}_s$ , an additive polynomial estimator of  $\mathcal{P}(S)$ .

Then it is possible to classically sample from a probability distribution  $\mathcal{P}' \in B(\mathcal{P}, 12\epsilon + \delta)$  efficiently in  $k$ ,  $t$ ,  $\epsilon^{-1}$  and  $\log \delta^{-1}$ .

We note that for every discrete probability distribution  $\mathcal{P}$ , there is some unique minimal function  $t(\epsilon)$  such that for all  $\epsilon \geq 0$ ,  $\mathcal{P}$  is  $\epsilon$ -approximately  $t$ -sparse. We note that if this function is upper-bounded by a polynomial in  $\epsilon^{-1}$ , then a randomized classical algorithm for sampling from estimators of  $\mathcal{P}(S)$  can be extended to a randomized classical algorithm for sampling from some probability distribution  $\mathcal{P}' \in B(\mathcal{P}, \epsilon)$  efficiently in  $\epsilon^{-1}$ . This fact motivates the following definition:

**Definition 6.** (poly-sparse) Let  $\mathcal{P}$  be a discrete probability distribution. We say that  $\mathcal{P}$  is poly-sparse if there exists a polynomial  $P(x)$  such that for all  $\epsilon > 0$ ,  $\mathcal{P}$  is  $\epsilon$ -approximately  $t$ -sparse whenever  $t \geq P(\frac{1}{\epsilon})$ .

Let  $\mathbb{P}$  be a family of probability distributions with  $\mathcal{P}_a \in \mathbb{P}$  a distribution over  $\{0,1\}^{k_a}$ . We say that  $\mathbb{P}$  is poly-sparse if there exists a polynomial  $P(x)$  such that for all  $\epsilon > 0$  and  $a \in \mathcal{A}^*$ ,  $\mathcal{P}_a$  is  $\epsilon$ -approximately  $t$ -sparse whenever  $t \geq P(k_a/\epsilon)$ .

The notion of poly-sparse is related to the notion of smooth max entropy  $H_{max}^\epsilon$ . In particular,  $\mathbb{P}$  is poly-sparse iff there exists a number  $d \in \mathbb{N}$  for every  $\mathcal{P} \in \mathbb{P}$  with domain cardinality  $2^n$ , such that we have:

$$2^{H_{max}^\epsilon(\mathcal{P})} \leq \left(\frac{n}{\epsilon}\right)^d. \quad (13)$$

## C. Conditions for $\epsilon$ -simulation

With this notion of output distributions that are poly-sparse, we are in a position to state our main theorem of this section:

**Theorem 2.** *Let  $\mathcal{C}$  be a family of quantum circuits with a corresponding family of probability distributions  $\mathbb{P}$ . Suppose there exists an efficient poly-box over  $\mathcal{C}$ , and that  $\mathbb{P}$  is poly-sparse. Then, there exists an  $\epsilon$ -simulator of  $\mathcal{C}$ .*

*Proof.* Let  $a \in \mathcal{A}^*$  and  $\epsilon > 0$  be arbitrary. Then there exist  $t = t(a, \epsilon)$  such that  $\mathcal{P}_a$  is  $\epsilon$ -approximately  $t$ -sparse. Further, due to the existence of the efficient classical poly-box over  $\mathcal{C}$ , for all  $S \in \{0, 1, \bullet\}^{k_a}$ , there exists an  $(s, k_a)$ -efficient randomized classical algorithm for sampling from an additive polynomial estimator of  $\mathcal{P}_a(S)$ . Thus by Lemma 2, it is possible to classically sample from a probability distribution  $\mathcal{P}_a^\epsilon \in B(\mathcal{P}_a, \epsilon)$  efficiently in  $\epsilon^{-1}, t$  and  $k_a$ . We note that here we have removed the dependence on  $\delta$  since we can make  $\delta \leq \epsilon$  whilst remaining efficient in  $\epsilon^{-1}, t$  and  $k_a$ . Finally, since poly-sparsity guarantees the existence of a  $t(a, \epsilon)$  that can be upper-bounded by a polynomial in  $\frac{k_a}{\epsilon}$ , we arrive at the desired result.  $\square$

As an example, consider families of quantum circuits  $\mathcal{C}$  where each circuit of size  $n$  can only produce outcomes from some set of size at most  $\text{poly}(n)$ . Then  $\mathcal{C}$  is poly-sparse (even if the output distributions are uniform over the  $\text{poly}(n)$  sized support). Hence, if  $\mathcal{C}$  also admits a poly-box, then by Thm. 2 one can with high probability repeatedly sample from this space of  $\text{poly}(n)$  outcomes hidden within an exponentially large space of bitstrings.

We have shown that having a poly-box and a poly-sparsity guarantee for a family of quantum circuits gives us a  $\epsilon$ -simulator. We emphasise that this approach allows for the  $\epsilon$ -simulation of families of quantum circuits for which no known weak simulation method exists. Specifically, by combining the results from Ref. [23] with the above theorem, we conclude that any family of quantum circuits  $\mathcal{C}$  that both:

1. has negativity that is polynomially bounded in circuit size; and
2. has an associated family of probability distributions which is poly-sparse;

can be  $\epsilon$ -simulated.

We emphasise that the proof of this Theorem is constructive, and allows for new simulation results for families of quantum circuits for which it was not previously known if they were efficiently simulable. As an example, our results can be straightforwardly used to show that Clifford circuits with sparse outcome distributions and with small amounts of local unitary (non-Clifford) noise, as described in Ref. [21], are  $\epsilon$ -simulable.

#### D. Necessity of a poly-box

We have shown that the existence of a poly-box over a poly-sparse family of quantum circuits is sufficient for  $\epsilon$ -simulation. We point out that the existence of a poly-box is also a necessary condition for  $\epsilon$ -simulation.

**Theorem 3.** *If  $\mathcal{C}$  is a family of quantum circuits that does not admit a poly-box algorithm, then  $\mathcal{C}$  is not  $\epsilon$ -simulable.*

*Proof.* We note that given an  $\epsilon$ -simulator of  $\mathcal{C}$ , a poly-box over  $\mathcal{C}$  can be constructed in the obvious way simply by observing the frequency with which the  $\epsilon$ -simulator outputs outcomes in  $S$  and using this observed frequency as the estimator for  $\mathcal{P}(S)$ .  $\square$

In summary, the results of Thms. 2 and 3 imply that in order to construct an  $\epsilon$ -simulator of any particular family of quantum circuits, it is necessary to construct a poly-box and further, if the family is poly-sparse, this is also sufficient. In Sec. IV A, we also showed that there exists a somewhat artificial family of quantum circuits  $\mathcal{C}_e$  with respect to which a poly-box is insufficient for  $\epsilon$ -simulation. In the next section, we show that this phenomenon also occurs with much more natural families of quantum circuits.

#### V. HARDNESS RESULTS

In the previous section, we have shown that one can construct an  $\epsilon$ -simulator for a family of quantum circuits  $\mathcal{C}$  given a poly-box for this family together with a promise of poly-sparsity of the corresponding probability distribution. We also discussed a contrived construction of a family of quantum circuits that admits a poly-box but is not  $\epsilon$ -simulable. In this section, we provide strong evidence (dependent only on

standard complexity assumptions and a variant of the now somewhat commonly used [7, 9, 10, 12, 16, 30] “average case hardness” conjecture) that the poly-sparsity condition is necessary even for natural families of quantum circuits. One such family has already been identified by noting that  $\mathcal{C}_{\text{IQP}}$  admits a poly-box and is hard to  $\epsilon$ -simulate [9]. Here, we also show the hardness of  $\epsilon$ -simulating the non-poly-sparse Clifford circuit family  $\mathcal{C}_{\text{PROD}}$  (defined in Sec. III). These results mean that there exist quantum circuits where the probability of individual outcomes and marginals can be efficiently estimated, but due to the vast number of relevant outcomes the system cannot be efficiently simulated.

Our hardness result for classical  $\epsilon$ -simulation of  $\mathcal{C}_{\text{PROD}}$  closely follows the structure of several similar results, and in particular that of the IQP circuits result of Ref. [9]. Despite the existence of a poly-box over  $\mathcal{C}_{\text{PROD}}$ , we show that there cannot exist a classical  $\epsilon$ -simulator of this family unless the polynomial hierarchy collapses to the third level. We note that the hardness of *exact* weak simulation of  $\mathcal{C}_{\text{PROD}}$  was shown in Ref. [34]. In contrast here we show the hardness of  $\epsilon$ -simulation for this family. Our proof relies on a conjecture regarding the hardness of estimating Born rule probabilities to within a small multiplicative factor for a substantial fraction of randomly chosen circuits from  $\mathcal{C}_{\text{PROD}}$ . This average case hardness conjecture is a strengthening of the worst case hardness of multiplicative precision estimation of probabilities associated with circuits from  $\mathcal{C}_{\text{PROD}}$ .

The hardness of  $\epsilon$ -simulating  $\mathcal{C}_{\text{PROD}}$  circuits is shown by first noting that the existence of a classical  $\epsilon$ -simulator implies, via the application of the Stockmeyer approximate counting algorithm [41], the existence of an algorithm (in the third level of the PH) for estimating the probabilities associated with the output distribution of the  $\epsilon$ -simulator to within a multiplicative factor. These estimates can then be related to estimates of the exact quantum probabilities by noting two points:

1. that the deviation between the  $\epsilon$ -simulator’s probability of outputting a particular outcome and that of the exact quantum probability will be exponentially small for the vast majority of outcomes. We show this fact using Markov’s inequality.
2. that a significant portion of outcomes associated with randomly chosen circuit in  $\mathcal{C}_{\text{PROD}}$  must have outcome probabilities larger than a constant fraction of  $2^{-n}$ . We show this property using our proof that these circuits anti-concentrate.

These observations are combined to show that if there exists an  $\epsilon$ -simulator of  $\mathcal{C}_{\text{PROD}}$ , then there exists a classical algorithm (in the third level of the PH) that can estimate Born rule outcome probabilities to within a multiplicative factor for almost 50% of circuits sampled from  $\mathcal{C}_{\text{PROD}}$ . This is in contradiction with Conjecture 1 thus implying that an  $\epsilon$ -simulator does not exist.

#### A. Conjecture regarding average case hardness

We begin by stating our conjecture that multiplicative precision estimation of  $\mathcal{C}_{\text{PROD}}$  is  $\#P$ -hard in the average case.

**Conjecture 1.** *There exist an input product state  $\rho$  over  $n$  qubits such that given a uniformly random Clifford unitary  $U$  acting on  $n$  qubits, estimating  $p := \text{tr}(U\rho U^\dagger|0\rangle\langle 0|)$  to within a multiplicative error of  $1/\text{poly}(n)$  for more than 50% of the sampled Clifford unitaries is  $\#P$ -hard.*

We note that this average case hardness conjecture has an analogous worst case hardness version<sup>5</sup>. The worst case hardness can be proven to hold by an argument essentially identical to the proof of Theorem 5.1 in Ref. [16]. We omit the proof here but note that this proof relies on three key facts:

1. that estimating Born rule probabilities for universal (indeed even IQP) circuits to within any multiplicative factor greater than unity is  $\#P$ -hard [42];
2. for gate sets with algebraic entries, all non-zero output probabilities are lower bounded by some inverse exponential [43];

---

<sup>5</sup> This is the same statement as per Conjecture 1 but with “more than 50%” replaced by “100%”.

3. that  $\mathcal{C}_{\text{PROD}}$  circuits with post-selection (or adaptivity) are universal for quantum computation [34].

We emphasize that similar conjectures are commonly used in related hardness proofs, such as Refs. [7, 9, 10, 12, 16, 30].

### B. Anti-concentration of outcomes for $\mathcal{C}_{\text{PROD}}$

Next, we prove that Clifford circuits chosen uniformly at random from the family  $\mathcal{C}_{\text{PROD}}$  satisfy an anti-concentration property. This result can be shown to imply that such circuits are not going to satisfy the poly-sparsity condition.

**Lemma 4.** *For each  $n \in \mathbb{N}$ , let  $c_n \in \mathcal{C}_{\text{PROD}}$  be an  $n$ -qubit Clifford circuit chosen by fixing an arbitrary  $n$  qubit input state  $\rho$ , applying a uniformly random Clifford unitary  $U$  acting on  $n$  qubits and doing a computational basis measurement on all qubits. Then for all  $\alpha \in (0, 1)$  and for any fixed choice of  $x \in \{0, 1\}^n$ :*

$$Pr_U \left( p_x \geq \frac{\alpha}{2^n} \right) > \frac{(1 - \alpha)^2}{2} \quad (14)$$

where  $p_x := \text{tr}(U\rho U^\dagger |x\rangle\langle x|)$  is the Born rule probability for the outcome  $x$ .

*Proof.* We use the unitary 2-design property of the Clifford group.

$$\mathbb{E}(p_x) = \text{tr}(\mathbb{E}(U\rho U^\dagger) |x\rangle\langle x|) = \text{tr}(\mathbb{1}/2^n |x\rangle\langle x|) = \frac{1}{2^n} \quad (15)$$

$$\mathbb{E}(p_x^2) = \text{tr}(\mathbb{E}(U \otimes U(\rho \otimes \rho) U^\dagger \otimes U^\dagger) |x\rangle\langle x| \otimes |x\rangle\langle x|) \quad (16)$$

$$= \frac{\text{tr}(P_{\text{Sym}}(\rho \otimes \rho))}{\text{tr} P_{\text{Sym}}} \text{tr}(P_{\text{Sym}} |x\rangle\langle x| \otimes |x\rangle\langle x|) \quad (17)$$

$$= \frac{2\text{tr}(P_{\text{Sym}}(\rho \otimes \rho))}{2^n(2^n + 1)} \quad (18)$$

$$= \frac{(\text{Tr}(\rho^2) + (\text{Tr}\rho)^2)}{2^n(2^n + 1)} \leq \frac{2}{2^n(2^n + 1)}, \quad (19)$$

where  $P_{\text{Sym}} = \frac{1}{2}(\mathbb{1} + \text{SWAP})$  is the projection onto the symmetric subspace of  $\mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ . We use the Paley-Zygmund inequality, which states that for a non-negative random variable  $R$  with finite variance, and for any  $\alpha \in (0, 1)$ :

$$Pr(R \geq \alpha \mathbb{E}[R]) \geq (1 - \alpha)^2 \frac{\mathbb{E}^2[R]}{\mathbb{E}[R^2]}, \quad (\text{Paley-Zygmund inequality}) \quad (20)$$

Application of this inequality with Eqs. (15-19) then gives the desired result.  $\square$

### C. Hardness theorem

We are now in a position to prove our main theorem:

**Theorem 4.** *If there exists an  $\epsilon$ -simulator of  $\mathcal{C}_{\text{PROD}}$  and Conjecture 1 holds, then the polynomial hierarchy collapses to the third level.*

*Proof.* Assuming there exists an  $\epsilon$ -simulator of  $\mathcal{C}_{\text{PROD}}$ , we can treat the  $\epsilon$ -simulator as a deterministic Turing machine with a random input. Let  $\mathcal{T}$  be the Turing machine that takes as an input  $\epsilon > 0$  (representing the  $L_1$  error required),  $r \in \{0, 1\}^{\text{poly}(n/\epsilon)}$  (representing the random bit-string) and  $d_c \in \mathcal{A}^{\text{poly}(n)}$  (representing an efficient description of an  $n$  qubit circuit  $c \in \mathcal{C}_{\text{PROD}}$ ) and outputs an outcome  $X^\epsilon \in \{0, 1\}^k$  with the correct



statistics (over uniformly random  $r$  inputs) up to  $\epsilon$  in  $L_1$  distance in time  $\text{poly}(n, 1/\epsilon)$ . That is, the output satisfies:

$$\|p - p^\epsilon\|_1 := \sum_{x \in \{0,1\}^k} |p_x - p_x^\epsilon| \leq \epsilon \quad (21)$$

where  $p_x := \Pr(X = x)$  is the probability of observing outcome  $x$  on a single run of the quantum circuit  $c$  and  $p_x^\epsilon := \Pr_{r \sim \text{unif}}(X^\epsilon = x)$  is the probability of observing outcome  $x$  on a single run of the Turing machine  $\mathcal{T}$  for a uniformly distributed random  $r$  and fixed  $\epsilon, d_c$  inputs.

We now note that the problem of computing the proportion  $p_x^\epsilon$  of bit-strings  $r$  that result in  $\mathcal{T}(\epsilon, r, d_c) = x$  is a problem in  $\#P$ . Thus, the Stockmeyer algorithm gives us a means of estimating  $p_x^\epsilon$  to within a multiplicative error in the complexity class  $BPP^{NP}$ .

More precisely, there exists an algorithm in  $\Delta_3^P$  (this is  $P^{NP^{NP}}$ ) which will output an estimate  $\tilde{p}_x^\epsilon$  such that:

$$|p_x^\epsilon - \tilde{p}_x^\epsilon| \leq \frac{p_x^\epsilon}{\text{poly}(n)} \quad (22)$$

Thus we have that for all  $c$  and for all  $x$ :

$$\begin{aligned} |p_x - \tilde{p}_x^\epsilon| &\leq |p_x - p_x^\epsilon| + |p_x^\epsilon - \tilde{p}_x^\epsilon| \\ &\leq |p_x - p_x^\epsilon| + \frac{p_x^\epsilon}{\text{poly}(n)} \\ &\leq |p_x - p_x^\epsilon| + \frac{p_x + |p_x - p_x^\epsilon|}{\text{poly}(n)} \\ &= |p_x - p_x^\epsilon| \left(1 + \frac{1}{\text{poly}(n)}\right) + \frac{p_x}{\text{poly}(n)} \end{aligned} \quad (23)$$

We note that the expectation value of  $|p_x - p_x^\epsilon|$  over random choice of  $x \sim \text{unif}(\{0,1\}^k)$  is upper-bounded by  $2^{-n}\epsilon$ . That is:

$$\begin{aligned} \mathbb{E}_x[|p_x - p_x^\epsilon|] &= \frac{1}{2^k} \sum_x |p_x - p_x^\epsilon| \\ &= \frac{1}{2^k} \|p - p^\epsilon\|_1 \\ &\leq \frac{\epsilon}{2^k} \end{aligned}$$

Restricting our attention to circuits in  $\mathcal{C}_{\text{PROD}}$  where all of the qubits are measured i.e.  $k = n$ , we have:

$$\mathbb{E}_x[|p_x - p_x^\epsilon|] \leq \frac{\epsilon}{2^n} \quad (24)$$

We apply Markov's inequality, which states that for  $R$  a non-negative random variable and  $\gamma > 0$ :

$$\Pr\left(R \geq \frac{\mathbb{E}[R]}{\gamma}\right) \leq \gamma, \quad (\text{Markov's inequality}) \quad (25)$$

we have that for all  $\beta > 0$ :

$$\Pr_x\left(|p_x - p_x^\epsilon| \geq \frac{\mathbb{E}_x[|p_x - p_x^\epsilon|]}{\beta}\right) \leq \beta \quad (26)$$

That is:

$$\Pr_x\left(|p_x - p_x^\epsilon| < \frac{\epsilon}{\beta 2^n}\right) > (1 - \beta) \quad (27)$$

Applying this to the upper bound in eq. 23, we find that for all  $\beta > 0$ :

$$\Pr_x \left( |p_x - \tilde{p}_x^\epsilon| < \frac{\epsilon}{\beta 2^n} \left( 1 + \frac{1}{\text{poly}(n)} \right) + \frac{p_x}{\text{poly}(n)} \right) > (1 - \beta) \quad (28)$$

For any fixed choices of  $\alpha \in (0, 1)$ ,  $\beta, \epsilon > 0$ , let us define the following events:

- Event A:  $\frac{p_x}{\alpha} \geq \frac{1}{2^n}$
- Event B:  $|p_x - \tilde{p}_x^\epsilon| < \frac{\epsilon}{\beta 2^n} \left( 1 + \frac{1}{\text{poly}(n)} \right) + \frac{p_x}{\text{poly}(n)}$ .

By eq. 14, we have  $\Pr_U(A) > \frac{(1-\alpha)^2}{2}$  and by eq. 28, we have  $\Pr_x(B) > (1 - \beta)$ . Recall that the intersection bound tells us that  $\Pr(A \cap B) \geq \max\{0, \Pr(A) + \Pr(B) - 1\}$  for events  $A$  and  $B$ . Thus, we have  $\Pr(A \cap B) \geq \frac{(1-\alpha)^2 - 2\beta}{2}$ . This immediately implies the following:

$$\Pr_{U,x} \left( |p_x - \tilde{p}_x^\epsilon| < \frac{\epsilon p_x}{\alpha \beta} \left( 1 + \frac{1}{\text{poly}(n)} \right) + \frac{p_x}{\text{poly}(n)} \right) > \frac{(1-\alpha)^2 - 2\beta}{2} \quad (29)$$

This can be further simplified by incorporating the randomness over  $x$  into the uniform randomness over the Clifford unitaries. Specifically, let  $y \in \{0, 1\}^n$  be arbitrarily fixed. Further, let  $U_x := \otimes_{i=1}^n X^{x_i}$ . Then, noting that for all  $n$  qubit Cliffords  $V$ :

$$\Pr_{U, U_x}(U_x U = V) = \Pr_{U, U_x}(U = U_x V) \quad (30)$$

$$= \Pr_U(U = V) \quad (31)$$

$$= \Pr_U(U_y U = V) \quad (32)$$

where probabilities over  $U_x$  are chosen uniformly over all  $x \in \{0, 1\}^n$ . Applying this to eq. 29 we find that for all  $y \in \{0, 1\}^n$  and for all  $n$  qubit product states  $\rho$ :

$$\Pr_U \left( |p_y - \tilde{p}_y^\epsilon| < \frac{\epsilon p_y}{\alpha \beta} \left( 1 + \frac{1}{\text{poly}(n)} \right) + \frac{p_y}{\text{poly}(n)} \right) > \frac{(1-\alpha)^2 - 2\beta}{2} \quad (33)$$

We recall that for an  $\epsilon$ -simulator,  $\epsilon > 0$  can be made polynomially small efficiently in run-time and  $n$ . Thus, as an example, we may assign the following scaling to  $\alpha, \beta, \epsilon$ :

$$\alpha = \frac{1}{n}, \quad \beta = \frac{1}{2n^2}, \quad \epsilon = \frac{\alpha \beta}{n}. \quad (34)$$

This argument shows that the existence of an  $\epsilon$ -simulator of  $\mathcal{C}_{\text{PROD}}$  implies that there exists an algorithm in  $\Delta_3^P$  that can for any fixed product states  $\rho$  and measurement outcomes  $x \in \{0, 1\}^n$ , output an  $O(1/n)$  multiplicative precision estimate of  $p_x := \text{Tr}(U \rho U^\dagger |x\rangle \langle x|)$  for almost 50% of randomly uniformly chosen Clifford unitaries  $U$  acting on  $n$  qubits. That is:

$$\Pr_U \left[ |p_x - \tilde{p}_x^\epsilon| < p_x O(1/n) \right] > \frac{1}{2} - \frac{1}{n} \quad (35)$$

By conjecture 1, this is  $\#P$ -hard. This implies that the third level of the PH contains the complexity class  $P^{\#P}$  (the set of decision problems solvable by a Turing machine with access to a  $\#P$  oracle). Since the PH is contained in  $P^{\#P}$  [44], this implies that the PH collapses to the third level.  $\square$

### Acknowledgments

The authors are grateful to Marco Tomamichel, Ryan Mann, Michael Bremner and Joel Wallman for helpful discussions. This research is supported by the ARC via the Centre of Excellence in Engineered Quantum Systems (EQuS) project number CE110001013 and by the U.S. Army Research Office through

grant W911NF-14-1-0103. HP also acknowledges support from the Australian Institute for Nanoscale Science and Technology Postgraduate Scholarship (John Makepeace Bennett Gift). The work of DG is supported by the Excellence Initiative of the German Federal and State Governments (ZUK 81), the DFG within the CRC 183 (project B01), and the DAAD.

- 
- [1] R. Feynman, “Simulating physics with computers,” *Internat. J. Theoret. Phys.*, **21**, 467 (1982).
  - [2] S. Aaronson and D. Gottesman, “Improved simulation of stabilizer circuits,” *Phys. Rev. A* **70**, 052328, (2004).
  - [3] L. G. Valiant, “Quantum Circuits That Can Be Simulated Classically in Polynomial Time,” *SIAM J. Comput.* **31**, 1229 (2002).
  - [4] B. M. Terhal and D. P. DiVincenzo, “Classical simulation of noninteracting-fermion quantum circuits,” *Phys. Rev. A* **65**, 032325 (2002).
  - [5] S. D. Bartlett, B. C. Sanders, S. L. Braunstein, and K. Nemoto, “Efficient Classical Simulation of Continuous Variable Quantum Information Processes,” *Phys. Rev. Lett.* **88**, 097904 (2002).
  - [6] L. Gurvits, “On the complexity of mixed discriminants and related problems.” In: J. Jedrzejowicz and A. Szepietowski (eds) *Mathematical Foundations of Computer Science 2005. MFCS 2005. Lecture Notes in Computer Science*, 3618. Springer, Berlin, Heidelberg.
  - [7] S. Aaronson and A. Arkhipov, “The computational complexity of linear optics,” in *Proc. ACM Symposium on Theory of Computing (STOC ’11)*, ACM, New York, NY, USA, pp. 333–342 (2011).
  - [8] M. J. Bremner, R. Jozsa, and D. J. Shepherd, “Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy,” *Proc. R. Soc. A* 2010. DOI: 10.1098/rspa.2010.0301.
  - [9] M. J. Bremner, A. Montanaro, and D. J. Shepherd, “Average-case complexity versus approximate simulation of commuting quantum computations,” *Phys. Rev. Lett.* **117**, 080501 (2016).
  - [10] X. Gao, S.-T. Wang, and L.-M. Duan, “Quantum supremacy for simulating a translation-invariant Ising spin model,” *Phys. Rev. Lett.* **118**, 040502 (2017).
  - [11] J. Bermejo-Vega, D. Hangleiter, M. Schwarz, R. Raussendorf, and J. Eisert, “Architectures for quantum simulation showing quantum supremacy,” *arXiv preprint arXiv:1703.00466* (2017).
  - [12] B. Fefferman and C. Umans, “The power of quantum fourier sampling,” *arXiv preprint arXiv:1507.05592* (2015).
  - [13] T. Morimae, K. Fujii, and J. F. Fitzsimons, “Hardness of classically simulating the one-clean-qubit model,” *Phys. Rev. Lett.* **112**, 130502 (2014).
  - [14] T. Morimae, K. Fujii, and H. Nishimura, “Power of one non-clean qubit,” *arXiv preprint arXiv:1610.07244* (2016).
  - [15] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, “Characterizing quantum supremacy in near-term devices,” *arXiv preprint arXiv:1608.00263* (2016).
  - [16] A. Bouland, J. F. Fitzsimons, and D. E. Koh, “Quantum Advantage from Conjugated Clifford Circuits,” *arXiv preprint arXiv:1709.01805* (2017).
  - [17] D. J. Brod, “Efficient classical simulation of matchgate circuits with generalized inputs and measurements,” *Phys. Rev. A* **93**, 062332 (2016).
  - [18] V. Veitch, C. Ferrie, D. Gross, and J. Emerson, “Negative quasi-probability as a resource for quantum computation,” *New J. Phys.* **14**, 113011 (2012).
  - [19] A. Mari and J. Eisert, “Positive Wigner functions render classical simulation of quantum computation efficient,” *Phys. Rev. Lett.* **109**, 230503 (2012).
  - [20] S. Bravyi, and D. Gosset, “Improved classical simulation of quantum circuits dominated by Clifford gates,” *Phys. Rev. Lett.* **116**, 250501 (2016).
  - [21] R. S. Bennink, E. M. Ferragut, T. S. Humble, J. A. Laska, J. J. Nutaro, M. G. Pleszkoch, R. C. Pooser, “Monte Carlo Simulation of Near-Clifford Quantum Circuits,” *Phys. Rev. A* **95**, 062337 (2017).
  - [22] E. T. Campbell, and M. Howard, “Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost,” *Phys. Rev. A* **95**, 022316 (2017).
  - [23] H. Pashayan, J. J. Wallman, and S. D. Bartlett, “Estimating outcome probabilities of quantum circuits using quasiprobabilities,” *Phys. Rev. Lett.* **115**, 070501 (2015).
  - [24] M. Van den Nest, “Efficient classical simulations of quantum fourier transforms and normalizer circuits over abelian groups,” *arXiv preprint arXiv:1201.4867*, (2012).
  - [25] J. Bermejo-Vega and M. Van den Nest, “Classical simulations of Abelian-group normalizer circuits with intermediate measurements,” *Quantum Inf. Comput.* **14**, 181-216 (2014).
  - [26] S. Bravyi, G. Smith, and J. A. Smolin, “Trading classical and quantum computational resources,” *Phys. Rev. X* **6**, 021043 (2016).
  - [27] K. Temme, S. Bravyi, and J. M. Gambetta, “Error Mitigation for Short-Depth Quantum Circuits,” *Phys. Rev. Lett.* **119**, 180509 (2017).
  - [28] S. Bravyi, D. Gosset, and R. Koenig, “Quantum advantage with shallow circuits,” *arXiv preprint*

- arXiv:1704.00690, (2017).
- [29] D. Shepherd, “Binary matroids and quantum probability distributions,” arXiv preprint arXiv:1005.1744 (2010).
  - [30] T. Morimae, “Hardness of classically sampling one clean qubit model with constant total variation distance error,” *Phys. Rev. A* **96**, 040302(R) (2017).
  - [31] Aaronson, Scott. ”The equivalence of sampling and searching.” *Theory Comput. Syst.* **55**, 281 (2014).
  - [32] D. Stahlke, “Quantum interference as a resource for quantum speedup,” *Phys. Rev. A* **90**, 022302 (2014).
  - [33] S. Aaronson and T. Hance, “Generalizing and derandomizing Gurvits’s approximation algorithm for the permanent,” arXiv preprint arXiv:1212.0025 (2012).
  - [34] R. Jozsa and M. Van den Nest, “Classical simulation complexity of extended Clifford circuits,” arXiv preprint arXiv:1305.6190 (2013).
  - [35] D. Shepherd and M. J. Bremner, “Temporally unstructured quantum computation,” *Proc. R. Soc. A* **465**, 1413 (2009).
  - [36] D. J. Shepherd, “Quantum complexity: restrictions on algorithms and architectures,” arXiv preprint arXiv:1005.1425 (2010).
  - [37] We thank an anonymous QIP referee for bringing this connection to our attention.
  - [38] D. Vertigan, “Bicycle dimension and special points of the Tutte polynomial,” *J. Combin. Theory Ser. B* **74**, 378-396 (1998).
  - [39] We thank an anonymous QIP referee for pointing out a simple example using encoding to smooth out Born rule probabilities and marginals.
  - [40] M. Schwarz and M. Van den Nest, “Simulating quantum circuits with sparse output distributions,” arXiv preprint arXiv:1310.6749 (2013).
  - [41] Stockmeyer, Larry, “The complexity of approximate counting,” *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, (1983).
  - [42] L. A. Goldberg and H. Guo, “The complexity of approximating complex-valued Ising and Tutte partition functions,” *comput. complex.* **26**, 765 (2017).
  - [43] G. Kuperberg, “How hard is it to approximate the Jones polynomial?,” *Theory of Computing* **11**, 183-219 (2015).
  - [44] S. Toda, “PP is as hard as the polynomial-time hierarchy.” *SIAM Journal on Computing* **20**, 865-877 (1991).

### Appendix A: Statistical indistinguishability proof

We first show a well know connection between the optimal probability of choosing the correct hypothesis in a hypothesis test and the  $L_1$  distance.

Suppose  $\mathcal{P}_1$  and  $\mathcal{P}_2$  are probability distribution over some finite set  $I$ , and suppose a sample  $X$  is observed from the distribution  $\mathcal{Q}$  where either  $\mathcal{Q} = \mathcal{P}_1$  (hypothesis  $H_1$ ) or  $\mathcal{Q} = \mathcal{P}_2$  (hypothesis  $H_2$ ). Then, any hypothesis test must have some  $H_1$  acceptance region  $A_1 \subseteq I$  and some  $H_2$  acceptance region  $A_2 := A_1^c \subseteq I$ . The probability of a type I error is  $\alpha := \Pr(X \in A_2 \mid X \sim \mathcal{P}_1)$  and the probability of a type II error is  $\beta := \Pr(X \in A_1 \mid X \sim \mathcal{P}_2)$ . The  $L_1$ -norm between  $\mathcal{P}_1$  and  $\mathcal{P}_2$  can be written as:

$$\begin{aligned} \|\mathcal{P}_1 - \mathcal{P}_2\|_1 &:= \sum_{x \in I} |\mathcal{P}_1(x) - \mathcal{P}_2(x)| \\ &= 2 \sup_{A_1 \subseteq I} [\mathcal{P}_1(A_1) - \mathcal{P}_2(A_1)] \\ &= 2 \sup_{A_1 \subseteq I} [(1 - \mathcal{P}_1(A_1^c) - \mathcal{P}_2(A_1))] \\ &= 2(1 - \alpha^* - \beta^*) \end{aligned}$$

where, the second equality can be verified by noting that the supremum is achieved when  $A_1 = \{x \in I \mid \mathcal{P}_1(x) \geq \mathcal{P}_2(x)\}$ . Here,  $\alpha^*$  and  $\beta^*$  are the type I and type II errors for the optimal choice of acceptance region / hypothesis test. We note that if a priori,  $H_1$  and  $H_2$  are equally likely, then the probability of choosing the correct hypothesis, based on a single sample, using the optimal test is thus given by:

$$\begin{aligned} P_{\text{correct}} &= 1 - \Pr(X \in A_2 \mid X \sim \mathcal{P}_1)\Pr(X \sim \mathcal{P}_1) - \Pr(X \in A_1 \mid X \sim \mathcal{P}_2)\Pr(X \sim \mathcal{P}_2) \\ &= 1 - \alpha^* \Pr(H_1) - \beta^* \Pr(H_2) \\ &= \frac{1}{2} + \frac{\|\mathcal{P}_1 - \mathcal{P}_2\|_1}{4}. \end{aligned}$$

The interactive protocol between the referee and the candidate will proceed as follows (see Figure 2):

1. Initially, the referee will fix a test by choosing a function  $a$  that dictates how all gathered data in prior rounds determines the next circuit request. We note that while this can be further generalized by allowing stochastic maps (rather than functions), this has no bearing on our results and our proof can fairly easily be extended if required.
2. Initially the referee will make the circuit request  $a_\emptyset \in \mathcal{A}^*$
3. The response from the candidate is denoted by the random variable  $\tilde{Y}_{a_\emptyset}$  and the string of random variables  $a_\emptyset, \tilde{Y}_{a_\emptyset}$  will be represented by  $\tilde{X}_1$
4. The referee may make another circuit request by applying the map  $a$  to  $\tilde{X}_1$  thus defining the next circuit request  $a(\tilde{X}_1)$ .
5. On the  $(j+1)^{th}$  round, the referee's circuit request will be represented by  $a(\tilde{X}_j)$  and the response will be represented by  $\tilde{Y}_{a(\tilde{X}_j)}$  where,  $\tilde{X}_{j+1}$  represents the string of random variables  $\tilde{X}_j, a(\tilde{X}_j), \tilde{Y}_{a(\tilde{X}_j)}$ .
6. In addition, at the end of the  $j^{th}$  round for  $j = 1, 2, \dots$ , a fixed stochastic binary map  $h$  will be applied to  $\tilde{X}_j$  with the outcome determining whether or non to halt the interactive procedure. We will assume that the test will eventually halt and represent the final round of any given test by  $m \in \mathbb{N}$ .
7. Finally, the referee will decide  $H_a$  vs  $H_b$  by applying a fixed binary map  $d$  to the full collected data set  $\tilde{X}_m$ .

We will use the notation convention above but in the case when the candidate is fixed to be Alice, we will remove the tilde (i.e.  $\tilde{X}, \tilde{Y} \rightarrow X, Y$ ) and alternatively when the candidate is fixed to be Bob, we will replace the tilde with a prime (i.e.  $\tilde{X}, \tilde{Y} \rightarrow X', Y'$ ).

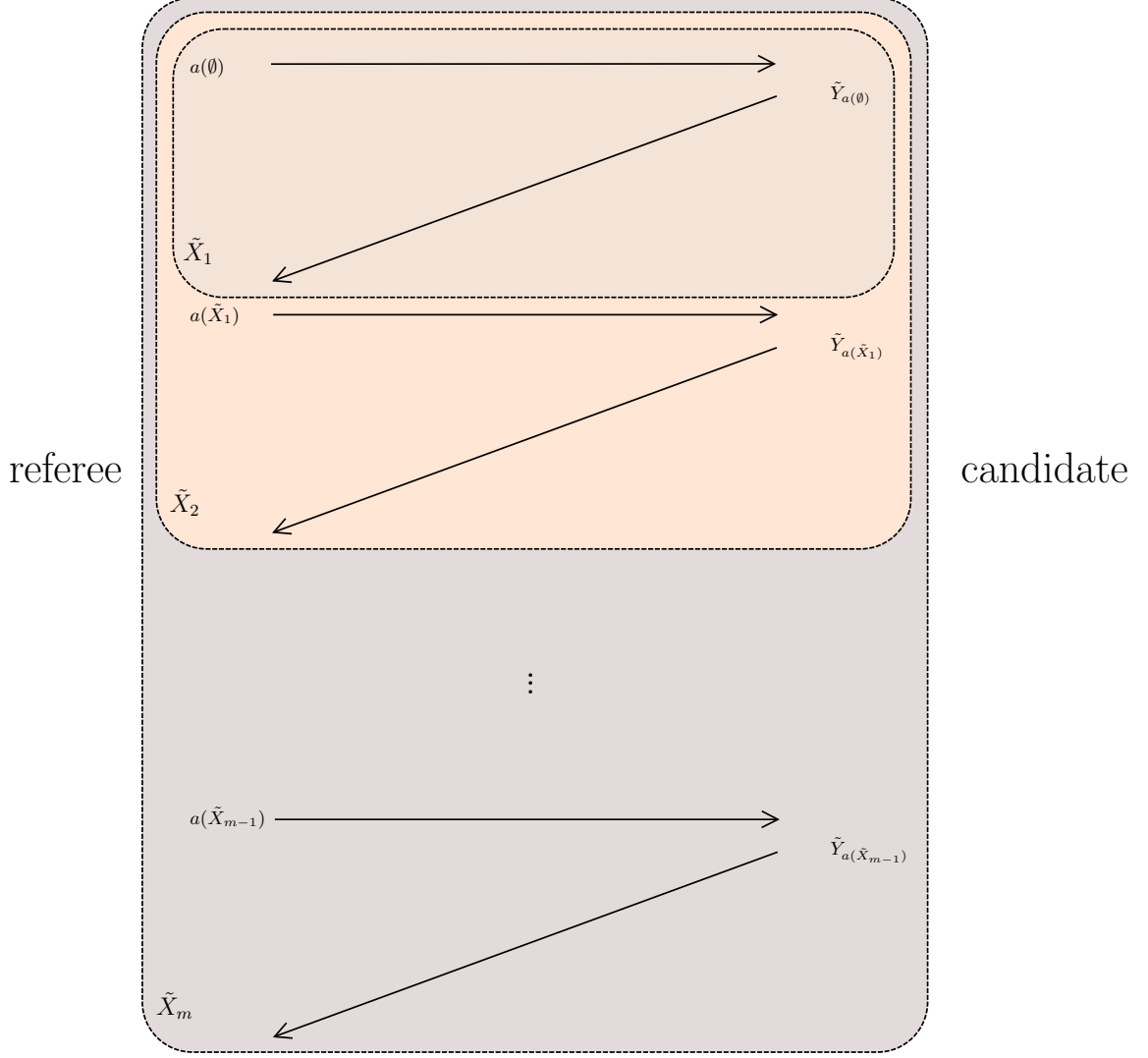


FIG. 2: The figure above shows the interactive protocol between the referee and the candidate. In each round of the protocol, the referee send a circuit description to the candidate. This circuit description is in general given by applying any fixed (possibly stochastic) map to all of the prior data collected by the referee. The candidates responses ( $\tilde{Y}$ ) may depend only on the circuit request from the current round and the round number. When the candidate is known to be Alice or Bob, we will represent the variables corresponding to  $\tilde{Y}$  and  $\tilde{X}$  by  $Y$  and  $X$  or  $Y'$  and  $X'$  respectively.

*Proof.* We now prove each direction of the “if and only if” statement of Theorem 1.

“ $\Rightarrow$ ” Here, we assume that Bob’s simulation scheme is an  $\epsilon$ -simulator over  $\mathcal{C}$  and explicitly specify a strategy for Bob which simultaneously achieves indistinguishability and efficiency.

Bob’s strategy will be as follows; if he becomes the candidate, then in the  $j^{th}$  round of the protocol, he will be asked to report the outcome of running some circuit indexed by  $a_j \in \mathcal{A}^*$ . In this case, Bob will  $\epsilon$ -simulate the circuit  $c_{a_j}$  with the precision setting given by:

$$\epsilon_j = \frac{24\delta}{\pi^2 j^2}$$

We note that Bob's strategy is independent of the referee's test. Further, we note that for all  $m \in \mathbb{N}$ :

$$\begin{aligned} \sum_{j=1}^m \epsilon_j &\leq \sum_{j=1}^{\infty} \epsilon_j \\ &= 4\delta \end{aligned} \quad (\text{A1})$$

We define the map  $\mathcal{E}[X, X']$  from any pair of random variables  $X$  with probability distribution  $\mathcal{P}$  and  $X'$  with probability distribution  $\mathcal{P}'$  to  $\mathbb{R}$  as the  $L_1$  distance between  $\mathcal{P}$  and  $\mathcal{P}'$ .

We will show that for every test, the quantity on the LHS of eq. A1 upper bounds  $\mathcal{E}[X_m, X'_m]$ . Hence:

$$\mathcal{E}[X_m, X'_m] \leq 4\delta \quad (\text{A2})$$

$$\mathcal{E}[X_{j+1}, X'_{j+1}] = \sum_{\alpha, \beta} \left| Pr(Y_{a(X_j)} = \beta | X_j = \alpha) Pr(X_j = \alpha) - Pr(Y'_{a(X'_j)} = \beta | X'_j = \alpha) Pr(X'_j = \alpha) \right| \quad (\text{A3})$$

$$= \sum_{\alpha, \beta} |Pr(Y_{a(X_j)} = \beta | X_j = \alpha) Pr(X_j = \alpha) - Pr(Y_{a(X_j)} = \beta | X_j = \alpha) Pr(X'_j = \alpha)| \quad (\text{A4})$$

$$+ Pr(Y_{a(X_j)} = \beta | X_j = \alpha) Pr(X'_j = \alpha) - Pr(Y'_{a(X'_j)} = \beta | X'_j = \alpha) Pr(X'_j = \alpha)| \quad (\text{A5})$$

$$\leq \sum_{\alpha, \beta} Pr(Y_{a(X_j)} = \beta | X_j = \alpha) |Pr(X = \alpha) - Pr(X' = \alpha)| \quad (\text{A6})$$

$$+ \sum_{\alpha, \beta} Pr(X'_j = \alpha) |Pr(Y_{a(X_j)} = \beta | X_j = \alpha) - Pr(Y'_{a(X'_j)} = \beta | X'_j = \alpha)| \quad (\text{A7})$$

$$\leq \mathcal{E}[X_j, X'_j] + \sum_{\alpha} \mathcal{E}[Y_{a(\alpha)}, Y'_{a(\alpha)}] Pr(X'_j = \alpha) \quad (\text{A8})$$

where the sums are taken over  $\alpha$  in the support of  $\tilde{X}_j$  and  $\beta$  in  $\bigcup_{a \in \mathcal{A}^*} \text{supp}(\tilde{Y}_a)$ .

We note that the precision of Bob's response in any round only depends on the round number. Thus, Eq. (A8) can be simplified by making the replacement  $\mathcal{E}[Y_{a(\alpha)}, Y'_{a(\alpha)}] \rightarrow \epsilon_{j+1}$ . Combined with the observation that  $\mathcal{E}[X_1, X'_1] = \epsilon_1$ , we have shown that:

$$\begin{aligned} \mathcal{E}[X_m, X'_m] &\leq \sum_{j=1}^m \epsilon_j \\ &\leq 4\delta \end{aligned}$$

This proves that Bob's strategy meets the indistinguishability property. We now consider the efficiency of the strategy. We recall that given a circuit request sequence  $\alpha$ , Alice's and Bob's resource costs are represented by  $N(\alpha)$  and  $T(\alpha)$  respectively.

By definition of  $\epsilon$ -simulation, there exists  $\kappa, c_1, c_2 \in \mathbb{N}$  such that for a given circuit index  $a$ , and precision  $\epsilon$ ,  $T(a) \leq c_1 \left( \frac{N(a)}{\epsilon} \right)^\kappa + c_2$ . For simplicity, we will set  $c_1 = 1$  and  $c_2 = 0$  as this is immaterial given sufficiently large  $N(\alpha)$  and  $\frac{1}{\epsilon}$ . For  $m = 1$ , clearly the strategy is efficient. Hence, given a string of inputs

$\alpha = (a_1, \dots, a_m)$ , with  $m \geq 2$  we have:

$$T(\alpha) = \sum_{j=1}^m T(a_j) \quad (\text{A9})$$

$$\leq \sum_{j=1}^m \left( \frac{N(a_j)}{\epsilon_j} \right)^\kappa \quad (\text{A10})$$

$$= \sum_{j=1}^m \left( \frac{\pi^2 j^2 N(a_j)}{24\delta} \right)^\kappa \quad (\text{A11})$$

$$\leq \left( \frac{\pi^2}{24\delta} \right)^\kappa \left[ \sum_{j=1}^{m-1} j^{2\kappa} + m^{2\kappa} [N(\alpha) - (m-1)]^\kappa \right] \quad (\text{A12})$$

$$\leq \left( \frac{\pi^2}{24\delta} \right)^\kappa \left[ \left( \frac{m-0.5}{2\kappa+1} \right)^{2\kappa+1} + m^{2\kappa} N(\alpha)^\kappa - m^{2\kappa} (m-1)^\kappa \right] \quad (\text{A13})$$

$$\leq \left( \frac{\pi^2 m^2 N(\alpha)}{24\delta} \right)^\kappa \quad (\text{A14})$$

$$\leq \left( \frac{\pi^2 N(\alpha)^3}{24\delta} \right)^\kappa \quad (\text{A15})$$

$$\in O \left( \text{poly}(N(\alpha), \frac{1}{\delta}) \right) \quad (\text{A16})$$

where:

- in eq. A12 we have used the fact that  $N(a_j) \geq 1$  for all  $j$  hence the expression is maximized when  $\alpha$  is chosen such that  $N(a_j) = 1$  for  $j = 1, \dots, m-1$  and  $N(a_m) = N(a_1) + \dots + N(a_m) - (m-1)$
- in eq. A13 we have used integration to show the inequality for any  $k \in \mathbb{N}$ ;  $\sum_{j=1}^m j^k < (\frac{m+0.5}{k+1})^{k+1}$  and
- in eq. A13 we have also used the fact that for  $\kappa > 1$  and  $N \geq m > 0$ , one can show that  $(N-m)^\kappa \leq N^\kappa - m^\kappa$
- in eq. A14 we have used the inequality  $\left( \frac{m-0.5}{2\kappa+1} \right)^{2\kappa+1} - m^{2\kappa} (m-1)^\kappa \leq 0$  for  $m \geq 2$  and  $\kappa \geq 1$

hence, there exists a polynomial  $f(x, y)$  such that for all request strings  $\alpha$  and  $\delta > 0$ ,  $T(\alpha) \leq f(N(\alpha), \frac{1}{\delta})$ .

“ $\Leftarrow$ ”: We restrict ourselves to interactive protocols consisting of only one round. For each fixed circuit request, under the optimal choice of the decision map  $d$ ,  $\delta \propto \epsilon$  hence for all  $c \in \mathcal{C}$  and for all  $\epsilon > 0$ , Bob must be able to sample from some distribution  $\mathcal{P}^\epsilon \in B(\mathcal{P}, \epsilon)$ . Further, since Bob’s strategy meets the efficiency condition, for every  $a \in \mathcal{A}^*$ , Bob must be able to output the sample using resources  $\in O(\text{poly}(N(a), \frac{1}{\delta})) \subseteq O(\text{poly}(n, \frac{1}{\epsilon}))$ .

□