The University of Sydney

On the classical simulability of quantum circuits

Author:
Hakop Pashayan

Supervisor:
Prof. Stephen D. Bartlett

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

School of Physics
Faculty of Science

December 6, 2019

Contents

Abstract

Research into the classical simulation of quantum circuits has produced influential conceptual leaps and, practical solutions to important problems. Literature in this area has generally focused on *what* circuits can be, and *how* these can be simulated. This thesis also aims to answer these questions but shifts its focus toward understanding how and why the answers to these questions change when we change the definition of what it means to classically simulate. Whether a class of quantum circuits can be efficiently simulated with a probabilistic classical computer, or is provably hard to simulate, depends quite critically on the precise notion of "classical simulation" and in particular on the required accuracy. We focus on two important notions of simulator, that we refer to as *poly-boxes* and EPSILON-*simulators* and, discuss how other notions of simulation relate to these. A poly-box is a *classical* algorithm that outputs additive $1/poly$ precision estimates of Born probabilities and marginals. Poly-boxes offer useful practical solutions for important problems that have eluded pragmatic classical solutions using stronger notions of simulation. We present a general mathematical framework that can be used to construct poly-boxes for certain quantum circuit families. This framework provides a flexible mathematical structure that can be combined with a number of free parameter choices (we call the *model*) to produce a poly-box. This framework (sometimes partially) generalizes a number of recent works on simulation. By reformulating simulation techniques used in these works into the general framework, we show how the poly-box's performance is influenced by the choice of model. As an application, we use the general framework to construct a classical additive $1/poly$ precision Born rule probability estimation algorithm for Clifford plus T circuits. Our algorithm scales exponentially in the number of T gates but polynomially in all other parameters and is intended to be state of the art for this estimation task. We expect this result to be particularly useful in the characterization and verification of near term quantum devices.

Investigating the classical simulability of quantum circuits also provides a promising avenue towards understanding the computational power of quantum systems. We argue that the notion of classical simulation we call EPSILON-simulation, captures the essence of possessing "equivalent computational power" to the quantum system it simulates: It is statistically impossible to distinguish an agent with access to an EPSILON-simulator from one possessing the simulated quantum system. We relate EPSILON-simulation to various alternative notions of simulation predominantly focusing on its relation to poly-boxes. Accepting some plausible computational theoretic assumptions, we show that EPSILON-simulation is strictly stronger than a poly-box by showing that IQP circuits and unconditioned magic-state injected Clifford circuits are both hard to EPSILON-simulate and yet admit a poly-box. In contrast, we also show that these two notions are equivalent under an additional assumption on the sparsity of the output distribution (*poly-sparsity*).

Statement of Student Contribution

The research presented in this thesis was conducted under the supervision of Professor Stephen Bartlett. The research presented can be categorized into 3 main components.

The first of these is contained in Ch. 4 and presents a general framework for probability estimation. The early ideas in this work relating to algorithm construction were inspired by a paper [1] I wrote with my co-authors, Joel Wallman and Stephen Bartlett. In the late stages of this research, some ideas have arisen out of discussions with Kamil Korzekwa and Earl Campbell. I derived all of the mathematical results in Ch. 4 with input from Stephen Bartlett. This chapter was written by me with input, feedback and corrections from Stephen Bartlett and Kamil Korzekwa.

The second research component is contained in Ch. 5 and presents an algorithm for estimating Born rule probabilities for Clifford plus T circuits. This research project emerged from my attempts to apply the simulation techniques presented in the general framework to the work of Bravyi and Gosset [2]. This is a joint research project with Kamil Korzekwa and Stephen Bartlett. Kamil Korzekwa has been involved in this project from its commencement and has made important technical and conceptual contributions. Ch. 5 was written by me, with the exception of Secs. 5.3 to 5.6, which were written jointly with Kamil Korzekwa and will ultimately form the basis of a paper with myself as first author. All of the content of this chapter benefited from input, feedback and corrections from Stephen Bartlett and Kamil Korzekwa.

The third research component is the content of Part II establishing the relation between EPSILON-simulation, poly-boxes and computational power. This research is based on a paper [3] I wrote with my co-authors, Stephen Bartlett and David Gross. The main ideas were developed by myself in ongoing discussions with Stephen Bartlett. At the later stages, ideas relating to a new research direction (to show the hardness of EPSILON-simulation) were developed in discussions with David Gross. All technical results were derived by me with the exception of the proof of Lem. 11. Here, I produced an early version of the anti-concentration proof that only held for pure states. David Gross realised that my proof did not hold for general mixed states and replaced it with a more concise and elegant proof. Ref. [3] (and consequently the content of Part II) was predominantly written by me but includes substantial contributions from Stephen Bartlett and David Gross.

All other parts of this thesis were written by me with input, feedback and corrections from Stephen Bartlett and Kamil Korzekwa.

Statement of Originality

I, Hakop Pashayan, hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. I certify that the intellectual content of this thesis is the product of my own work and that any assistance received in preparing this thesis and sources have been acknowledged, either in the text or as outlined in the Statement of Student Contribution. This dissertation contains fewer than 80,000 words, excluding appendices and bibliography.

Hakop Pashayan

December 6, 2019

Acknowledgements

I am extremely grateful for the support, wisdom and mentorship provided to me by my PhD supervisor, Professor Stephen Bartlett.

I am fortunate to be surrounded by such knowledgeable and generous collegues. I would like to thank Adrian Chapman, Alan Robertson, Andrew Doherty, Arkin Inas Tikku, Arne Grimsmo, Ben Brown, Kamil Korzekwa, Marcus Appleby, Nicolas Menicucci, Parth Girdhar, Paul Webster, Rafael Alexander, Robin Harper, Steven Flammia, Wei-Wei Zhang and Zachary Cristina. Each of you has helped me grow and made my time in the office more enjoyable. I am particularly thankful to Christopher Chubb, Angela Karanjai and Sam Roberts who have been extremely helpful and generous with their time.

I would like the thank the following people each of whom has imparted knowledge or an enlightening perspective: Abhinav Deshpande, Alex Moylett, Alexander Kubica, Álvaro M. Alhambra, Andrew Darmawan, Andrew Guo, Aniruddha Bapat, Anurag Anshu, Ashley Montanaro, Beni Yoshida, Brian Swingle, Carlo Sparaciari, Carlton Caves, Chris Jackson, Dan Browne, Daniel Brod, Daniel Gottesman, David Gosset, David Gross, David Poulin, David Schmid, Dax Koh, Debbie Leung, Denis Rosset, Dominik Hangleiter, Earl Campbell, Eddie Schoute, Elie Wolfe, Fabio Costa, Felipe Montealegre-Mora, Hammam Elyas, Huangjun Zhu, Ingo Roth, Ivan Deutsch, Jamie Sikora, Jens Eisert, Joel Wallman, Johan Åberg, Johannes Bausch, Jon Yard, Joseph Emerson, Juani Bermejo-Vega, Lorenzo Catani, Lucas Kocia, Magdalena Zych, Marco Tomamichel, Marek Gluza, Mark Howard, Markus Frembs, Markus Heinrich, Mateus Araújo, Michael Bremner, Michael Vasmer, Mithuna Yoganathan, Nadish de Silva, Ninnat Dangniam, Peter Turner, Ravi Kunjwal, Richard Cleve, Richard Jozsa, Richard Kueng, Robert Spekkens, Roberto Di Candia, Ryan Mann, Sathyawageeswar Subramanian, Sergii Strelchuk, Shane Mansfield, Simone Severini, Stephen Jordan, Stephen Piddock, Terry Rudolph, Thomas Galley, Tobias Fritz, Troy Sewell and Zachary Fisher. Thank you and apologies to all of those who I have forgotten.

I owe an immense debt of gratitude to my lovely wife Alexandra Goymour, my inspirational mother and my family for their unshakeable support and continuous encouragement.

Finally, to my dearest daughter Amalia Myee Pashayan. Thank you for being so delightfully cute and adorably clever.

Chapter 1

Introduction

A computation is a physical process. Here, the initial and final configuration of a dynamic physical system are used to map an encoding of a problem to an encoding of a solution. The time, energy or other resources required in order to perform a particular computation using a particular computational device are inevitably governed by the laws of physics. Specifically, the computational device is limited in the physical processes it can exploit, and the laws of physics govern the resource cost of executing these processes.

There has been almost a century of research exploring many facets of computation in the context of physical systems that "solve" computational tasks by relying on *classical* dynamical processes. Early work mathematically formalized the theory of computation. The computational device was formalized to the Turing machine [4] and the dynamical mapping between its configurations was abstracted to a computation of a function on the natural numbers.

The weak form of the Church-Turing Thesis (CTT) states that a function on the natural numbers is physically computable, using finite resources, by any mechanical means if and only if it is computable by a Turing machine. This thesis was proposed as a result of the equivalence of a number of different computational models (Turing's machines [4], Gödel's *general recursive functions* [5] and Church's λ *calculus* [6]) and is still unchallenged and widely accepted.

From the perspective of computability using finite resources, the CTT asserts the equivalence of many models of computation including classical and quantum computation. A more resource sensitive formalism was developed by shifting from the concept of computability to *efficiency*. Given an infinite family of computational tasks, in this formalism, the resource requirements of a computational device are considered as a function of the size of the input. The computational device is said to *efficiently* solve the family of tasks (problem) if its resource requirements scale at most polynomially in the size of the input. Many classes of problems known as complexity classes have been defined in terms of sets of problems which can be solved efficiently by a particular computational device. For example, the complexity classes BPP and BQP define the set of binary outcome problems – *decision problems* – that can be efficiently solved by a (probabilistic, universal) classical computer and by a universal quantum computer respectively.

The strong form of the CTT states that any function on the natural numbers that can be efficiently computed physically can be efficiently computed by a Turing machine. This strong form is now seriously challenged by the emergence of quantum computation.

The brilliant insight of Feynman [7] was that for certain tasks, quantum computers appear to exploit the quantum behavior of certain physical systems in a way that perform computations using fewer resources. This and other pioneering works leveraged insights about important differences between classical and quantum mechanical laws of nature to construct the foundations of the field of quantum computation.

The early insights into quantum phenomena such as exponentially growing Hilbert spaces, superposition and interference and how to use these in computation led to the first quantum algorithms including the Deutsch-Jozsa algorithm [8], Shor's factoring algorithm [9] and Grover's search algorithm [10]. In a similar vein, an appreciation of the believed classical hardness of distinguishing probability distributions [11] motivates important quantum tools such as the Hadamard test and the SWAP test [12]. More recently, the classical hardness of computing the permanent of random binary matrices motivated Aaronson and Arkhipov's seminal work on boson sampling [13] and the sub-field focused on demonstrating quantum advantage.

Classical simulation of quantum systems is the primary formal tool for the theoretical exploration of the computational advantages of quantum mechanics. The inability to efficiently classically simulate quantum interactions sparked Feynman's idea of quantum computation. And the appearance of matrix permanents in attempts to classically simulate bosonic quantum systems combined with our modern understanding of classical complexity theory (in particular the average case hardness of classically estimating matrix permanents [14]), gave rise to the active study of intermediate models of quantum computation such as boson sampling [13] and instantaneous quantum polynomial time circuits [15, 16]. In this historical context, the role of classical simulation of quantum systems is to deliver deeper theoretical insights about the fundamental differences between classical and quantum systems. We refer to this domain of application as "classical-hardness inspired quantum insights".

The classical simulation of quantum systems is much broader than the few significant examples that spring to mind [17, 18, 19, 20, 21]. What it means to classically "simulate" a quantum system is a matter of definition. Such a definition picks out a particular qualitative or quantitative feature the classical system must replicate to successfully achieve "simulation" of the quantum system. One may define the classical simulation of a family of quantum circuits as the ability to solve decision problems with a bounded error probability. Alternatively, one may define it in terms of the ability to estimate, within a specified accuracy, the Born rule probability of certain events or the expectation value of certain observables. As yet another alternative, simulation can be defined in terms of the ability to sample from the output distribution of a quantum circuit. A crucial observation is that subtle changes to the notion of simulation will exhibit different manifestations of the fundamental differences between classical and quantum systems. Under one particular notion of simulation, a qualitative property of the quantum system may indicate a clear separation between classically simulable and classically hard to simulate. Under another notion of simulation this indicator of "quantumness" may be obscured or perhaps appear as a quantitive property that measures "quantumness".

As an example, Refs. [22, 23] showed that the discrete Wigner function can be used to give an efficient simulation of Clifford circuits acting on odd dimensional quantum systems provided that the states and measurements have non-negative discrete Wigner functions (see also Refs. [24, 23] for the continuous variable analog). The notion of simulation used here, required that the classical algorithm be able to sample from the quantum output distribution. This has been referred to as *weak simulation* in the literature [25, 26, 16, 27]. Within this model it is believed that weak simulation is not possible for circuits where the input state or measurement has a discrete Wigner function that takes a negative value at some point in phase space. This is in part supported by the fact that many negative states can be distilled to magic states allowing universal quantum computation [28, 29], the connections between negativity and quantum phenomenon of contextuality [30] and the folklore from the quantum optics community that negativity of the Wigner function is a uniquely quantum phenomenon [31]. Despite the fact that from the perspective of efficient weak simulation the presence of any negativity appears to be an insurmountable obstacle, our work [1] showed that under an alternate notion of efficient classical simulation, the presence of negativity merely results in a computational runtime overhead that increases incrementally with increases in negativity. This gave us an operationally meaningful way to quantify negativity as a resource for quantum computation. This insight arose from the contrast between what is classically simulable under two different notions.

In light of the above, the development of classical-hardness inspired quantum insights will be greatly aided by a broader exploration of notions of simulation, their advantages and limitations as well as work focused on tailoring notions of simulation that best highlight specific distinctions between the classical and quantum paradigms. The second half of this thesis is dedicated towards this endeavor. In this thesis (see also Ref. [3]), we develop a notion of simulation (EPSILON-simulation) that faithfully captures the ability to solve computational problems in a very general sense. By leveraging other results on classical "simulation" of quantum system and relating EPSILON-simulation to other notions of simulation, we are able to develop a more refined picture of how various properties of (non-universal) quantum computers come together to result in either a computer that can efficiently solve a problem that is not solvable by efficient classical means, or otherwise.

The study of classical simulation of quantum systems also produces direct practical benefits and will become increasingly important moving forward. The vastly different structure of quantum mechanics means that it more naturally lends itself to solving certain problems. However, often there is also a classical solution to the problem which may become easier or more important to identify given the quantum solution. In this context, the development of classical algorithms for quantum tasks offers cheap, fast and presently accessible solutions to computational problems that are of practical significance. A prominent recent example is the work of Ewin Tang [32] on a classical algorithm for recommendation systems. We borrow terminology from Tang in calling this application of classical simulation "quantum inspired classical solutions" but note that it has a rich history with prominent examples including the classical simulation of Clifford circuits (Gottesman-Knill theorem [17, 18]) and classical simulation of non-interacting fermionic quantum systems [19, 20, 21]. These classical simulation algorithms have been invaluable tools with practical applications in the sub-fields of quantum error correction; quantum characterization, validation and verification; measurement based quantum computation and condensed matter physics.

We use our work on classical simulation [1] to give examples of the practical benefits of studying classical simulability. In Ref. [1], we presented a classical simulation algorithm for estimating the probabilities associated with certain quantum events. This result has found a number of important applications. For example, in Ref. [33], our algorithm was used to numerically assess the performance of the Steane code in the more realistic regime where all Clifford operations are subject to small coherent noise. Howard and Campbell [34] built on our work to show lower bounds on magic state distillation protocols and Temme, Bravyi and Gambetta applied our techniques to the task of using noisy quantum circuits to simulate ideal quantum circuits [35]. In addition, this work has inspired a number of important developments in the classical simulation literature [36, 2, 34, 3, 37, 38, 39]. In this thesis, we will present a general framework for the estimation of Born rule probabilities. This framework substantially generalizes our earlier work [1] as well as a number of additional results. This work provides a general framework that affords us a common lens through which we can study and compare many of the significant developments in the field including Refs. [40, 41, 2, 34, 3, 37, 38].

This thesis is structured into two parts. Part I is focused on the development of a general technique for the classical simulation of quantum systems. In particular, this technique focuses on classically estimating Born rule probabilities associated with certain events. In Ch. 2 we will give a more focused introduction to the topic of classical estimation of Born rule probabilities followed by some background material in Ch. 3. We present the general framework for Born probability estimation in Ch. 4. Here, we will also give a detailed overview of a number of classical simulation techniques from recent literature and discussion how they relate to the general framework. Ch. 5 is an application of the general framework and presents a classical simulation algorithm for the estimation of Born rule probabilities associated with the universal family of circuits generated by Clifford and T gates. This algorithm is a work in progress but is intended to be the state of the art method for the task of computing additive polynomial precision estimates of Born rule probabilities. In Ch. 6, we conclude with a discussion of Born rule probability estimation and provide an outlook.

In Part II of this thesis, the theme will shift from the pragmatic focus of "quantum inspired classical solutions" to the more concept focused "classical-hardness inspired quantum insights". This work identifies a notion of simulation, EPSILON-simulation, that in a certain precise sense minimally captures the notion of computational power. In Ch. 7 we will give a more detailed introduction to the context and central ideas motivating this work. Ch. 8 contains the mathematical results that underpin the conceptual significance of EPSILON-simulation. In Ch. 9 we will briefly return to discussing Born rule probability estimation but this time from a complexity theoretic perspective. The connection between various notions of simulation and EPSILON-simulation is presented in Ch 10. We devote particular attention to the more subtle relationship between poly-boxes (additive polynomial precision Born rule probability estimation) and EPSILON-simulators where we show that the two simulators are equivalent under certain conditions. Ch. 11 focuses on showing that the two simulators are also distinct. We conclude with a discussion in Ch. 12.

Part I

Estimation of Born probabilities

Chapter 2

Introduction: Estimation of Born probabilities

The Born rule is the essential ingredient that translates the abstract world of complex Hilbert spaces to observable predictions about the physical world. It provides a formula for computing the probability associated with observing any particular outcome in a given quantum experiment. In any given quantum process, the set of Born rule probabilities associated with every possible outcome fully characterize all physically observable predictions of quantum theory. For this reason, it is very natural to define notions of classical simulation of quantum systems in relation to Born rule probabilities.

As we have already briefly discussed, there is a great diversity of candidates when choosing a notion of simulation. With respect to notions that relate to Born rule probabilities, two important categories are those based on *estimation* and those based on *sampling*. Definitions of classical simulations falling into the sampling category require the classical simulator to approximately or exactly *sample* for the quantum outcome distribution (specified by the Born rule probabilities). Part II of this thesis will focus on this category of notions.

Definitions of classical simulations falling into the estimation category require the classical simulator to *calculate* an approximation of the Born rule probability associated with a specified events. The variation in definitions within this category arises from differences in the demands placed on a simulator in relation to the precision of estimates, the run-time and the range of inputs for which the simulator must satisfactorily perform. Within the estimation category, the literature has almost entirely focused on *strong simulation* and *multiplicative error* or *relative error* estimation [17, 19, 20, 26, 42, 2]. These notions of simulation impose very strong requirements on the precision to run-time tradeoff. As an example, under strong simulation, the classical device is required to efficiently estimate Born rule probabilities to within an exponentially small (in the number of qubits) additive error.

The adaptability of these precise estimation algorithms has rightfully motivated research focus in this direction. As we will discuss in Ch. 10, the precise estimation algorithms of this type can be employed as sub-procedures in efficient classical protocols to construct accurate simulators for sampling tasks amongst other applications. Thus, in a sense, these precise estimation algorithms can be seen as bona fide simulators because they offer substantial freedom to manipulate the algorithm's output (e.g. from estimates to samples) to suit a range of applications.

In Part I of this thesis, we will instead focus on a weaker notion of Born rule probability estimation. Under this notion of simulation the classical device is required to efficiently estimate Born rule probabilities (and all marginal probabilities) to within an inverse polynomially small (in the number of qubits) additive error. We call a computational device that satisfies this notion of simulation (to be precisely defined later) a *poly box*. Following our work [1] and the related work of Ref. [43], estimation algorithms targeting this weaker additive polynomial level of precision have started to gain prominence [34, 44, 37, 3]. This notion of simulation is in general limited in its range of application compared to strong and multiplicative precision estimation as well as weak simulation. However, the study of classical simulation under this notion also offers a number of significant advantages. Before discussing these, let us put into context the computational hardness of Born rule probability estimation to various levels of precision.

First let us note that the exact computation of Born rule probabilities associated with universal quantum circuits is as hard as the hardest problems in the complexity class #P (also known as #P-hard). This is an extremely powerful complexity class containing counting problems believed to be well outside the reach of classical and even universal quantum computers. Estimating these probabilities to additive exponential precision or even to within a multiplicative factor of $\sqrt{2}$ remains #P-hard [45, 46]. Thus it is believed that the estimation of general Born rule probabilities to these high levels of accuracy is not possible even with an ideal universal quantum computer. We will see that reducing the accuracy requirement to the level of polynomial additive errors, makes Born rule probability estimation realizable on a universal quantum computer. However, this is believed to still be well outside of classical computational power. In fact, even the computational ability to estimate general Born rule probabilities to within an additive error of $1/6$ is sufficient to solve the set of all problems in BQP. Thus, the efficient classical estimation of general Born rule probabilities to within an inverse polynomial additive error is not possible unless BQP = BPP.

It is strongly believed that counting problems in the complexity class #P-hard cannot be solved by a universal *quantum* computer. Accepting this, it is evident that undertaking to compute Born rule probabilities to the aforementioned high levels of precision is superfluous and prohibitively hard. Additionally, in some cases, estimating the corresponding quantity to additive polynomial precision is easy. As a classical example, given an arbitrary n variable Boolean formula in Conjunctive Normal Form (CNF), the task of computing the proportion of bit-strings that satisfy the Boolean formula is #P-hard (see #SAT or #3SAT) but an additive polynomial precision approximation can be computed simply by sampling random bit-strings and approximating the proportion using the observed frequency with which the sampled bit-strings satisfy the Boolean formula. Looking beyond notions based on high precision estimation, our work in Ref. [3] shows that even when we consider the easier task of sampling from the output distribution of general quantum circuits, there exist a number of known examples of intermediate models of quantum computing (IQP circuits-$\mathcal{C}_{\mathrm{IQP}}$, unconditioned magic-state injected qubit Clifford circuits-$\mathcal{C}_{\mathrm{PROD}}$, and a significantly more general version of the corresponding qudit class-$\mathcal{C}_{poly\mathcal{N}}$) that do not plausibly admit a classical (sampling based) simulator but allow efficient classical estimation to inverse polynomial additive precision. These results and the above named families of quantum circuits, $\mathcal{C}_{\mathrm{PROD}}$, $\mathcal{C}_{\mathrm{IQP}}$, $\mathcal{C}_{poly\mathcal{N}}$ will be discussed in detail in Part II.

In light of this discussion, the study of classical simulation in terms of additive polynomial precision estimation of Born rule probabilities offers greater scope for simulation but at a reduced level of precision. This non-trivial extension in scope to include interesting computational models including a number of known intermediate models of quantum computing offers and has already delivered new conceptual insights and practical benefits.

Part I of this thesis significantly generalizes and refines our earlier work [1]. In Ref. [1], we presented a method for (possibly inefficiently) classically estimating (to additive polynomial precision) the probabilities and marginals associated with any given quantum circuit. This technique works by transforming the mathematical objects that we conventionally use to describe quantum processes into new mathematical objects that can more naturally be interpreted as describing a generalized stochastic process. In particular this procedure maps density states, unitary transformations and positive operator valued measurement (POVM) elements to their corresponding objects in a quasi-probability representation. This mapping is not uniquely specified but rather depends on a choice of dual frame [47, 48]. We will define dual frames later but for now point out that a well known example is the set of phase point operators [49, 50, 51] that define a self-dual frame and for odd dimensional quantum systems produce the quasi-probability representation known as the discrete Wigner function. For certain quantum processes (and choice of dual frame), the objects in the quasi-probabilistic representation corresponding to density states, unitary transformations and POVM elements are probability distributions, stochastic maps and conditional probability distributions respectively. However, in general the state and POVM element representations may take on negative values and the representation of a unitary transformation may map some probability distributions to a quasi-probability distribution (that has a negative value somewhere). Prior to our work, it was known [24, 23] that if the Wigner function was non-negative, then, the outcome distribution of the quantum circuit could be sampled via a Markov chain Monte-Carlo procedure. In the case of odd dimensional quantum systems, these results generalized the Gottesman-Knill theorem. Importantly, this Monte-Carlo sampling procedure is undefined when any of the quasi-probabilities are negative. We were able to show that for any choice of quasi-probabilistic representation, there is a quantity (negativity) associated with every state, unitary transformation and POVM element. For example, the negativity of a quantum state is the l_1 norm of the associated quasi-probability distribution. Additionally, there is a quantity, $\mathcal{N}$ (the negativity), associated with the entire quantum circuit and this can be small even if the quantum state or POVM element negativity is extremely large. We showed that the circuit negativity can be efficiently classically computed. Furthermore, one can classically estimate any Born rule probability or marginal probability associated with the circuit, to within $\pm\epsilon$, with high probability, in run-time $O(poly(n, 1/\epsilon, \mathcal{N}))$. Thus, for the set of quantum circuit families $C_{poly\mathcal{N}}$ where the negativity of each circuit is upper bounded by a polynomial in the number of qudits, our algorithm produces an efficient additive polynomial precision estimate of Born rule probabilities.

The work of Veitch, Mousavian, Gottesman and Emerson [29] showed that negativity in the discrete Wigner function is a monotone for the amount of non-stabilizer resource. Thus, our work established an efficiently computable monotone which polynomially upper-bounds the classical simulation run-time of any quantum circuit.

In this thesis, we refine and extend our earlier work from Ref. [1] in several significant ways. Firstly, we extend the simulation technique from one that only applies to the mixed state formalism of quantum mechanics to the pure state formalism as well. In the mixed state formalism, n qudit quantum states are represented by density operators and live in a Hilbert space of *linear operators* from $\left(\mathbb{C}^d\right)^n$ to $\left(\mathbb{C}^d\right)^n$. In the pure state formalism, quantum states are represented by kets and live in a Hilbert space of vectors $\left(\mathbb{C}^d\right)^n$. The mathematical extension of our methods to the pure state formalism can result in a significantly faster run-time for the algorithm compared to simulation of the pure system within the mixed state formalism. In the present work, we also make several important relaxations to the mathematical construction in Ref. [1]. Here, we remove a technical condition related to the normalization of frames, move away from the dual frame formalism instead requiring a choice of only one frame and remove the Hermiticity requirement of this frame. These relaxations mean that the simulated quantum system is no longer represented by a quasi-probability distribution in the sense of Ref. [1] but a much more general object, an un-normalized, complex valued field. The advantage of these relaxations is increased flexibility resulting in faster run-times combined with a less constrained mathematical structure.

Part I of this thesis is structure as follows. In Ch. 3 we review some useful background material. In Ch. 4 we present the general framework for Born probability estimation. In Sec. 4.1 we discuss some preliminary details covering the scope the simulation and some introductory concepts. In Sec. 4.2, we present the general framework for Born rule probability estimation along with a number of examples. In Sec. 4.3, we present an overview of simulation algorithms from the recent literature and discuss these from the perspective of our general framework. In Ch. 5 we apply our general framework to the construction of an estimation algorithm for a universal gate-set. This algorithm has exponential run-time in the number of T gates, scales polynomially in all other parameters and is intended to be the state of the art method for inverse polynomial precision Born rule probability estimation.

Chapter 3

Background and notation

3.1 Notation

$\mathbb{N}$ represents the natural numbers including zero. For $n \in \mathbb{N}\backslash\{\,0\,\}$, $[n]$ represents the set of positive integers $\{\,1, 2, \ldots, n\,\}$. $\mathbb{Z}$ represents the integers and for $n \in \mathbb{N}\backslash\{\,0\,\}$, we use $\mathbb{Z}_n = \{\,0, 1, \ldots, n-1\,\}$ to refer to the quotient ring $\mathbb{Z}/n\mathbb{Z}$.

For x a vector and j a positive integer, x_j will represent the j^{th} entry of x unless specified otherwise. Products of non-commuting objects will be indexed ascending from left to right i.e.

$$\prod_{i \in [n]} g_i = \prod_{i=1}^{n} g_i := g_1 \times g_2 \times \ldots \times g_n. \tag{3.1}$$

Products in the reverse ordering will be written:

$$\prod_{i=n}^{1} g_i := g_n \times g_{n-1} \times \ldots \times g_1. \tag{3.2}$$

We will apply this convention to tensor products, Kronecker products as well as matrix products.

The convex polytope of all probability distributions (non-negative vectors with l_1-norm of unity) over a finite set S are represented by $\mathbb{P}_S$. $\mathcal{U}(S) \in \mathbb{P}_S$ will be used to represent the uniform distribution over $|S|$ elements and my be written as $\mathcal{U}$ where S is clear from the context.

3.2 Commonly used gates

The non-identity, qubit Pauli operator are written X, Y, Z and can be written in the computational basis as:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{3.3}$$

The set of Clifford gates acting on qubits can be generated from the single qubit gates referred to as the Hadamard gate (H); the phase gate (S) and the two qubit controlled-NOT gate (CX or CNOT). Other commonly used gates include the controlled-Z gate (CZ) and the swap gate (SWAP) which is also a Clifford gate and the T gate (T) which promotes the Clifford gates to universality. Written in matrix form, in the computational basis, these are:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \qquad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \tag{3.4}$$

and:

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \qquad SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

3.3 The qudit stabilizer sub-theory

We present some elementary mathematical objects related to the stabilizer sub-theory for *qudits* [49]. Throughout this section, we assume that d is a prime, although everything can be extended to arbitrary integer d. First, the qudit generalization of the Pauli X and Z operators are defined:

$$X\left|j\right\rangle = \left|j+1 \pmod{d}\right\rangle, \qquad Z\left|j\right\rangle = \omega^j \left|j\right\rangle, \tag{3.6}$$

where d is the dimensionality of the qudits, $\omega := \exp(\frac{2\pi i}{d})$ and $\left|j\right\rangle$, for $j \in \mathbb{Z}_d$, forms an orthonormal basis of the qudit state space known as the computational basis.

We note that $ZX = \omega XZ$. For all prime d, the X and Z operators are unitary. However, unlike the $d = 2$ case, for $d > 2$, they are not Hermitian.

We will use the notation X_j or Z_j to represent the X and Z operators acting on the j^{th} qubit and padded with identitites. We will also apply this notation to other single particle and two particle operators acting on an n particle system.

We write the qudit generalization of the controlled-NOT gate as CX. Specifically, $\text{CX}_{i,j}$ represents the CX gate acting non-trivially on qudits i and j where qudit i is the controll and qudit j is the target. The action of CX is defined by the linear extension of it's action on the computational basis states:

$$\text{CX}_{1,2}\left|a\right\rangle\left|b\right\rangle = \left|a\right\rangle\left|a \oplus b\right\rangle, \tag{3.7}$$

where "$\oplus$" is the sum modulo d. Alternatively, it can be written:

$$\text{CX}_{1,2} = \sum_{k=0}^{d-1} \Pi(k) \otimes X^k, \tag{3.8}$$

where $\Pi(k) = |k\rangle\langle k|$ acts only on one qudit and is the projector onto the $(k+1)^{\text{th}}$ computational basis state.

The single particle Weyl-Heisenberg displacement operators are defined as follows:

$$\mathbf{D}_d := \{D_{z,x} = \omega^{-2^{-1}zx} Z^z X^x | x, z \in \mathbb{Z}_d\}, \tag{3.9}$$

where, for d odd, 2^{-1} is the multiplicative inverse of 2 in the finite field $\mathbb{Z}_d$ while for $d = 2$, $2^{-1} := \frac{1}{2} \in \mathbb{R}$. The qubit Pauli operators $\{ I, X, Y, Z \}$ can be regarded as a special case of the Weyl-Heisenberg displacement operators for $d = 2$.

We will use the compact notation X^x and Z^z to represent strings of generalized Pauli operators. More specifically, for $x, z \in \mathbb{Z}_d^n$, we define:

$$X^x := \otimes_{j=1}^n X^{x_j}, \qquad Z^z := \otimes_{j=1}^n Z^{z_j}, \tag{3.10}$$

where $X^0 = Z^0 = I$. The set of multi-particle Weyl-Heisenberg displacement operators is then written as:

$$\mathbf{D}_d^{\otimes n} = \{D_{z,x} = \omega^{-2^{-1}zx} Z^z X^x | x, z \in \mathbb{Z}_d^n\}. \tag{3.11}$$

For vectors $\alpha_z, \alpha_x, \beta_z, \beta_x \in \mathbb{Z}_d^n$, we represent the *symplectic product* between the two vectors $\alpha := (\alpha_z, \alpha_x)^T$ and $\beta := (\beta_z, \beta_x)^T$ in $\mathbb{Z}_d^{2n}$ by the notation $[\alpha, \beta] := \alpha_z \cdot \beta_x - \beta_z \cdot \alpha_x$. Using this notation, the product of two multi-particle Weyl-Heisenberg displacement operators is given by:

$$D_{\alpha_z,\alpha_x} D_{\beta_z,\beta_x} = \omega^{2^{-1}[\alpha,\beta]} D_{\alpha_z+\beta_z,\alpha_x+\beta_x}, \tag{3.12}$$

where $D_\alpha =: D_{\alpha_z,\alpha_x}$. These operators satisfy:

$$\operatorname{tr}(D_\alpha) = d^n \delta_{0,\alpha}. \tag{3.13}$$

Thus, with a normalization of $d^{-n/2}$, these operators form an orthonormal basis (of the space of linear operators from $\left(\mathbb{C}^d\right)^n$ to itself) with respect to the trace inner product. That is:

$$\frac{1}{d^n} \operatorname{tr}\left(D_\alpha^\dagger D_\beta\right) = \delta_{0,\alpha-\beta}. \tag{3.14}$$

The Clifford group $\mathbf{C}_{d,n}$ is defined to be the normalizer of the group $\langle \mathbf{D}_d^{\otimes n} \rangle$, that is:

$$\mathbf{C}_{d,n} = \{ U \in \mathcal{U}(d^n) \mid U \langle \mathbf{D}_d^{\otimes n} \rangle U^\dagger = \langle \mathbf{D}_d^{\otimes n} \rangle \} , \tag{3.15}$$

where $\langle \mathbf{D}_d^{\otimes n} \rangle$ is the group generated by the set of displacement operators $\mathbf{D}_d^{\otimes n}$. The set of stabilizer states is the image of the computational basis under the Clifford group $\mathbf{C}_{d,n}$. The stabilizer polytope is the convex hull of the set of stabilizer states. The stabilizer sub-theory is then the set of preparations of states in the stabilizer polytope, stabilizer measurements and convex combinations of Clifford transformations.

3.4 Discrete Wigner functions and negativity

A discrete Wigner function is defined over a phase space $\mathbb{Z}_d^2$ via the phase-point operators

$$\begin{aligned} A_{0,0} &= \frac{1}{d} \sum_{x,z \in \mathbb{Z}_d} D_{x,z}, \\ A_{x,z} &= D_{x,z} A_{0,0} D_{x,z}^{\dagger}\,, \end{aligned} \tag{3.16}$$

which form a trace-orthogonal basis for the space of bounded linear operators acting on $\mathbb{C}^d$, that is:

$$\begin{aligned} \operatorname{tr}(A_\alpha) &= 1, \\ \operatorname{tr}(A_\alpha A_\beta) &= d\delta_{\alpha,\beta}, \end{aligned} \tag{3.17}$$

where we write $\alpha = (x, z) \in \mathbb{Z}_d^2$ for brevity.

The multi-particle phase point operators can be constructed as the tensor product of the single particle phase point operators or alternatively can be written in terms of the multi-particle Weyl-Heisenberg operators:

$$\begin{aligned} A_{0,0} &= \frac{1}{d^n} \sum_{x,z \in \mathbb{Z}_d^n} D_{x,z}, \\ A_{x,z} &= D_{x,z} A_{0,0} D_{x,z}^{\dagger}\,, \end{aligned} \tag{3.18}$$

which still form a trace-orthogonal basis for the space of bounded linear operators acting on $\left(\mathbb{C}^d\right)^n$ such that:

$$\begin{aligned} \operatorname{tr}(A_\alpha) &= 1, \\ \operatorname{tr}(A_\alpha A_\beta) &= d^n\delta_{\alpha,\beta}. \end{aligned} \tag{3.19}$$

A discrete Wigner function of an n-qudit state ρ, unitary U and positive operator valued measurement (POVM) effect E can be defined as follows:

$$\begin{aligned} W_\rho(\alpha) &= \frac{1}{d^n} \operatorname{tr}(\rho A_\alpha)\,, \\ W_U(\beta|\alpha) &= \frac{1}{d^n} \operatorname{tr}\left(A_\beta U A_\alpha U^\dagger\right), \\ W(E|\alpha) &= \operatorname{tr}(E A_\alpha)\,, \end{aligned} \tag{3.20}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_d^{2n}$, $\alpha_j \in \mathbb{Z}_d^2$ and $A_\alpha = \otimes_{j=1}^n A_{\alpha_j}$. The trace-orthogonality of the phase-point operators can then be used to show that

$$W_{U_2 U_1}(\beta|\alpha) = \sum_\gamma W_{U_2}(\beta|\gamma) W_{U_1}(\gamma|\alpha)\,. \tag{3.21}$$

The Born rule, which determines the probability $\Pr(E|\rho, U)$ of measuring an effect E given a quantum state ρ evolved according to a unitary U, can be written as

$$\begin{aligned}\Pr(E|\rho, U) &= \mathrm{tr}\left(U\rho U^\dagger E\right) \\ &= \sum_{\alpha,\beta\in\mathbb{Z}_d^{2n}} W(E|\beta) W_U(\beta|\alpha) W_\rho(\alpha)\,. \end{aligned} \tag{3.22}$$

While a Wigner function of a generic quantum state ρ or effect E has negative values, the negativity is bounded by the eigenvalues of the A_α, which are all ±1 [51]. Since the A_α are Hermitian, we have:

$$\begin{aligned}|W_\rho(\alpha)| &\le d^{-n}, \\ |W_U(\beta|\alpha)| &\le d^{-n}, \\ |W(E|\alpha)| &\le 1, \end{aligned} \tag{3.23}$$

for all α, β, ρ, U and E.

In odd-prime-dimension, one special property of the discrete Wigner function is that every ρ, E or U in the stabilizer sub-theory has a non-negative Wigner function [49]. In fact, there are also some mixed states (and the corresponding effects) that have non-negative Wigner function and are not convex combinations of stabilizer states, so simulations based upon the discrete Wigner function provide strict generalizations of the Gottesman-Knill theorem in odd-prime-dimension [22]. In even dimension, whether a stabilizer state is negative or not depends on the choice of phase convention used in defining the Wigner function.

Chapter 4

A general framework for Born probability estimation

In this chapter, we present a generalized framework for the construction of Born rule probability estimators. Through this generalized framework, we consolidate a number of known results in the existing literature and bring into focus the interesting and pressing open questions in the field.

4.1 Preliminaries

We begin with an overview of some of the key concepts and background needed to proceed.

4.1.1 Estimators and sampling methods

An *estimator* is a rule for computing an *estimate* of some target quantity (the *estimand*) from sampled data. A simple example is the average of a sample used to estimate the population mean. In our setting, producing an estimate will require both an estimator and the specification of a technique for sampling from some desired probability distribution.

In Born rule probability estimation, a commonly used approach to the sampling component is the Monte Carlo method [1, 34, 33]. Other examples of sampling techniques such as *rejection sampling*, *pseudo-random number sampling* and *inverse transform sampling* may also find application in Born rule probability estimation and can sometimes be used in conjunction with the Monte Carlo method.

Let p be an unknown parameter we wish to estimate, e.g., a Born rule probability. In the Monte Carlo based approach of Ref. [1], p is estimated by observing a number of random variables $X_1, \ldots, X_s$ generated through a Markov Chain Monte Carlo sampling procedure and computing some function of the outcomes $\hat{p}_s(X_1, \ldots, X_s)$, chosen so that $\hat{p}_s$ is close to p in expectation. In this case, $\hat{p}_s$ is an estimator of p.

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

$$\Pr\big(|p - \hat{p}_s| \geq \epsilon\big) \leq \delta\,, \qquad \text{additive } (\epsilon, \delta)\text{-precision.} \tag{4.1}$$

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

$$\Pr\big(|p-\hat{p}_s|\geq \epsilon p\big)\leq \delta\,,\qquad \text{multiplicative } (\epsilon,\delta)\text{-precision.} \tag{4.2}$$

In the case where $p \leq 1$ is a probability, a multiplicative precision estimator is more accurate than an additive precision estimator for any given (ϵ, δ).

For estimation using sampling methods based on Monte Carlo, 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. More generally, s may represent some resource invested in computing the estimator $\hat{p}_s$ such as the computation run-time. For this reason, we may wish to classify additive/multiplicative (ϵ,δ)-precision estimators by how s scales with $1/\epsilon$ and $1/\delta$. We say that $\hat{p}_s$ is an additive (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 an additive (multiplicative) (ϵ,δ)-precision estimator for all $s\geq f(\epsilon^{-1},\log\ \delta^{-1})$.

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 (i.e. $\mathbb{E}(\hat{p}_1)=p$). Let $\hat{p}_s$ be defined as the average of s independent observations of $\hat{p}_1$. Then, by the Hoeffding inequality [52], we have:

$$\Pr\big(|p-\hat{p}_s|\geq \epsilon\big)\leq 2\exp\Big(\frac{-2s\epsilon^2}{(b-a)^2}\Big)\,, \tag{4.3}$$

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 (ϵ,δ)-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.

4.1.2 The circuits we consider

The quantum systems we will consider will be multi-particle systems each with d levels (qudits) where d can be any positive integer (including 2). We will consider quantum circuits both in the pure state and the mixed state setting.

In the pure state setting, we consider an n qudit system initially in a computational basis state. This state is acted on by a sequence of L unitaries $U_1,\ldots,U_L$ each of which acts non-trivially on at most two qudits. All of the qudits are then simultaneously measured using a projective measurement. The case where an arbitrary subset of qubits is measured also fits within this framework since our algorithm is required to estimate all *marginal* probabilities as well as probabilities for n-qubit measurement outcomes.

In the mixed state setting, we consider quantum circuits on n qudits initially in a product of qudit density states. This state is acted on by a sequence of L completely positive trace preserving (CPTP) maps $\mathcal{E}_1,\ldots,\mathcal{E}_L$ which act non-trivially on at most two qudits. All of the qudits are then simultaneously measured using a positive operator valued measurement (POVM) $\mathcal{M}=\{\,E_{\vec{x}}\mid \vec{x}\in\mathbb{Z}_d^n\,\}$ constructed from a product of local POVMs $\mathcal{M}_i=\{\,E_{x_i}\mid x_i\in\mathbb{Z}_d\,\}$ where, $E_{\vec{x}}=E_{(x_1,\ldots,x_n)}=\bigotimes_{i=1}^n E_{x_i}$.

We note that any n qudit quantum circuit consistent with the above can be accurately described using at most $poly(n)$ bits of information.

4.1.3 Born rule probabilities

In the pure state formalism, we consider the description $c = \{| \psi\rangle, U, \mathcal{M}\}$ of some ideal quantum circuit, with $|\psi\rangle$ an initial state, $U = U_L \times U_{L-1} \times \cdots \times U_1$ a sequence of unitaries, and $\mathcal{M} = \{ \Pi_x \mid x \in \mathbb{Z}_d^n \}$ a set of rank one projectors where each projector is formed by product of projections onto each qudit. That is:

$$\Pi_x = \prod_{j \in [n]} \Pi_{x_j}^{(j)}, \tag{4.4}$$

where $\Pi_{x_j}^{(j)}$ is the projector onto the measurement outcome $x_j \in \mathbb{Z}_d$ for qudit j padded with identities acting on all other qudits. The Born rule gives us the exact quantum predictions associated with observing any particular outcome x:

$$\begin{aligned} \mathcal{P}(x) :&= \langle\psi| U^\dagger \Pi_x U |\psi\rangle && (4.5)\\ &= \operatorname{tr}\left(U |\psi\rangle\langle\psi| U^\dagger \Pi_x\right). && (4.6) \end{aligned}$$

In the mixed state formalism, we consider the description $c = \{ \rho, \mathcal{E}, \mathcal{M} \}$ of some ideal quantum circuit, with ρ an initial state, $\mathcal{E} = \mathcal{E}_L \circ \mathcal{E}_{L-1} \circ \cdots \circ \mathcal{E}_1$ a sequence of CPTP maps, and $\mathcal{M} = \{E_x \mid x \in \mathbb{Z}_d^n\}$ a set of measurement operators each associated with the measurement outcome $x \in \mathbb{Z}_d^n$. The Born rule gives us the exact quantum predictions associated with observing any particular outcome x:

$$\mathcal{P}(x) := \operatorname{tr}\left(\mathcal{E}(\rho) E_x\right). \tag{4.7}$$

We will refer to the map $\mathcal{P} : x \mapsto \mathcal{P}(x)$ as the probability distribution $\mathcal{P}$. Sometimes we will refer to such a probability distribution by $\mathcal{P}_c$. Here, c is the description of the quantum circuit which provides an implicit description of the probability distribution $\mathcal{P}_c$. We will omit the subscript when the probability distribution is clear from the context.

In both the pure and mixed state formalism, the marginal probabilities associated with events $(\mathcal{W}, x) \equiv S$ are given by:

$$\mathcal{P}(S) := \sum_{y \in \mathbb{Z}_d^{n-w}} \operatorname{tr}\left(\mathcal{E}(\rho) E_{(x,y)_\mathcal{W}}\right) \tag{4.8}$$

where we have used the notation $(x, y)_\mathcal{W} =: z$ to represent a vector formed by interlacing the entries from vectors $x \in \mathbb{Z}_d^w$ and $y \in \mathbb{Z}_d^{n-w}$ such that within $z \in \mathbb{Z}_d^n$, the entries of x each appear in the order that that they appear in x (and similarly for y) but the position of the entries of x within z is specified by $\mathcal{W} \subseteq [n]$.

4.1.4 The complexity of Born probability estimation

The task of efficiently classically estimating these probabilities with respect to general quantum circuits is of great practical interest, but is known to be hard 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.

We note that for universal families of quantum circuits, exact computation (and even multiplicative precision estimation) of Born probabilities is #P-hard. On this basis, it is unlikely that universal quantum computers can be used to estimate Born rule probabilities to these levels of precision. In contrast, we note that a universal quantum computer can be used to produce $1/poly$ additive precision estimates of Born rule probabilities associated[1] with any language in BQP. Let us demonstrate this point using an example with broader conceptual significance.

Example 1. *Suppose an agent is given access to a universal quantum computer and required to produce an estimate for any requested Born rule probability. Than given a description of a quantum circuit $c_a \in \mathcal{C}_{\text{UNIV}}$ and a description of an event $S \subseteq \mathbb{Z}_d^n$ such that membership in S can be decided with high probability, the agent can efficiently estimate the probability of the event occurring, $p = \mathcal{P}_a(S)$.*

A simple approach is to construct the estimator $\hat{p}_s$ by independently running the circuit s times. On each of the runs $i = 1, \ldots, s$, the agent computes/decides if the outcome x is in the event S (in this case, $X_i = 1$) or not in S (in this case, $X_i = 0$). We then define $\hat{p}_s = \frac{1}{s}\sum_{i=1}^{s} X_i$. Using the Hoeffding inequality, it is easy to show that the Born rule probability estimator $\hat{p}_s$ is an additive polynomial precision estimator of p. Thus, for all $c_a \in \mathcal{C}_{\text{UNIV}}$, $\epsilon, \delta > 0$, there is a choice of $s \in \mathbb{N}$ such that this procedure can be used to compute an estimate $\hat{p}$ of $p := \mathcal{P}_a(S)$ such that $\hat{p}$ satisfies the accuracy requirement:

$$\Pr\big(|p - \hat{p}| \geq \epsilon\big) \leq \delta \tag{4.9}$$

and the run-time required to compute the estimate $\hat{p}$ is $O(poly(n, \epsilon^{-1}, \log\ \delta^{-1}))$.

As a minor technicality we note that, when deciding membership in the event is non-deterministic, the chance of error can be exponentially suppressed. Thus this source of error makes an insignificant contribution to the estimator error.

4.1.5 The poly-box: generating an additive polynomial precision estimate

In this section, we define a class of classical estimation algorithms that are of practical interest. We will refer to such a class of algorithms as a *poly-box*.

For a given n qudit circuit, let $\mathcal{W} \subseteq [n]$ be a subset of qudits. We will be interested in the events where there is a fixed measurement outcome of the qudits in $\mathcal{W}$ and the measurement outcomes of the remaining qudits are ignored. For a fixed $\mathcal{W}$, these events can be indexed by a string $x \in \mathbb{Z}_d^w$ where $w := |\mathcal{W}|$. Thus, these events can be represented as the pair $(\mathcal{W}, x)$.

[1] A Born rule probability is a probabilities of observing an event when we run a particular quantum circuit. However an event can be defined as the set of all length n strings in a language. In this way, a language in BQP can be viewed as sequence of events decidable in BQP. The specification of a quantum circuit specifies the event of interest through n, the number of measured qubits.

Alternatively, we can represent these events by the string $S \in (\mathbb{Z}_d \cup \{\bullet\})^n$ where an entry of "$\bullet$" in the j^{th} position of the string indicates that the j^{th} qudit is not in $\mathcal{W}$ and hence it's measurement outcome can be ignored while an entry $b \in \mathbb{Z}_d$ imposes the restriction that the measurement outcome of the qudit must be b.

We will focus on estimating the probability p of observing the event S for some given $S \in (\mathbb{Z}_d \cup \{\bullet\})^n$. We point out that the set of allowed events we are considering is significantly restricted. In particular, for fixed n, the number of allowed events is $|(\mathbb{Z}_d \cup \{\bullet\})^n| = (d+1)^n$. In contrast, the total number of events that can be defined on n qudit measurements is the number of subsets of $\mathbb{Z}_d^n$ which is 2^{d^n}. Our Born rule probability estimate will be denoted by $\hat{p}$. We are interested in estimation algorithms that satisfy the following accuracy requirement: for all $\epsilon, \delta > 0$, the algorithm outputs an additive (ϵ, δ)-precision estimate in run-time $O(poly(n, \epsilon^{-1}, log\ \delta^{-1}))$.

Let us now more precisely define a notion of simulation we call a *poly-box.*

Definition 1. (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, for all $a \in \mathcal{A}^*, \epsilon, \delta > 0$ and $S \in \{0, 1, \bullet\}^n$, can be used to compute an estimate $\hat{p}$ of $\mathcal{P}_a(S)$ such that $\hat{p}$ satisfies the accuracy requirement:*

$$\Pr\big(|p - \hat{p}| \geq \epsilon\big) \leq \delta \tag{4.10}$$

and, the run-time required to compute the estimate $\hat{p}$ is $O(poly(n, \epsilon^{-1}, \log\ \delta^{-1}))$.

Eq. (4.10), gives an upper bound on the probability that the computed estimate, $\hat{p}$, is far from the target quantity. This probability is over the potential randomness in the process used to generate the estimate $\hat{p}$. For simplicity, we additionally require that the output of a poly-box is independent of prior output. In particular, let $\alpha = (a, \epsilon, \delta, S)$ be an input into a poly-box and $\hat{p}_\alpha$ the observed output. Then, we require that the probability distribution of $\hat{p}_\alpha$ only depends on the choice of input α and in particular is independent of prior output.

4.2 The general framework for estimation

In this section, we outline a general framework that can be applied to construct a poly-box for a given family of quantum circuits. This framework is a generalization of our earlier work [1] and broadly uses the approach outlined in Sec. 4.1.1 where samples from some probability distribution are used to compute an estimate. Within this framework, a number of parameters are free choices that select a particular estimation algorithm from a family of possible algorithms. These free parameters will be, collectively referred to as the estimation *model.* We first discuss the choice of model.

4.2.1 The estimation model

In order to specify a Born rule probability estimation algorithm, we must fix a choice of the following mathematical objects:

1. Fix a Hilbert space $\mathcal{H}^n$ representing the quantum state space of the system. For pure quantum systems of n qudits, $\mathcal{H}^n$ can be chosen to be $\left(\mathbb{C}^d\right)^n$. Alternatively, for mixed quantum systems of n qudits, $\mathcal{H}^n$ can be chosen to be the set of linear maps $\left(\mathbb{C}^d\right)^n \to \left(\mathbb{C}^d\right)^n$.

2. Fix a base field $\mathbb{K} \in \{\,\mathbb{R}, \mathbb{C}\,\}$.

3. For $M \geq dim(\mathcal{H}^n)$, fix an ordered set of elements $\mathcal{B} := [\mathcal{B}_1, \dots, \mathcal{B}_M]^T$ with $\mathcal{B}_j \in \mathcal{H}^n$ for $j \in [M]$ such that:

$$\underset{\mathbb{C}}{\operatorname{span}}\ \{\,\mathcal{B}_1, \dots, \mathcal{B}_M\,\} = \mathcal{H}^n \tag{4.11}$$

   i.e. the $\mathbb{C}$ span of $\mathcal{B}$ contains $\mathcal{H}^n$. We will call $\mathcal{B}$ a *frame* for $\mathcal{H}^n$. The frame may be overcomplete. In this case $M > dim(\mathcal{H}^n)$ and the frame is not linearly independent.

4. Fix a map $\mathcal{P}^{\mathcal{B}} : \mathcal{H}^n \to \mathbb{R}^M$ such that for all $H \in \mathcal{H}^n$, $\mathcal{P}^{\mathcal{B}}(H)$ is a probability distribution over the set $[M] := \{\,1, \dots, M\,\}$. This step defines the probability distribution we will be sampling from in the Monte Carlo procedure. The map $\mathcal{P}^{\mathcal{B}}$ will not be linear. We will discuss a number of examples shortly.

5. Fix a complementary map $\mathcal{F}^{\mathcal{B}} : \mathcal{H}^n \to \mathbb{K}^M$ such that for all $H \in \mathcal{H}^n$:

$$\underset{X \sim \mathcal{P}^{\mathcal{B}}(H)}{\mathbb{E}} \left[\mathcal{F}^{\mathcal{B}}_X(H)\mathcal{B}_X\right] = H. \tag{4.12}$$

   That is, the vector $H \in \mathcal{H}^n$ is the expectation value of sampling a random variable X from the distribution $\mathcal{P}^{\mathcal{B}}(H)$ then computing $\mathcal{F}^{\mathcal{B}}_X(H)\mathcal{B}_X$, the pointwise product of the X^{th} row of $\mathcal{F}^{\mathcal{B}}(H)$ with the X^{th} row of $\mathcal{B}$.

We will omit the $\mathcal{B}$ superscript from the maps $\mathcal{P}^{\mathcal{B}}$ and $\mathcal{F}^{\mathcal{B}}$ when the frame is clear from context.

We will also refer to the triple $(\mathcal{B}, \mathcal{P}, \mathcal{F})$ as the estimation model when the Hilbert space and base field are clear from the context. The model will need to satisfy a number of additional efficiency constraints for the estimation algorithm to execute efficiently. We postpone discussion on the efficiency constraints for now and return to these in Sec. 4.2.5. Instead, we now briefly comment on the purpose and use of the objects defined above.

We aim to produce an estimate of the Born rule probability associated with some quantum circuit and event. In doing so, our algorithm will need to first represent the initial state of the system. Then, this representation needs to be updated with each transformation that the quantum system undergoes up until immediately prior to the final measurement. Finally, this representation will be updated due to the final measurement.

The elements of the frame $\mathcal{B}$ will be used to represent the quantum state of the system we wish to simulate. After each transformation is applied to the quantum system, we will use a single element of $\mathcal{B}$ and a single number in $\mathbb{K}$ to represent the quantum state. Specifically, after the k^{th} transformation, we will represent the quantum state of the system by some element $\mathcal{B}_{x_k}$ and some number $\mathcal{F}^{\mathcal{B}}_{x_k}$. For non-trivial examples, we do not expect the quantum state after the k^{th} transformation to be a scalar multiple of $\mathcal{B}_{x_k}$. However, the quantum state is some linear combination of the vectors $\mathcal{B}_{x_k}$ for $x_k \in [M]$. A crucial observation is that the state can be "on average" accurately represented by $\mathcal{F}^{\mathcal{B}}_{x_k}\mathcal{B}_{x_k}$, if we choose x_k probabilistically. A typical quantum state can only be represented as a linear combination of *exponentially* many elements from $\mathcal{B}$. Specifying the weights in this linear combination immediately precludes an efficient simulation. The simulation procedure we describe avoids this by only ever retaining in memory a single index $x_k \in [M]$ that specifies a part of the information required to describe the full linear combination, namely the element of $\mathcal{B}$ and $\mathcal{F}^{\mathcal{B}}$ that is relevant. The remainder of the information specifying the linear combination is encoded in the probability distribution from which the value x_k is sampled. Eq. (4.12) imposes a requirement on our model choice ensuring that our representation of the quantum state is "on average" accurate. To ensure that Eq. (4.12) is satisfied, we must choose $\mathcal{F}^{\mathcal{B}}_{x_k}$ and $\mathcal{P}^{\mathcal{B}}_{x_k}$ such that their product is the weight of the coefficient of $\mathcal{B}_{x_k}$ in a linear expansion of the quantum state.

Since the quantum state lives in a high dimensional vector space, this simulation algorithm avoids some inefficiency pitfalls by representing this state with the pair $(\mathcal{B}_x, \mathcal{F}^{\mathcal{B}}_x)$. However, it is also crucial that one can efficiently manipulate the elements of the frame $\mathcal{B}$ as required by the estimation algorithm. We will present the estimation algorithm later and also discuss these efficiency requirements. For now, we note that without the ability to efficiently manipulate these objects our representation of the quantum state by the pair $(\mathcal{B}_x, \mathcal{F}^{\mathcal{B}}_x)$ is potentially no more useful than its representation as a quantum circuit.

In this section we described a set of abstract objects (the estimation model) and conveyed a rough intuition for the role of these objects in our estimation algorithm. In Sec. 4.2.3 we will concretely state how an estimation algorithm can be constructed based on a given model. Before proceeding to the estimation algorithm, we present some examples of estimation models.

4.2.2 Examples of estimation models

We now present a number of examples for the choice of model. We will start with two simple examples. The first will present a simple choice of model in the pure state formalism and the next will be an example of a simple model within the mixed state formalism. Following these, we will present two substantially more general examples. One of these within the pure and the other applying to the mixed state formalism.

Let us first consider a simple example based on the pure state formalism.

Example 2. (pure state formalism - simple)
Let $\mathbb{K} = \mathbb{C}$, $\mathcal{H}^n = \left(\mathbb{C}^d\right)^n$ and set the frame to be the computational basis, $\mathcal{B} = \{ \ |x\rangle \ \mid \ x \in \mathbb{Z}^n_d \}$. We define the maps $\mathcal{P} : \mathcal{H}^n \to \mathbb{R}^{d^n}$ and $\mathcal{F} : \mathcal{H}^n \to \mathbb{C}^{d^n}$ entry-wise as follows:

$$\mathcal{P}_x(|\psi\rangle) := \frac{|\langle x|\,\psi\rangle|}{\mathcal{N}}, \qquad \mathcal{F}_x(|\psi\rangle) := \mathcal{N} \exp\left(i \operatorname{Arg}(\langle x|\,\psi\rangle)\right), \tag{4.13}$$

where $\mathcal{N} := \sum_{x\in\mathbb{Z}^n_d} |\langle x|\,\psi\rangle|$ and we have chosen the strings $x \in \mathbb{Z}^n_d$ as the more natural index for the entries of $\mathcal{P}$ and $\mathcal{F}$.

We note that $\mathcal{B}$ spans the Hilbert space; for all $|\psi\rangle \in \mathcal{H}^n$, $\mathcal{P}(\psi)$ is a valid probability distribution *over $x \in \mathbb{Z}_d^n$; $\mathcal{F}_x(|\psi\rangle)$ is a scalar and Eq. (4.12) is satisfied:*

$$\begin{aligned} \mathop{\mathbb{E}}_{X\sim\mathcal{P}(|\psi\rangle)}\left[\mathcal{F}_X(|\psi\rangle)\mathcal{B}_X\right] &= \sum_x \mathcal{P}_x(|\psi\rangle)\mathcal{F}_x(|\psi\rangle)\mathcal{B}_x &(4.14)\\ &= \sum_x \langle x|\,\psi\rangle\,|x\rangle &(4.15)\\ &= |\psi\rangle\,. &(4.16)\end{aligned}$$

We now present an example based on the mixed state formalism.

Example 3. (mixed state formalism - simple)
Let $\mathbb{K} = \mathbb{C}$, $\mathcal{H}^n$ be the set of linear maps from $\left(\mathbb{C}^d\right)^n$ to itself and set the frame to be proportional to the Weyl-Heisenberg operators defined in Sec. 3.3, i.e. $\mathcal{B} = \{\, \frac{1}{d^n}D_{z,x} \;\mid\; x,z\in\mathbb{Z}_d^n \,\}$. We define the maps $\mathcal{P}:\mathcal{H}^n\to\mathbb{R}^{d^{2n}}$ and $\mathcal{F}:\mathcal{H}^n\to\mathbb{C}^{d^{2n}}$ entry-wise as follows:

$$\mathcal{P}_{x,z}(H) := \frac{\left|\operatorname{tr}\left(H^\dagger\mathcal{B}_{x,z}\right)\right|}{\mathcal{N}}, \qquad \mathcal{F}_{x,z}(\rho) := \mathcal{N}\exp\left(i\operatorname{Arg}(\operatorname{tr}\left(H^\dagger\mathcal{B}_{x,z}\right))\right), \tag{4.17}$$

where $\mathcal{N} := \sum_{x,z}\left|\operatorname{tr}\left(H^\dagger\mathcal{B}_{x,z}\right)\right|$ and we have used the string pair, $(x,z)\in\mathbb{Z}_d^{2n}$, as the index for the entries of $\mathcal{P}$ and $\mathcal{F}$.

Denoting the Hadamard product (entry-wise product) by " $*$ ", we note that $\mathcal{P}(H) * \mathcal{F}(H)$ is known as the characteristic function *of H [49]. The frame, $\mathcal{B}$, spans the Hilbert space; for all $H\in\mathcal{H}^n$, $\mathcal{P}(H)$ is a valid probability distribution over $(x,z)\in\mathbb{Z}_d^{2n}$; $\mathcal{F}_{x,z}(H)$ is a scalar and by the application of Eq. (3.14), Eq. (4.12) is satisfied:*

$$\begin{aligned} \mathop{\mathbb{E}}_{X\sim\mathcal{P}(H)}\left[\mathcal{F}_X(H)\mathcal{B}_X\right] &= \sum_{(x,z)} \mathcal{P}_{x,z}(H)\mathcal{F}_{x,z}(H)\mathcal{B}_{x,z} &(4.18)\\ &= \sum_{(x,z)} \operatorname{tr}\left(H^\dagger\mathcal{B}_{x,z}\right)\mathcal{B}_{x,z} &(4.19)\\ &= H. &(4.20)\end{aligned}$$

We note that for qubits ($d = 2$), this frame is Hermitian hence $\mathcal{F}_X(H)\in\mathbb{R}$ for all Hermitian H.

We now present a more general choice of model. Examples 2 and 3 can be generated as special cases of these more general examples of models.

Example 4. (pure state formalism - general)
Let $\mathbb{K} = \mathbb{C}$ and $\mathcal{H}^n = \left(\mathbb{C}^d\right)^n$. Let G be a group and for $g\in G$, let $U_g\in SU(d^n)$ be a unitary representation of G and let $\alpha : G\to\mathbb{C}$ such that:

$$\sum_{g\in G}\alpha_g U_g\,|0^n\rangle\!\langle 0^n|\,U_g^\dagger = I. \tag{4.21}$$

Set $\mathcal{B} = \{\, U_g \;\mid 0^n\rangle \;\mid\; g\in G \,\}$.

We define the maps $\mathcal{P} : \mathcal{H}^n \to \mathbb{R}^{d^n}$ and $\mathcal{F} : \mathcal{H}^n \to \mathbb{C}^{d^n}$ entry-wise as follows:

$$\mathcal{P}_g(|\psi\rangle) := \frac{\left|\alpha_g \langle 0^n| U_g^\dagger |\psi\rangle\right|}{\mathcal{N}}, \qquad \mathcal{F}_g(|\psi\rangle) := \mathcal{N} \exp\left(i \operatorname{Arg}(\alpha_g \langle 0^n| U_g^\dagger |\psi\rangle)\right), \tag{4.22}$$

where $\mathcal{N} := \sum_{g\in G} \left|\alpha_g \langle 0^n| U_g^\dagger |\psi\rangle\right|$ and we have chosen the elements of G as the more natural index for the entries of $\mathcal{P}$ and $\mathcal{F}$.

Eq. (4.21) *insures that $\mathcal{B}$ is spanning. To show this, we will show that $|\phi\rangle$, an arbitrary element of $\mathcal{H}^n$, can be written as a linear combination of the frame:*

$$\begin{aligned} |\phi\rangle &= I |\phi\rangle && (4.23)\\ &= \sum_{g\in G} \alpha_g U_g |0^n\rangle\!\langle 0^n| U_g^\dagger |\phi\rangle && (4.24)\\ &= \sum_{g\in G} \left(\alpha_g \langle 0^n| U_g^\dagger |\phi\rangle\right) U_g |0^n\rangle . && (4.25) \end{aligned}$$

We note that for all $|\psi\rangle \in \mathcal{H}^n$, $\mathcal{P}(|\psi\rangle)$ is a valid probability distribution over $g \in G$ and $\mathcal{F}_g(|\psi\rangle)$ is a scalar. Using Eq. (4.21)*, we show that Eq.* (4.12) *is satisfied:*

$$\begin{aligned} \mathop{\mathbb{E}}_{X\sim\mathcal{P}(|\psi\rangle)} [\mathcal{F}_X(|\psi\rangle)\mathcal{B}_X] &= \sum_x \mathcal{P}_g(|\psi\rangle)\mathcal{F}_g(|\psi\rangle)\mathcal{B}_g && (4.26)\\ &= \sum_g \alpha_g \langle 0| U_g^\dagger |\psi\rangle U_g |0\rangle && (4.27)\\ &= \sum_g \left[\alpha_g U_g |0\rangle\!\langle 0| U_g^\dagger\right] |\psi\rangle && (4.28)\\ &= |\psi\rangle . && (4.29) \end{aligned}$$

As a simple special case of Example 4, we can choose $G = \mathbb{Z}_d^n$ with $U_x = X^x$. This choice recovers Example 2.

We now present an example based on the mixed state formalism.

Example 5. (mixed state formalism - general) *Let $\mathbb{K} = \mathbb{C}$ and $\mathcal{H}^n$ be the set of linear maps from $\left(\mathbb{C}^d\right)^n$ to itself. Let G be a group and for $g \in G$, let $U_g \in SU(d^n)$ be a unitary representation of G; let $\alpha : \mathcal{H}^n \times G \times G \to \mathbb{C}$ be a map and $\mathcal{O} \in \mathcal{H}^n$ a fiducial operator such that for all $H \in \mathcal{H}^n$:*

$$\sum_{(g,h)\in G\times G} \alpha_{g,h}(H) U_g \mathcal{O} U_h^\dagger = H. \tag{4.30}$$

We set the frame to be $\mathcal{B} = \left\{U_g \mathcal{O} U_h^\dagger \mid g, h \in G\right\}$.

We define the maps $\mathcal{P} : \mathcal{H}^n \to \mathbb{R}^{d^{2n}}$ and $\mathcal{F} : \mathcal{H}^n \to \mathbb{C}^{d^{2n}}$ entry-wise as follows:

$$\mathcal{P}_{g,h}(H) := \frac{|\alpha_{g,h}(H)|}{\mathcal{N}(H)}, \qquad \mathcal{F}_{g,h}(\rho) := \mathcal{N}(H) \exp\left(i \operatorname{Arg}(\alpha_{g,h}(H))\right), \tag{4.31}$$

where $\mathcal{N}(H) := \sum_{g,h} |\alpha_{g,h}(H)|$ and we have used the group element pair, $(g,h) \in G \times G$, as the index for the entries of $\mathcal{P}$ and $\mathcal{F}$.

Eq. (4.30) imposes the requirement that $\mathcal{B}$ spans the Hilbert space. We note that for all $H \in \mathcal{H}^n$, $\mathcal{P}(H)$ is a valid probability distribution over $(g,h) \in G \times G$; $\mathcal{F}_{g,h}(H)$ is a scalar and using Eq. (4.30), it is easy to show that Eq. (4.12) is satisfied:

$$\begin{aligned} \underset{X\sim\mathcal{P}(H)}{\mathbb{E}}\left[\mathcal{F}_X(H)\mathcal{B}_X\right] &= \sum_{(g,h)} \mathcal{P}_{g,h}(H)\mathcal{F}_{g,h}(H)\mathcal{B}_{g,h} && (4.32)\\ &= \sum_{(g,h)} \alpha_{g,h}(H)\mathcal{B}_{g,h} && (4.33)\\ &= H. && (4.34)\end{aligned}$$

For concreteness, let us now briefly discuss two special cases.

1. From this example, we can construct the model that corresponds to that used in Ref. [1] based on the Wigner function. To do so, we set:
   - $\mathcal{O} := A_{0,0}$ (this is a phase point operator defined in Eq. (3.18)),
   - The unitary group $\{\, U_g \,\}_{g\in G}$ to be the Weyl-Heisenberg operator defined in Eq. (3.11) i.e. $\mathbf{D}_d^{\otimes n} = \{D_{z,x} = \omega^{-2^{-1}zx} Z^z X^x | x, z \in \mathbb{Z}_d^n\}$
   - $\alpha_{(z,x),z',x'}(H) := \delta_{z,z'}\delta_{x,x'}\frac{1}{d^n}\mathrm{tr}\left(H^\dagger A_{z,x}\right) = \delta_{z,z'}\delta_{x,x'}W_{H^\dagger}(z,x)$
2. From this example, we can construct the model that corresponds to that used in Ref. [34] based on the set of stabilizer states. To do so, let us set:
   - $\mathcal{O} := |0\rangle\langle 0|$,
   - The unitary group $\{\, U_g \,\}_{g\in G}$ to be the group of Clifford operators $\mathbf{C}_{d,n}$ defined in Eq. (3.15). There, we will label the elements of $\mathbf{C}_{d,n}$ by $g \in G$ for notational simplicity.
   - $\alpha_{g,h}(H) := \delta_{g,h}\ f_g(H)$ where the function $f : \mathcal{H}^n \times G \to \mathbb{C}$ only needed to be partially defined in Ref. [34]. Here, for all $g, h \in G$, the map is defined as $f_g(U_h\,|0\rangle\langle 0|\,U_h^\dagger) = \delta_{g,h}$ and the map is also defined on up to 5 copies of the magic state $|T\rangle\langle T|$ (defined in Eq. (5.6)) but there, it was chosen through a numerical convex optimization done to minimize the quantity $\mathcal{N}$.

In Sec. 4.2.1 we discussed a set of free parameters (the model) within the general framework. To elucidate the abstract formalism, we provided some simple, and some more general examples of models. In Sec. 4.2.3 we will discuss how these models are used to produce Born rule probability estimates.

4.2.3 An estimation algorithm

We aim to produce an estimate of the Born rule probability associated with some quantum circuit and event. In Sec. 4.2.1 we presented the mathematical objects that will be used to "simulate" the evolution of the quantum system throughout the string of transformations that are applied to it (e.g. one and two qubit unitaries and a final projective measurement). Sec. 4.2.1 and the examples that followed in Sec. 4.2.2, show how a classical algorithm in the general framework will (often efficiently) represent the quantum state at each stage of its evolution. We have not yet discussed how this representation will be used to produce a Born rule probability estimate.

We now present an algorithm for keeping track of the quantum system through each of the transformations applied in a circuit and discuss techniques for producing a Born rule probability estimate from the updated state representation.

Let us now assume that we have fixed a choice of model. Our algorithm will require three components. First, we require a method for converting the input state[2] into the "natural representation" given by the model. Next, for each transformation that acts on the state throughout the quantum circuit, we require the ability to update our state representation to a new state representation. Finally, we require a method for producing an estimate of the Born probability associated with a measurement performed on the state stored in the natural representation given by the model. We discuss each of these components in turn.

Throughout this section, we will ignore all efficiency constraints. We ask the reader to think of the algorithm presented in this section as a (possibly inefficient) alternative to the standard exponential sized matrix representation. In the next section, we will return to each of the steps presented here and discuss any additional constraints that will be required to ensure the efficiency of the algorithm.

Translation of input state to the model's state representation

We now present the first component of our general framework. This component specifies how the input quantum state is to be *translated* into the representation that is used by the algorithm.

Let $\sigma \in \mathcal{H}^n$ be the input quantum state to the quantum circuit. This may be a state represented in either the pure or mixed state formalism. We will represent the state σ by the pair $(\mathcal{F}^{(0)}, \mathcal{B}_x)$. Where $\mathcal{F}^{(0)}$ is a number defined by $\mathcal{F}^{(0)} := \mathcal{F}_x(\sigma)$ and $\mathcal{B}_x$ is an element of the frame $\mathcal{B}$. The value of x will be randomly sampled from the probability distribution $\mathcal{P}(\sigma)$. We represent this random variable by $X^{(0)}$.

Thus the translation of state representation involves three steps:

1. For $\sigma \in \mathcal{H}^n$, sample $X^{(0)}$ from the distribution $\mathcal{P}(\sigma)$
2. Compute and store $\mathcal{F}^{(0)} := \mathcal{F}_{X^{(0)}}(\sigma)$
3. Store $X^{(0)}$ (this will serve as the index for $\mathcal{B}_{X^{(0)}}$)

Updating the state under transformations

We now present the second component of algorithms in the general framework; this component specifies how the input quantum state is to be *transformed* from one state representation to the next as the quantum state is updated through each elementary step of the quantum circuit.

[2] Represented in the standard form, i.e. as a tensor product of a quantum states for a bounded number of qudits with each of these smaller parts given explicitly as a vector/matrix.

Let the map $T_t : \mathcal{H}^n \to \mathcal{H}^n$ represent the t^{th} state transformation in either the pure or mixed state formalism. We will be representing the state of the system prior to the application of T_{t+1} by the double $(\mathcal{F}^{(t)}, \mathcal{B}_{X^{(t)}})$. The application of T_{t+1} will update our representation to $(\mathcal{F}^{(t+1)}, \mathcal{B}_{X^{(t+1)}})$ where $X^{(t+1)}$ is sampled from the probability distribution $\mathcal{P}(T_{t+1}(\mathcal{B}_{X^{(t)}}))$ and $\mathcal{F}^{(t+1)} := \mathcal{F}^{(t)} \times \mathcal{F}_{X^{(t+1)}}(T_{t+1}(\mathcal{B}_{X^{(t)}}))$.

Thus the state update involves three steps:

1. Sample $X^{(t+1)}$ from the probability distribution $\mathcal{P}(T_{t+1}(\mathcal{B}_{X^{(t)}}))$
2. Compute entry $X^{(t+1)}$ of $\mathcal{F}(T_{t+1}(\mathcal{B}_{X^{(t)}}))$, multiply the result by $\mathcal{F}^{(t)}$ and store the result as $\mathcal{F}^{(t+1)}$
3. Store $X^{(t+1)}$ (this will serve as the index for $\mathcal{B}_{X^{(t+1)}}$)

Producing Born rule probability estimates

We now present the third and final component of algorithms in the general framework. This component specifies how the state representation will be used to produce a Born rule probability for a given measurement outcome. In the pure and mixed state formalisms, this component of the algorithm is distinct. In the mixed state formalism, the mapping between the quantum state (density operators) and the Born rule probability is linear. In the pure state formalism, the mapping between the quantum state (ket vectors) and amplitudes is linear but the mapping to Born rule probabilities is not. This distinction underlies the need to treat this component of our algorithm differently in the two formalisms.

We now present the third component of algorithms in the general framework first as it applies to the mixed state formalism (as this makes use of more familiar elements) then as it applies to the pure state formalism.

Producing Born rule probability estimates in the mixed state formalism

Consider a projective measurement with the outcome of interest corresponding to the projector Π. We will be representing the state of the system prior to measurement by the double $(\mathcal{F}^{(L)}, \mathcal{B}_{X^{(L)}})$. The application of Π will update our representation to $(\mathcal{F}^{(L+1)}, \mathcal{B}_{X^{(L+1)}})$.

In the first two components of the algorithm, we defined each of the distributions in the string of probability distributions $\big(\mathcal{P}(\sigma), \mathcal{P}(T_1(\mathcal{B}_{x_0})), \ldots, \mathcal{P}(T_L(\mathcal{B}_{x_{L-1}})), \mathcal{P}(\Pi\mathcal{B}_{x_L})\big)$. These described the distributions and conditional distributions we sampled from in order to choose the next scalar and frame element to use in the updated state representation. At this stage, it may be helpful to think of the string of sampled points as a "path" through some "discretized phase space" $[M]$. In this picture, the process of sampling the string of points $(x_0, \ldots, x_{L+1})$ is a Markov process for sampling from the joint distribution over all length $L+1$ paths in $[M]$.

For $j = 0, 1, \ldots, L+1$, it is useful to also define $\mathcal{Q}^{(j)}$, the joint probability distribution of

$X := \left(X^{(0)}, X^{(1)}, \ldots, X^{(j)}\right)$ i.e. the distribution of paths of length j.

$$\mathcal{Q}^{(j)}_{(x_0,\ldots,x_j)} = \begin{cases} \mathcal{P}_{x_0}(\sigma); & \text{for } j = 0 \\ \mathcal{P}_{x_0}(\sigma) \times \prod_{i=1}^{j} \mathcal{P}_{x_i}(T_i(\mathcal{B}_{x_{i-1}})); & \text{for } j = 1, \ldots, L \\ \mathcal{P}_{x_0}(\sigma) \times \prod_{i=1}^{L} \mathcal{P}_{x_i}(T_i(\mathcal{B}_{x_{i-1}})) \times \mathcal{P}_{x_{L+1}}(\Pi \mathcal{B}_{x_L}); & \text{for } j = L+1 \end{cases} .$$

We now note that the expectation value, over randomly sampled paths, of our state representation at the end of the circuit $\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}$ is given by:

$$\begin{aligned} \mathop{\mathbb{E}}_{X\sim\mathcal{Q}^{(L+1)}}\left[\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}\right] &= \sum_{x_1,\ldots,x_{L+1}} \mathcal{Q}^{(L+1)}\mathcal{F}^{(L+1)}\mathcal{B}_{x_{L+1}} \\ &= \sum_{x_1,\ldots,x_{L+1}} \mathcal{Q}^{(L)}\mathcal{F}^{(L)}\mathcal{P}_{x_{L+1}}(\Pi\mathcal{B}_{x_{L+1}})\mathcal{F}_{x_{L+1}}(\Pi\mathcal{B}_{x_{L+1}})\mathcal{B}_{x_{L+1}} && (4.35) \\ &= \sum_{x_1,\ldots,x_L} \mathcal{Q}^{(L)}\mathcal{F}^{(L)}\Pi\mathcal{B}_{x_L} && (4.36) \\ &\vdots \\ &= \Pi \times T_L \circ \ldots \circ T_1(\sigma) && (4.37) \end{aligned}$$

where from Eqs. (4.35) to (4.36), we have summed over x_{L+1} using Eq. (4.12) and from Eqs. (4.36) to (4.37) we use the linearity of the the action of the operators Π and T_i to move these outside the sum.

Recall that the RHS, $\Pi \times T_L \circ \ldots \circ T_1(\sigma)$ is the initial state σ evolved under each of the transformation applied within the circuit including a final projection relating to the measurement. Hence, we have shown that the algorithm's final state representation $\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}$ will, on expectation equal the circuit's output quantum state.

We now note that in the mixed state case, for a given state ρ and projector Π, the Born probability can be written $\operatorname{tr}(\Pi\rho) = \operatorname{tr}(\rho\Pi) = \operatorname{tr}(\Pi\rho\Pi)$. Here, we are free to choose a representation where our state is multiplied by Π from the left, or the right, or both or a combination of these by splitting Π into a tensor product. For simplicity, we have presented the above based on the "multiplication from the left approach".

We now output the Born rule probability estimate:

$$\hat{p}_1 := \mathcal{F}^{(L+1)} \times \operatorname{tr}\left(\mathcal{B}_{X^{(L+1)}}\right). \tag{4.38}$$

By the linearity of the trace we have:

$$\begin{aligned} \mathop{\mathbb{E}}_{X\sim\mathcal{Q}^{(L+1)}}[\hat{p}_1] &= \mathop{\mathbb{E}}_{X\sim\mathcal{Q}^{(L+1)}}\left[\mathcal{F}^{(L+1)}\operatorname{tr}\left(\mathcal{B}_{X^{(L+1)}}\right)\right] && (4.39) \\ &= \operatorname{tr}\left(\Pi \times T_L \circ \ldots \circ T_1(\sigma)\right) && (4.40) \\ &= p. && (4.41) \end{aligned}$$

Given this unbiased estimator, we can independently compute s estimates and average these arriving at the average $\hat{p}_s$. By the Hoeffding inequality:

$$\Pr\Big(|p - \hat{p}_s| \geq \epsilon\Big) \leq 2\exp\Big(\frac{-2s\epsilon^2}{\mathcal{R}^2}\Big)\,, \tag{4.42}$$

where $\mathcal{R}$ is defined as the maximum range of $\hat{p}_1$ over all possible sample outcomes X.

We now present the third component of the algorithm in the pure state formalism.

Producing Born rule probability estimates in the pure state formalism

In the pure state case, the update of representation due to measurement is the same as it was in the mixed state formalism. To update the state representation based on the action of the measurement projector, Π; $X^{(L+1)}$ is sampled from the probability distribution $\mathcal{P}(\Pi\mathcal{B}_{X^{(L)}})$. Using $\mathcal{F}^{(L+1)} := \mathcal{F}^{(L)} \times \mathcal{F}_{X^{(L+1)}}(\Pi\mathcal{B}_{X^{(L)}})$, the final state post measurement is represented by the pair $(\mathcal{F}^{(L+1)}, \mathcal{B}_{X^{(L+1)}})$.

In the pure state formalism, it is still helpful to have in mind the "path through phase space" picture we discussed earlier in relation to the mixed state formalism. For $j = 0, 1, \ldots, L+1$, we will still define a joint distribution of paths of length j using $\mathcal{Q}^{(j)}$ as we did in the mixed state formalism. However, in the pure state formalism, the projector Π must always act by multiplication from the left. This ensures that as per the mixed state case:

$$\mathop{\mathbb{E}}_{X\sim\mathcal{Q}^{(L+1)}}\Big[\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}\Big] = \Pi \times T_L \circ \ldots \circ T_1(\sigma). \tag{4.43}$$

The squared l_2 norm of this expectation vector is the desired Born rule probability. We will produce this estimate in two steps. First, we will independently sample from $\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}$ a total of s times and compute (or estimate) the sample average. This vector will be used as an estimate of the expectation vector. We will then compute the squared l_2 norm of the sample average vector (or approximation to the sample average vector) as an estimate for the target Born probability. The accuracy of such an estimate can be bounded by a vector version of the Hoeffding inequality.

Lemma 1. *Let $\chi, s \in \mathbb{N}$ and $m > 0$. Let $\mathcal{FB} := \{\ |\ \phi_j\rangle\ \}_{j\in[\chi]}$ be a set of D-dimensional vectors over $\mathbb{C}$ with lengths $\||\phi_j\rangle\|_2 \leq m$. Let $\mathcal{P} := \{\,\mathcal{P}_j\,\}_{j\in[\chi]}$ be a probability distribution over $[\chi]$ and define $|\mu\rangle$ as the D-dimensional vector over $\mathbb{C}$ that is the expectation of $|\phi_X\rangle$ with respect to the random variable X with probability distribution $\mathcal{P}$:*

$$|\mu\rangle = \mathop{\mathbb{E}}_{X\sim\mathcal{P}}[|\phi_X\rangle] = \sum_{j\in[\chi]} \mathcal{P}_j\,|\phi_j\rangle\,. \tag{4.44}$$

For $j \in [s]$, let $x_j \in [\chi]$ be independently sampled from the probability distribution $\mathcal{P}$. We define a vector sample mean over s samples by:

$$\overline{|\phi\rangle} = \frac{1}{s}\sum_{j=1}^{s} |\phi_{x_j}\rangle. \tag{4.45}$$

Then for all $\epsilon > 0$:

$$\Pr\left(\left\|\overline{|\phi\rangle} - |\mu\rangle\right\|_2 \geq \epsilon\right) \leq 2e^2 \exp\left(\frac{-s\epsilon^2}{2(m+p)^2}\right) \tag{4.46}$$

where $p := \||\mu\rangle\|_2$.

The proof of Lem. 1 is presented in Sec. 4.2.4.

Let us denote the s independent samples of $\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}$ by $|\phi_{x_1}\rangle, \ldots, |\phi_{x_s}\rangle$ and the average of these by $\overline{|\phi\rangle}$. If the squared l_2 norm of the sample average, $\left\|\overline{|\phi\rangle}\right\|_2^2$, can be exactly computed, then this quantity serves as our estimator $\hat{p}_s$ for the Born rule probability p. In particular, we will now show that for all $\epsilon > 0$:

$$\Pr\left(|\hat{p}_s - p| \geq \epsilon\right) \leq 2e^2 \exp\left(\frac{-s\left(\sqrt{p+\epsilon} - \sqrt{p}\right)^2}{2(\mathcal{R}+p)^2}\right), \tag{4.47}$$

where $\mathcal{R}$ is the upper bound on $\left\|\mathcal{F}^{(L+1)}\mathcal{B}_{X^{(L+1)}}\right\|_2$.

By application of Lem. 1, we have:

$$\Pr\left(\left\|\overline{|\phi\rangle} - |\mu\rangle\right\|_2 \geq \epsilon\right) \leq 2e^2 \exp\left(\frac{-s\epsilon^2}{2(\mathcal{R}+p)^2}\right), \tag{4.48}$$

where, $|\mu\rangle$ is the expectation of $|\phi\rangle$ given by Eq. (4.43) and has a squared l_2 norm of p. We now note that by the reverse triangle inequality:

$$\begin{aligned}\left\|\overline{|\phi\rangle} - |\mu\rangle\right\|_2 &\geq \left|\left\|\overline{|\phi\rangle}\right\|_2 - \||\mu\rangle\|_2\right| \\ &= \left|\sqrt{\hat{p}_s} - \sqrt{p}\right|.\end{aligned} \tag{4.49}$$

This gives:

$$\Pr\left(\left|\sqrt{\hat{p}_s} - \sqrt{p}\right| \geq \epsilon\right) \leq 2e^2 \exp\left(\frac{-s\epsilon^2}{2(\mathcal{R}+p)^2}\right). \tag{4.50}$$

We now note that if $\left|\sqrt{\hat{p}_s} - \sqrt{p}\right| \leq \epsilon$ then $|\hat{p}_s - p| \leq \epsilon(\epsilon + 2\sqrt{p})$. This can be seen by applying the triangle inequality as follows:

$$\begin{aligned}|\hat{p}_s - p| &= \left|\sqrt{\hat{p}_s} + \sqrt{p}\right|\left|\sqrt{\hat{p}_s} - \sqrt{p}\right| \\ &= \left|\sqrt{\hat{p}_s} - \sqrt{p} + 2\sqrt{p}\right|\left|\sqrt{\hat{p}_s} - \sqrt{p}\right| \\ &\leq \left(\left|\sqrt{\hat{p}_s} - \sqrt{p}\right| + 2\sqrt{p}\right)\left|\sqrt{\hat{p}_s} - \sqrt{p}\right| \\ &\leq (\epsilon + 2\sqrt{p})\epsilon.\end{aligned} \tag{4.51}$$

This gives:

$$\Pr\left(|\hat{p}_s - p| \geq \epsilon(\epsilon + 2\sqrt{p})\right) \leq 2e^2 \exp\left(\frac{-s\epsilon^2}{2(\mathcal{R}+p)^2}\right). \tag{4.52}$$

We can now define a new variable $\varepsilon = \epsilon(\epsilon + 2\sqrt{p})$ and solve this quadratic equation for ϵ. Taking only the positive solution gives $\epsilon = \sqrt{p+\varepsilon} - \sqrt{p}$. Substituting into Eq. (4.52) gives:

$$\Pr\left(|\hat{p}_s - p| \geq \varepsilon\right) \leq 2e^2 \exp\left(\frac{-s\left(\sqrt{p+\varepsilon} - \sqrt{p}\right)^2}{2(\mathcal{R}+p)^2}\right), \tag{4.53}$$

which is equivalent to Eq. (4.47).

We note that the RHS of Eq. (4.47) can be upper bounded by setting $p = 1$. This gives a p independent bound on the accuracy of the estimator however, we note that performance improves when the Born probability p is smaller.

In Ch. 5, we will present an algorithm that uses the approach outlined here but in the slightly more complicated setting where the l_2 norm of the sampled average vector, $\left\|\overline{|\phi\rangle}\right\|_2$, is itself estimated.

4.2.4 Proof of the vector version of the Hoeffding inequality

In this section we prove Lem. 1. This will use a theorem from Ref. [53] and the definition of a *very-weak martingale* given below.

Definition 2. *(very-weak martingale) Let $N \in \mathbb{N}$, Ω be a sample space and for all $j \in \mathbb{N}$, let $X_j : \Omega \to \mathbb{R}^N$ be a random variable taking values in $\mathbb{R}^N$ such that $X_0 = 0$, $\mathbb{E}\left[\|X_j\|_2\right] < \infty$ and $\mathbb{E}\left[X_j \mid X_{j-1}\right] = X_{j-1}$. Then we call the sequence $(X_0, X_1, \ldots)$ a* very-weak martingale in $\mathbb{R}^N$.

Theorem 2. *(Hayes: Theorem 1.8) Let X ba a very-weak martingale taking values in $\mathbb{R}^N$ such that $X_0 = 0$ and for every j, $\|X_j - X_{j-1}\|_2 \leq 1$. Then for every $a > 0$:*

$$\Pr\left(\|X_s\|_2 \geq a\right) \leq 2e^{1-(a-1)^2/2s} < 2e^2 \exp\left(-a^2/2s\right) \tag{4.54}$$

We now prove Lem. 1

Proof. The proof is a simple application of Thm. 2. Let us use $R : \mathbb{C}^D \to \mathbb{R}^{2D}$ to denote the two-norm preserving linear map $R(a_1 + ib_1, \ldots, a_D + ib_D) = (a_1, b_1, \ldots, a_D, b_D)$. For $s \in \mathbb{N}$, we define the random variable $Y_s \in \mathbb{R}^{2D}$ as follows. $Y_0 = (0, \ldots, 0)$ and for $s > 0$:

$$Y_s := \frac{s}{m+p} R\left(\overline{|\phi\rangle} - |\mu\rangle\right),$$

where we note that $\overline{|\phi\rangle}$ depends on s as per Eq. (5.19).

We now note that Y_s is a very-weak martingale since $Y_0 = 0$, and:

$$\begin{aligned}\mathbb{E}\left[\|Y_s\|_2\right] &= \frac{s}{m+p}\mathbb{E}\left[\sqrt{\left(\overline{\langle\phi|} - \langle\mu|\right)\left(\overline{|\phi\rangle} - |\mu\rangle\right)}\right] \\ &< \infty\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\left[Y_s \mid Y_{s-1}\right] &= \mathbb{E}\left[Y_{s-1} + \frac{1}{m+p}R\left(|\phi_{x_s}\rangle - |\mu\rangle\right) \mid Y_{s-1}\right] \\ &= Y_{s-1}.\end{aligned}$$

Additionally, we note that $\|Y_s - Y_{s-1}\|_2 \leq 1$ since:

$$\begin{aligned}\|Y_s - Y_{s-1}\|_2 &= \left\|\frac{1}{m+p}R\left(|\phi_{x_s}\rangle - |\mu\rangle\right)\right\|_2 \\ &= \frac{1}{m+p}\sqrt{\langle\phi_{x_s}|\phi_{x_s}\rangle - \langle\phi_{x_s}|\mu\rangle - \langle\mu|\phi_{x_s}\rangle + \langle\mu|\mu\rangle} \\ &\leq \frac{1}{m+p}\sqrt{m^2 + 2mp + p^2} \\ &= 1.\end{aligned}$$

Hence, by Thm. 2:

$$\begin{aligned}\Pr\left(\|Y_s\|_2 \geq a\right) &= \Pr\left(\left\|\overline{|\phi\rangle} - |\mu\rangle\right\|_2 \geq \frac{a(m+p)}{s}\right) \\ &< 2e^2 \exp\left(-a^2/2s\right).\end{aligned}$$

Substituting $\epsilon = \frac{a(m+p)}{s}$ proves the claim.

□

4.2.5 Efficiency constraints

Our exposition of the general framework has to this point introduced the main mathematical structure and estimation procedure while avoiding discussion relating to efficiency. The choice of the triple $(\mathcal{B}, \mathcal{P}, \mathcal{F})$ must satisfy a number of additional constraints to render the estimation algorithm efficiently executable. In particular, we impose the following requirements.

1. Recall that each of the three components of our framework requires sampling from probability distributions in order to update the state representation. Hence, we require that the probability distributions $\mathcal{P}(\sigma)$, $\mathcal{P}(T_i(\mathcal{B}_j))$ and $\mathcal{P}(\Pi\mathcal{B}_j)$ can be efficiently sampled for all $i \in [L]$ and $j \in [M]$.

2. Recall that each of the three components of our framework requires computing certain "$\mathcal{F}$-factors" in order to update the state representation. Hence, we require that the scalars $\mathcal{F}_k(\sigma)$, $\mathcal{F}_k(T_i(\mathcal{B}_j))$ and $\mathcal{F}_k(\Pi\mathcal{B}_j)$ can be efficiently computed for all $i \in [L]$ and $j, k \in [M]$.

3. Recall that the quantity $\mathcal{R}$ quantifies the range of values that our Born rule probability estimate, $\hat{p}$ can take. We will require that $\mathcal{R}$ is upper bounded by a polynomial in n.

4. If applying the pure state formalism using the vector Hoeffding method, we will also require that one can efficiently (in n and s) estimate the l_2 norm of the sample average vector.

We discuss each of these points in turn. First let us note that sampling from a probability distribution is not a computationally trivial task. In particular, the probability distributions we are considering are over a space of M elements where M will typically grow like d^n (or significantly faster for over-complete frames). In general, there do not exist efficient protocols for sampling from probability distribution families over exponentially growing spaces. However, there exists a large toolbox of techniques that can be used to efficiently sample from such distributions. The simplest examples occur when the random variable we wish to sample from has a product distribution or can be invertibly transformed into a product distribution. Additionally there exist a plethora of techniques such as rejection sampling, metropolis sampling and other Monte Carlo sampling methods. Sampling from another distribution that is a good approximation to the target distribution may be possible provided that the effect of this approximation on the final Born probability estimate is acceptable.

Secondly, we note that computation of the $\mathcal{F}$-factors is non-trivial. Approximation to these may also be used provided that the effect of this approximation on the final Born probability estimate is acceptable.

To address the third point, we note that the simulation run-time will quadratically depend on the quantity $\mathcal{R}$. Recall that:

$$\mathcal{R} := \begin{cases} \max\limits_{X \in \text{Supp } \mathcal{Q}^{L+1}} \left\{ \left\| \mathcal{F}^{(L+1)} \mathcal{B}_{X^{(L+1)}} \right\|_2 \right\}; & \text{in the pure state formalism} \\ \max\limits_{X \in \text{Supp } \mathcal{Q}^{L+1}} \left\{ \left| \mathcal{F}^{(L+1)} \times \operatorname{tr}\left(\mathcal{B}_{X^{(L+1)}}\right) \right| \right\}; & \text{in the mixed state formalism,} \end{cases} \tag{4.55}$$

quantifies the range of values that the Born rule probability estimates $\hat{p}_1$ can take. Based on Eqs. (4.42) and (4.47), the number of samples s that are required to achieve an (ϵ, δ)-additive precision estimate for a particular (ϵ, δ) scales proportionally to $\mathcal{R}^2$. It is important to note that the magnitude of $\mathcal{R}$ is driven by the magnitude of $\mathcal{F}^{(L+1)}$. This quantity in turn is determined by multiplying $\mathcal{F}$-factors from the initial state and through each step of the evolution. Finally, we point out that as seen in all examples in Sec. 4.2.2, the $\mathcal{F}$-factors are driven by the l_1 norms $\mathcal{N}$ in the linear decomposition of states into the frame elements.

We note that it is possible for the estimator to achieve an (ϵ, δ)-additive precision despite the number of samples being "insufficiently large" based on Eqs. (4.42) and (4.47). This is because these performance bounds, based on the Hoeffding inequality may not be tight.

Finally, the application of the pure state formalism produces a linear combination of sampled vectors such that its l_2 norm is the desired estimate. Computing or estimating this norm is a key step in producing the final Born rule probability estimate. The inaccuracy introduced through the use of approximations of this l_2 norm will impact the estimator accuracy.

The above discussion focuses on the efficiency of each step in the estimation protocol. However, there are a number of interesting application where the state of the art Born rule probability estimation protocols have exponential run-time in n. In this setting, often the goal is to improve the scaling of run-time despite the inefficiency of the algorithm. The focus of Chapter 5 is the exposition of such an algorithm.

4.2.6 Summary of the general framework

In Sec. 4.2, we defined a general framework for constructing algorithms that produce Born rule probability estimates. We discussed a key ingredient known as the model by first providing a mathematical definition then giving a number of examples. We then presented an algorithm for Born rule probability estimation given a fixed choice of model. Finally, we discussed the requirements that must be satisfied to produce an efficient estimation algorithm.

4.3 Related works within the general framework

In Sec. 4.2, we presented a general framework for the construction of Born rule probability estimation algorithms. In this section we present selected recent works on classical simulation algorithms [1, 40, 41, 2, 38, 34, 37]. We do not intend to give a detailed review of these works or aim to compare their relative performance/merits. Instead, our exposition will focus on certain features of each algorithm with the primary aim of motivating and elucidating the abstract mathematical framework we presented in Sec. 4.2.

4.3.1 Dual frame mixed state formalism

Ref. [1] presented a general family of additive $1/poly$ precision estimation algorithms based on a slightly more constrained model than we are considering[3]. Ref. [1] restricts its focus to the mixed state formalism where, similarly to the present work, they provide an “incomplete recipe” for the construction of an estimation algorithm.

Ref. [1] can be used to construct simulation algorithms for n-particle, d-level quantum systems with a state space given by the Hilbert space $\mathcal{H}^n$ of linear operators from $\left(\mathbb{C}^d\right)^n$ to itself. The construction of a simulation algorithm requires a choice of a *dual frame*. Here, two frames $F := \{ F_\lambda \}_{\lambda\in\Lambda}$ and $G := \{ G_\lambda \}_{\lambda\in\Lambda}$ are defined where each frame is a set of Hermitian operators in $\mathcal{H}^n$ such that:

1. Each frame is spanning:

$$\operatorname*{span}_{\mathbb{C}} F = \operatorname*{span}_{\mathbb{C}} G = \mathcal{H}^n. \tag{4.56}$$

2. The F frame satisfies the following normalization condition:

$$\sum_{\lambda\in\Lambda} F_\lambda = I. \tag{4.57}$$

[3] However, they consider positive operator valued measurements which are more general than the projective measurements considered here.

3. The G frame satisfies the following normalization condition:

$$\operatorname{tr}\left(G_\lambda\right) = 1 \tag{4.58}$$

for all $\lambda \in \Lambda$.

4. the pair F, G satisfy a duality condition. This requires that for all $H \in \mathcal{H}^n$:

$$H = \sum_{\lambda \in \Lambda} \operatorname{tr}\left(H F_\lambda^\dagger\right) G_\lambda. \tag{4.59}$$

The dual frame in this construction is a free choice analogous to the free choice of models. However, given the choice of dual frame, one has a complete recipe for constructing a *quasi-probabilistic representation* and an estimation algorithm. Specifically, a quasi-probabilistic representation is defined by the mapping of the quantum states, transformations and POVM elements as follows:

$$(\rho, \lambda) \mapsto W_\rho(\lambda) := \operatorname{tr}\left(\rho F_\lambda^\dagger\right) \tag{4.60}$$

$$(U, \lambda, \lambda') \mapsto W_U(\lambda' \mid \lambda) := \operatorname{tr}\left(U G_\lambda U^\dagger F_{\lambda'}^\dagger\right) \tag{4.61}$$

$$(E, \lambda') \mapsto W(E \mid \lambda') := \operatorname{tr}\left(E G_{\lambda'}\right). \tag{4.62}$$

The notation is in analogy to probabilities and conditional probabilities because like these, $W_\rho(\lambda), W_U(\lambda' \mid \lambda)$ and $W(E \mid \lambda')$ are real values and normalized like (conditional) probability distributions i.e.

$$\sum_{\lambda \in \Lambda} W_\rho(\lambda) = 1. \tag{4.63}$$

For all $\lambda \in \Lambda$:

$$\sum_{\lambda' \in \Lambda} W_U(\lambda' \mid \lambda) = 1. \tag{4.64}$$

Finally, for any positive, Hermitian set of operators $\{ E_1, \ldots, E_m \}$ that sums to the identity and hence define a positive operator valued measurement, we have:

$$\sum_j W(E_j \mid \lambda') = 1. \tag{4.65}$$

These are called quasi-probabilistic distributions because they can be negative. Despite this, the Born rule probability can be written in terms of these quasi-probabilities in an identical way to a probability associated with a Markov chain process. Specifically, letting $U := \prod_{j \in [L]} U_j$, then the Born rule probability associated with starting in the state ρ, acting on it with U then observing the first outcome upon measuring $\{ E, I - E \}$ is given by $p := \operatorname{tr}\left(U \rho U^\dagger E\right)$. In the quasi-probabilistic language, this can be written as:

$$p = \sum_{\lambda^{(0)},\ldots,\lambda^{(L)} \in \Lambda} W_\rho(\lambda^{(0)}) \times \prod_{j=1}^{L} W_{U_j}(\lambda^{(j)} \mid \lambda^{(j-1)}) \times W(E \mid \lambda^{(L)}). \tag{4.66}$$

This allows the construction of an estimator $\hat{p}_1$ as follows. We first sampling the string $(\lambda^{(0)}, \ldots, \lambda^{(L)})$ from the distributions:

$$\Pr(\lambda^{(0)}) = \frac{\left|W_\rho(\lambda^{(0)})\right|}{\mathcal{N}_\rho} \qquad \Pr(\lambda^{(j)} \mid \lambda^{(j-1)}) = \frac{\left|W_{U_j}(\lambda^{(j)} \mid \lambda^{(j-1)})\right|}{\mathcal{N}_{U_j}(\lambda^{(j-1)})} \tag{4.67}$$

where $\mathcal{N}_\rho := \sum_\lambda |W_\rho(\lambda)|$ and $\mathcal{N}_{U_j}(\lambda^{(j-1)}) := \sum_\lambda \left|W_{U_j}(\lambda \mid \lambda^{(j-1)})\right|$. We then compute the estimate:

$$\hat{p}_1(\lambda^{(0)}, \ldots, \lambda^{(L)}) := \mathrm{sgn}(W_\rho(\lambda^{(0)}))\mathcal{N}_\rho \times \left[\prod_{j=1}^{L} \mathrm{sgn}(W_{U_j}(\lambda^{(j)} \mid \lambda^{(j-1)}))\mathcal{N}_{U_j}(\lambda^{(j-1)})\right] \times W(E \mid \lambda^{(L)}).$$

Using Eq. (4.66), it is easy to show that $\hat{p}_1$ is an unbiased estimator of p and hence, by the application of the Hoeffding inequality produces (ϵ, δ)-additive precision estimates of Born probabilities with run-time that scales like $poly(n, 1/\epsilon, \log\ 1/\delta, \mathcal{N})$ where $\mathcal{N} \leq \max_{\lambda^{(0)},\ldots,\lambda^{(L)}} \left|\hat{p}_1(\lambda^{(0)}, \ldots, \lambda^{(L)})\right|$. The quantity $\mathcal{N}$ is analogous to the quantity $\mathcal{R}$ discussed in Sec. 4.2.3. In Ref. [1], $\mathcal{N}$ is referred to as the *negativity* of the circuit because the imposed normalization conditions ensure that $\mathcal{N}$ is a natural measure of the total "amount of negative quasi-probabilities" in the quasi-probabilistic representation of the quantum circuit elements.

Ref. [1] also makes the useful observation that a number of symmetries of the Born rule probability are not shared by the estimator. Thus, the application of symmetry transformations can result in the construction of better performing estimators for the same Born rule probability. One particularly useful symmetry is the circuit reversal symmetry and can be applied when the evolution is unitary. Under this symmetry on can apply the following mapping to the input state ρ, each of the unitaries U_i and the final POVM effect E:

$$\tilde{\rho} \leftarrow \frac{1}{\mathrm{tr}\,(E)} E \qquad \tilde{U}_i \leftarrow U_i^\dagger \qquad \tilde{E} \leftarrow \mathrm{tr}\,(E)\rho, \tag{4.68}$$

where the tilded objects are the post-transformation circuit inputs. It is straightforward to verify that the Born rule probability is unchanged under this transformation i.e.

$$\mathrm{tr}\left(U\rho U^\dagger E\right) = \mathrm{tr}\left(\tilde{U}\rho\tilde{U}^\dagger\tilde{E}\right). \tag{4.69}$$

The key differences between the model we present in Sec. 4.2.3 and that of Ref. [1] are that:

1. The present model unifies the pure and mixed state formalism.

2. The normalization conditions imposed in Ref. [1] can be weakened to include additional estimation algorithms with practical applications. We will discuss this in more detail in

Sec. 4.3.5.

3. The present model does not make explicit reference to a choice of dual frame. In the model presented in Ref. [1], the frame G acts as the state representative in analogy to the $\mathcal{B}$ frame of the present model. The frame F serves to decide the coefficients used in the expansion of the state in terms of the G frame. In the special case where $|\Lambda| = M = dim(\mathcal{H}^n)$, the frame F is uniquely determined by the choice of G. Thus, in this case, all else being equal, setting $\mathcal{B} = G$ results in very similar estimation algorithms. However, in a more general setting, for a D dimensional space an over-complete spanning set of vectors $\{ v_1, \ldots, v_m \}$ has an $m - D$ dimensional degree of freedom in the choice of dual frame. Thus, when the frames $\mathcal{B} = G$ are over-complete, this degree of freedom need not be fixed by the algorithm and may in fact vary from one state representation to another within a single estimation run. In Sec. 4.3.4 we will discuss an example where this degree of freedom is used to optimize for algorithm run-time. In principal, one may view this as an optimization over the choice of F frame however, in this setting it is unclear how the mathematical structure of F is at all useful or natural for the optimization task.

4. In the present model, we do not impose the condition of Hermiticity on the frame $\mathcal{B}$. Ref. [54] shows that removal of such a requirement can result in substantially improved run-time for a fixed estimation accuracy.

4.3.2 Path integral pure state formalism

A path integral view of quantum circuits was used in Ref. [40] to study the complexity of computing quantum amplitudes. We present the path integral view in the general framework but look at it in the context of Born rule probability estimation. We use this example to illustrate how the run-time performance of the algorithms constructed using the general framework can change as we change the choice of model parameters such as the frame.

Let us revisit Example 2 and consider the simulation of a circuit where the quantum system starts in the state $|\phi\rangle$, evolves under a string of unitaries $U = U_L \times \ldots \times U_1$ and is finally measured in the computational basis. Let us consider the step by step evolution of the initial state written in the computational basis frame of Example 2.

$$\begin{aligned}
|\phi\rangle &= \sum_{x^{(0)}} \langle x^{(0)}|\phi\rangle \, |x^{(0)}\rangle \\
U_1 \, |\phi\rangle &= \sum_{x^{(0)}} \langle x^{(0)}|\phi\rangle U_1 \, |x^{(0)}\rangle \\
&= \sum_{x^{(0)},x^{(1)}} \langle x^{(1)}| \, U_1 \, |x^{(0)}\rangle \, \langle x^{(0)}|\phi\rangle \, |x^{(1)}\rangle \\
&\vdots \\
U_L \times \ldots \times U_1 \, |\phi\rangle &= \sum_{x^{(0)},\ldots x^{(L)}} \langle x^{(L)}| \, U_L \, |x^{(L-1)}\rangle \times \ldots \times \langle x^{(1)}| \, U_1 \, |x^{(0)}\rangle \, \langle x^{(0)}|\phi\rangle \, |x^{(L)}\rangle
\end{aligned}$$

Finally, we note that the amplitude associated with outcome $y \in \mathbb{Z}_d^n$ can be written as the sum:

$$\langle y| U |\phi\rangle = \sum_{x^{(0)},\ldots x^{(L)}} \langle y|x^{(L)}\rangle \langle x^{(L)}| U_L |x^{(L-1)}\rangle \times \ldots \times \langle x^{(1)}| U_1 |x^{(0)}\rangle \langle x^{(0)}|\phi\rangle. \tag{4.70}$$

Thus, we note that the discretized path integral approach to amplitude estimation is naturally related to the present model. Ref. [40] considers the application of the path integral formalism for the universal gate-set generated by Toffoli and Hadamard gates acting on qubits (see also the related work of Montanaro [41]). For this choice, all gate amplitudes associated with the Toffoli gate are deterministic. Specifically, for $|a\rangle, |b\rangle \in \{ \ | x\rangle \ | \ x \in \mathbb{Z}_2^n \} = \mathcal{B}$ and $T_{i,j,k}$ representing the Toffoli gate with target qubit k controlled on qubits i and j, the transition amplitudes appearing in Eq. (4.70) are given by:

$$\langle b| T_{i,j,k} |a\rangle = \delta_{\bar{b}^k,\bar{a}^k}\delta_{b_k,a_k\oplus a_i a_j} \tag{4.71}$$

where $\bar{v}^k$ represents the vector v with the the k^{th} entry removed and "$\oplus$" represents addition modulo 2. This results in a simplification of the sum corresponding to deterministic (and efficiently computable) jumps in the Markov chain at each step corresponding to a Toffoli gate. In this model, up to a normalization factor, the Hadamard gates act only on the phase of each term as follows:

$$\langle b| H_k |a\rangle = \frac{1}{\sqrt{2}}\delta_{\bar{b}^k,\bar{a}^k}(-1)^{a_k b_k}. \tag{4.72}$$

For the choice of model we are presently considering, the non-deterministic jumps associated with Hadamard gates can be sampled with probability $\mathcal{P}_b(|a\rangle)$ and corresponding $\mathcal{F}$-factor $\mathcal{F}_b(|a\rangle)$ given by:

$$\mathcal{P}_b(|a\rangle) := \frac{|\langle b| H_k |a\rangle|}{\mathcal{N}} \qquad \mathcal{F}_b(|a\rangle) := \mathcal{N}\exp\left(i\,\mathrm{Arg}(\langle b| H_k |a\rangle)\right) \tag{4.73}$$

where $\mathcal{N} := \sum_b |\langle b| H_k |a\rangle| = \sqrt{2}$. This simplifies to the following:

$$\mathcal{P}_b(|a\rangle) := \begin{cases} 0.5; & \text{if } \bar{b}^k = \bar{a}^k \text{ and } b_k = a_k \\ 0.5; & \text{if } \bar{b}^k = \bar{a}^k \text{ and } b_k = a_k \oplus 1 \\ 0; & \text{otherwise} \end{cases} \tag{4.74}$$

and

$$\mathcal{F}_b(|a\rangle) := \begin{cases} \sqrt{2}; & \text{if } \bar{b}^k = \bar{a}^k \text{ and } a_k b_k = 0 \\ -\sqrt{2}; & \text{if } \bar{b}^k = \bar{a}^k \text{ and } a_k b_k = 1. \end{cases} \tag{4.75}$$

For our choice of frame, we get a particularly simple sampling algorithm which unfortunatly perform poorly in the number of Hadamard gates. This is because when a Hadamard gate acts on a computational basis state (our chosen frame), it produces a state that has maximal l_1 norm (in the computational basis) over all possible states with unit l_2 norm. Nevertheless, for the purpose of illustrating the techniques through an example, we apply Lem. 1 by using s independent samples from the paths appearing in the sum in Eq. (4.70). We note that each sampled path produces a computational basis state with a phase. Thus the l_2 norm of the sample average vector can be exactly computed in run-time[4] $O(ns\log\ s)$. This quantity serves as the estimator $\hat{p}_s$ for the Born rule probability $p = |\langle y|\, U\, |\phi\rangle|^2$. Thus for all $\epsilon > 0$:

$$\Pr\left(|\hat{p}_s - p| \geq \epsilon\right) \leq \exp\left(\frac{-s\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2}{2(2^{h/2}+p)^2}\right), \tag{4.76}$$

where h is an the total number of Hadamard gates appearing in U. Of course, this estimator is not useful because it requires using more than 2^h samples despite the fact that Eq. (4.70) in conjunction with Eq. (4.71) gives a formula for exactly computing $\langle y|\, U\, |\phi\rangle$ that involves the sum of 2^h terms each of which is efficiently computable.

We point out that the situation can be significantly improved by choosing a different model. As an example, we can move to the model presented in Example 4. Here, the choice of $\{\, U_g\,\}_{g\in G} = \langle X_1, \ldots, X_n\rangle$ reproduces Example 2 but we can extend from this base case. As an example, we can consider $\{\, U_g\,\}_{g\in G} = \langle X_1, \ldots, X_n, H_1, \ldots, H_n\rangle$. This now makes Hadamard gates free by increasing the number of elements in the frame. This results in two additional complications. First, the frame element transitions under the action of the Toffoli gate are no longer fully specified by Eq. (4.71) since this now only deals with a subset of the full set of frame elements. As a consequence, when it acts on some of the frame elements, the Toffoli gate will no longer produce a deterministic transition. The run-time cost associated with this may be outweighed by the cost saving associated with free Hadamard gates. The second complication which presents itself whenever the frame is over-complete relates to the choice of function $\alpha : G \to \mathbb{C}$ subject to $\sum_{g\in G} \alpha_g U_g\, |0\rangle\!\langle 0|\, U_g^\dagger = I$. This choice affects run-time and can be greedily chosen at each step to locally minimize run-time. To be specific, suppose we are part way through the estimation protocol and are currently representing the state of the system by some frame element $|\mathcal{B}_h\rangle$ and adjustment factor $\mathcal{F}$. We now wish to act with the next transformation $\tilde{U}$. At this point we can freely choose a function $\alpha : G \to \mathbb{C}$ subject to $\sum_{g\in G} \alpha_g U_g\, |0\rangle\!\langle 0|\, U_g^\dagger = I$ such that the updated state:

$$\tilde{U}\, |\mathcal{B}_h\rangle = \sum_{g\in G} \alpha_g U_g\, |0\rangle\!\langle 0|\, U_g^\dagger \tilde{U}\, |\mathcal{B}_h\rangle \tag{4.77}$$

can be represented by sampling $|\mathcal{B}_g\rangle := U_g\, |0\rangle$ with probability:

$$\mathcal{P}_g(\tilde{U}\, |\mathcal{B}_h\rangle) := \frac{\left|\alpha_g\, \langle 0|\, U_g^\dagger\, |\mathcal{B}_h\rangle\right|}{\mathcal{N}}. \tag{4.78}$$

This will require updating the adjustment factor by multiplying it by:

$$\mathcal{F}_g(\tilde{U}\, |\mathcal{B}_h\rangle) := \mathcal{N} \exp\Big(i \operatorname{Arg}(\alpha_g\, \langle 0|\, U_g^\dagger\, |\mathcal{B}_h\rangle)\Big) \tag{4.79}$$

[4] One way to do this is to use *mergesort* to sort the sampled computational basis states. This will require $O(s\log\ s)$ comparisons of basis states with each basis state comparison achievable in $O(n)$ runtime.

where $\mathcal{N} := \sum_{g\in G} \left|\alpha_g \langle 0| U_g^\dagger |\mathcal{B}_h\rangle\right|$. By choosing α such that $\max_{g\in G} \{ \mathcal{F}_g(\tilde{U} \mid \mathcal{B}_h\rangle) \}$ is minimized, we can in this way greedily locally minimize the adjustment factors (although this may not be a global minimum). In many cases, this optimization can be pre-computed for each gate in the gate-set and conditioned on the the state representative $|\mathcal{B}_h\rangle$ immediately prior to the application of the gate. In particular, this is always possible when the gate-set is generated by gates that act non-trivially on at most one or two qubits and the frame is constructed from a tensor product of single qudit frames. In Sec. 4.3.4 we will see an example of a non-product frame choice (the set of stabilizer states in the mixed state formalism) where this convex optimization is extremely computationally intensive but can nevertheless be applied to give useful upper bounds to the run-time of the simulation algorithm.

4.3.3 Stabilizer frame pure state formalism

In Ref. [2], Bravyi and Gosset presented two classical simulation algorithms. These simulate n-qubit pure quantum systems that are initialized in the computational zero state, evolve under the universal gate set consisting of Clifford and T gates with the first w qubits being measured in the computational basis at the end of the circuit. In both algorithms, the quantum state space is $\mathcal{H}^n = \left(\mathbb{C}^2\right)^n$.

Both algorithms use a notion of simulation that is substantially stronger than additive polynomial precision estimation. The first of these is an exponential time algorithm for approximately sampling from the circuit's output distribution.The second of these is an exponential time algorithm for estimating to multiplicative precision the Born rule probability associated with any specified outcome. We note that this also allows for the estimation of marginal probabilities simply by estimating the outcome probabilities for a slightly modified circuit with a smaller number of measured qubits.

This work appears to be very different for the approach of Ref. [1] but, the two results are nevertheless closely related. The connection between these is captures within the general framework which incorporates some of the multiple important contributions made in Ref. [2]. The recent work of Bravyi et. al. [38] extends and refines the techniques of Ref. [2] to improves on the run-time performance as well as broadening the range of gate-sets considered. This work also focuses on the relationship between the stabilizer rank and the stabilizer extent. The first of these is the crucial parameter that causes the exponential run-time of the algorithms in Ref. [2]. The second of these is a quantity that is closely connected to the negativity in Ref. [1] and is equivalent to the quantity $\mathcal{R}$ of the general framework under a specific choice the models. The algorithm we present in Ch. 5 is a application of the general framework under this specific choice the models, showcasing the connection between these techniques.

Ref. [38] and Ref. [55] also consider a classical simulation technique based on the linear decomposition of unitary gates into Clifford gates. These techniques are currently not incorporated into our general framework but we believe that this extension is useful future work.

4.3.4 Stabilizer frame mixed state formalism

Howard and Campbell [34] further extended probability estimation techniques by considering the qubit stabilizer operator frame in the magic state injection model of quantum computation. Specifically, using a procedure known as gadgetization (this will be discuss later in Sec. 5.3), they are able to convert a Clifford plus T circuit into a Clifford circuit with a number of magic states as inputs. Then the input product state was broken up into blocks of k qubits with each block decomposed into a linear combination of stabilizer states. Then using a sampling techniques very similar to Ref. [1], a stabilizer state is sampled based on the weights in the decomposition. Since the sampled states are now stabilizers, Clifford operations can be cheaply implemented using the Gottesman-Knill theorem [18].

After the gadgetization step, this simulation algorithm can be represented within the general framework by defining the mixed state stabilizer frame $\mathcal{B} = \{ \, |s\rangle\langle s| \mid |s\rangle \in \mathrm{Stab}_n \, \}$ where Stab_n is the set of n-qubit stabilizer states. See also Example 5 for a similar construction and related discussion. We note that the run-time of the simulation protocol scales in a quantity known as the *robustness of magic* (RoM). This quantity is analogous to the negativity of Ref. [1] and to the quantity $\mathcal{R}$ in the present work. Specifically, it is the sum of the absolute values of the coefficients in the stabilizer decomposition of the input magic states.

We note that the stabilizer frame is overcomplete. As a consequence, the decomposition of states into this frame is in general non-unique. Howard and Campbell showed that the RoM can be minimized by using a linear program to find the optimal decomposition into stabilizer states. Further, the RoM of a block of k magic states grows sub-multiplicativly in k. Thus, the simulation run-time can improve by increasing the block size k. However, due to the extremely fast growth in the number of stabilizer states, the optimization protocol is very computation intesive and Howard and Campbell were only able to compute the RoM of magic states for up to blocks of size 5.

By exploiting the symmetries of the stabilizer polytope, Ref. [56] was able to compute the RoM for up to blocks of size 9.

4.3.5 Weyl-Heisenberg frame mixed state formalism

In this section, we consider the use of the Weyl-Heisenberg frame, $\mathcal{B} = \{ \, \frac{1}{d^n} D_{z,x} \mid x, z \in \mathbb{Z}_d^n \, \}$. For the case of qubits ($d = 2$), these operators are the Pauli operatos; thus in the case of qubits, we also refer to this frame as the *Pauli frame*. This choice of model was considered in Example 3. This frame forms a basis of the Hilbert space (is not over-complete) and consequently has a uniquely defined dual frame. It is easy to show that, up to a normalization constant each frame element is dual to itself. Letting $\mathcal{B} = G = \{ \, G_\alpha = \frac{1}{d^n} D_{z,x} \mid x, z \in \mathbb{Z}_d^n \, \}$ and $F = \{ \, D_{z,x} \mid x, z \in \mathbb{Z}_d^n \, \}$ we note that any operator $H \in \mathcal{H}^n$ can be written in the G basis as:

$$\begin{aligned} H &= \sum_{\beta \in \mathbb{Z}_d^{2n}} a_\beta G_\beta \qquad (4.80) \\ &= \frac{1}{d^n} \sum_{\beta \in \mathbb{Z}_d^{2n}} a_\beta D_\beta, \end{aligned}$$

for some coefficients $a_\beta \in \mathbb{C}$. By Eq. (3.14) we note that:

$$\begin{aligned}\operatorname{tr}\left(HF_\alpha^\dagger\right) &= \sum_{\beta\in\mathbb{Z}_d^{2n}} \frac{a_\beta}{d^n}\operatorname{tr}\left(D_\alpha^\dagger D_\beta\right) \\ &= a_\alpha. \end{aligned} \tag{4.81}$$

By substituting Eq. (4.81) into Eq. (4.80), we see that the duality property from Eq. (4.59) is satisfied by the G, F frame pair:

$$\begin{aligned}H &= \sum_{\beta\in\mathbb{Z}_d^{2n}} a_\beta G_\beta \\ &= \sum_{\beta\in\mathbb{Z}_d^{2n}} \operatorname{tr}\left(HF_\alpha^\dagger\right) G_\beta.\end{aligned}$$

Technically, this example does not fit within the model presented in Ref. [1] and summarized in Sec. 4.3.1 because $\mathcal{B} = G$ fails to satisfy the normalization condition Eq. (4.58) and its dual, F, fails to satisfy the corresponding normalization condition Eq. (4.57). In fact, applying the F frame normalization gives the phase point operator at the origin, up to a dimensional factor:

$$\sum_{\alpha\in\mathbb{Z}_d^{2n}} F_\alpha = d^n A_{0,0}, \tag{4.82}$$

and applying the G frame normalization gives:

$$\operatorname{tr}\left(G_\alpha\right) = \delta_{0,\alpha}. \tag{4.83}$$

The normalization condition ensured that the quasi-probabilities constructed from the dual frame were normalized like (conditional) probability distributions. In this case, failing this results in:

$$\sum_\lambda W_\rho(\lambda) = \sum_\lambda \operatorname{tr}\left(\rho F_\lambda^\dagger\right) \tag{4.84}$$

$$= d^n \operatorname{tr}\left(\rho A_{0,0}\right). \tag{4.85}$$

For a general input state ρ, this quantity resides in the interval $[-d^n, d^n]$ and in particular, can result in an exponentially large $\mathcal{R}$ causing inefficiency of the estimation protocol.

We note that the normalization constant of $1/d^n$ can be moved from the G frame to the F frame but this results in a similar issue potentially arising at the end of the circuit (upon measurement resulting in a large factor of $\operatorname{tr}\left(ED_\alpha\right)$) rather than the start. As previously discussed in Sec. 4.3.1, the simulation procedure from Ref. [1] can also be executed in reverse. Thus, one can also consider attempting the estimation procedure in this setting. We note that while these attempts will fail in the most general cases, there are many interesting and/or useful cases where efficient simulation is nevertheless achievable using the Weyl-Heisenberg frame. Ref. [3] presents one such algorithm that we will make use of in Part II. Rall et. al. [37] consider application of this estimation technique based on the qubit Pauli frame in more detail outlining its limitations and the regimes in which it remains efficient and/or outperforms other simulation techniques.

We note that the normalization conditions from Ref. [1] can sometimes be usefully weakened. However, they must for all operators $H \in \mathcal{H}^n$, satisfy the following:

$$\operatorname{tr}(H) = \sum_\lambda \operatorname{tr}\left(HF_\lambda^\dagger\right)\operatorname{tr}(G_\lambda) \tag{4.86}$$

$$= \operatorname{tr}\left(H\sum_\lambda F_\lambda^\dagger \operatorname{tr}(G_\lambda)\right). \tag{4.87}$$

This imposes the weaker normalization requirement that:

$$\sum_\lambda F_\lambda^\dagger \operatorname{tr}(G_\lambda) = I. \tag{4.88}$$

This requirement is indeed satisfied by the Weyl-Heisenberg frame.

Below we present an algorithm from our earlier work [3], where a Pauli frame was used to produce a poly-box for a family of quantum circuits $\mathcal{C}_{\text{PROD}}$.

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 $\mathcal{C}_{\text{PROD}}$. This family consists of quantum circuits with an n-qubit input states ρ is an arbitrary product state[5] (with potentially exponential Wigner function negativity [1] in the input state). The allowed transformations are non-adaptive Clifford unitary gates, and $k \leq n$ qubits are measured at the end of the circuit, in the computational basis. Such a circuit family has been considered by Jozsa and Van den Nest [57], where it was referred to as INPROD, OUTMANY, NON-ADAPT. This circuit family will be discussed again in Sec. 11 where we will show the classical hardness of simulating this family according to another notion of simulation. Aaronson and Gottesman [18] 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 3. *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:

$$\Pi_i = \begin{cases} \frac{I+Z}{2} & \text{if the } i^{th} \text{ entry of } S \text{ is } 0 \\ \frac{I-Z}{2} & \text{if the } i^{th} \text{ entry of } S \text{ is } 1 \\ I & \text{if the } i^{th} \text{ entry of } S \text{ is } \bullet \end{cases} \tag{4.89}$$

[5] 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 Gottesman-Knill theorem [18], 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:

$$\hat{p}_1 := \operatorname{tr}\left(\rho P'\right) = \prod_{i=1}^{n} \operatorname{tr}\left(\rho_i P'_i\right) . \tag{4.90}$$

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,

$$\Pr(|\hat{p}_s - p| \geq \epsilon) \leq 2e^{\frac{-s\epsilon^2}{2}} . \tag{4.91}$$

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. □

4.4 Summary and technical discussion

In this chapter, we presented a generalized framework for the construction of Born rule probability estimators. We discussed a set of free parameters (the model) within the general framework. To elucidate the abstract formalism, we provided some simple, and some more general examples of models. We presented an algorithm that, for a given choice of model, can be used to produce Born rule probability estimates. Through this generalized framework, we consolidated a number of known results in the existing literature.

In our discussion on efficiency constraints, we showed that the quantity $\mathcal{R}$ plays a key role in determining the run-time of the estimation algorithm. We noted that this quantity is driven by the l_1 norm in the decomposition of states into frame elements at each stage of the system's evolution. Through an example in our discussion in Sec. 4.3.2, we showed the complex behavior of the l_1 norms ($\mathcal{N}$) under variation of frame elements. In particular, we showed how the addition of new frame elements can cause costly gates to become free and free ones become costly. We also noted that in the case of overcomplete frames, $\mathcal{R}$ can be minimized by optimizing the linear decomposition into frame elements.

We note that significant run-time advantage is offered by representing a pure state in the pure state formalism rather than in the mixed state formalism. To see this, let us fix a frame

$\mathcal{B}_p = \{ \mid u_i\rangle \}_{i\in[M]}$, a pure state $|\phi\rangle$, and assume the following decomposed of $|\phi\rangle$ into the frame elements is optimal:

$$|\phi\rangle = \sum_{i\in[M]} \alpha_i \, |u_i\rangle \,, \tag{4.92}$$

in the sense that the l_1 norm of the coefficients is minimized, i.e. α minimizes:

$$\mathcal{N}(|\phi\rangle) = \sum_{i\in[M]} |\alpha_i| \,. \tag{4.93}$$

In this case, we note that the representing this state in the mixed state formalism using the corresponding frame $\mathcal{B}_p = \{ \mid u_i\rangle\langle u_j \mid \}_{i,j\in[M]}$ we find that the optimal decomposition must be:

$$|\phi\rangle\langle\phi| = \sum_{i,j\in[M]} \alpha_i\alpha_j \, |u_i\rangle\langle u_j| \,. \tag{4.94}$$

This has an l_1 norm given by:

$$\mathcal{N}(|\phi\rangle\langle\phi|) = \sum_{i,j\in[M]} |\alpha_i\alpha_j| = \mathcal{N}(|\phi\rangle)^2. \tag{4.95}$$

Our general framework has many aspects that are not yet well understood. Going froward, we hope to better understand the transformations that can act on model and how these influence $\mathcal{R}$. For example, consider the set of transformation to the model that leaves the frame unchanged and the state decomposition (given by the product $\mathcal{P} * \mathcal{F}$) unchanged but transforms $\mathcal{P}$ and $\mathcal{F}$. As a second example we can consider the set of transformation that fix the frame but allows the decomposition to vary. This is the set of transformation over which the RoM was minimized in Refs. [34]. Developing better tools for understanding the relationships between model choice and $\mathcal{R}$ is an important pursuit with significant benefits.

Chapter 5

An estimation algorithm for Clifford plus T gates

5.1 Introduction: Clifford plus T algorithm

In this chapter, we present an algorithm for computing additive polynomial precision estimates of Born rule probabilities and marginals. We anticipate that efficient classical methods for Born rule probability estimation will become increasingly important in the near future. With state-of-the-art experimental control over noisy intermediate-scale quantum (NISQ) systems [58], we are now already seeing claims of experimentally observed quantum supremacy [59]. Thus, between now, the age of NISQ, and the age of universal, fault tolerant quantum devices there will inevitably be a need to compare classically computed theoretical predictions with the observed frequency of particular events generated by a quantum device. Techniques such as direct fidelity estimation [60] already rely on such comparisons. We expect the rapidly developing field of quantum characterization, validation and verification to make increasingly important contributions using innovations in estimation of Born rule probabilities [61].

In this setting, we argue that developments in *additive polynomial precision* estimation are particularly important. As per our discussion in Sec. 4.1, by repeatedly running a quantum device, experimentally observed frequencies of the occurrence of a particular event can be used to estimate the probability of the occurrence of this event. These estimates will be additive inverse polynomial precision in the number of independent trials. For correctly functioning quantum devices, these probabilities should be consistent with the theoretical predictions given by the Born rule. In comparing the consistency between the probability of the event's occurrence using the quantum device with the associated Born rule probability, two sources of error arise. The first of these is the statistical error between the observed frequency and the true probability of the quantum event occurring on any independent trial executed on the quantum device. The second source is the estimation error associated with the classical estimation protocol. The effectiveness of the statistical tests for checking consistency are driven by the sum of the two errors. As a consequence, the marginal benefits of suppressing the classical estimation error diminish as the statistical error starts to dominate. Hence, assuming we are interested in hard to classically compute Born rule probabilities, the run-time to precision trade-off for classical

Born rule estimation will be chosen such that the estimation error is comparable to the statistical error. For these reasons, we believe that additive polynomial precision estimation algorithms are effective in just the right parameter regime to play an important role in the characterization and non-adversarial verification of near term quantum devices.

The algorithm we present here is an application of the general framework presented in Ch. 4 to the universal circuit family considered in Ref. [2]. In particular, this algorithm is based on the pure state formalism using the set of qubit stabilizer states as the choice of frame. The algorithm simulates n-qubit pure quantum systems that are initialized in the computational zero state, evolve under the universal gate set consisting of Clifford and T gates with the first w qubits being measured in the computational basis at the end of the circuit.

Our algorithm has a run-time that scales exponentially in t, the number of T gates in the quantum circuit. In particular, ignoring polynomial factors, the run-time scales like $2^{\gamma t}$ where $2^{\gamma} \approx 2^{0.228}$ is known as the *stabilizer extent* of the $|T\rangle$ state [38]. This exponential component of run-time is identical to that of the sampling algorithm presented in Ref. [2], which we will refer to as the Bravyi-Gosset (BG) sampling algorithm. Compared to the run-time of the BG sampling algorithm, we anticipate our algorithm to exhibit modest improvements in the polynomially scaling pre-factor to the exponential component of the run-time.

For the specific task of fast Born rule probability estimation, there is an additional run-time cost associated with modifying the BG sampling algorithm. The BG sampling algorithm is more flexible and can be used to produce additive polynomial precision estimates of Born rule probabilities and marginals. However, we note that this conversion, from samples to estimates carries an additional polynomial overhead that contributes to further widening the expected run-time differential, compared to the algorithm we present in this chapter. Although the run-time differential will depend on many unknowns including the choice of parameters, we believe it is reasonable to expect a few orders of magnitude improvement in run-time for simulations that are currently near the classical computational limits.

To compare our estimation algorithm with that of the BG multiplicative precision estimation algorithm, we consider run-time and accuracy. In the case where the Born rule probability is large, say > 0.1, the multiplicative precision estimates have errors comparable to our estimates. However, with each order of magnitude reduction in the Born rule probability, errors due to multiplicative precision estimates must improve by an order of magnitude while the errors due to additive precision estimates remain unchanged. Thus, BG's multiplicative precision estimation algorithm achieves substantially higher precision estimates particularly for low Born rule probabilities. In comparing run-times, to leading order our algorithm (as well as BG's sampling algorithm) are substantially faster for high T gate count circuits. In our algorithm, the exponential component of run-time scales as approximately $2^{0.228t}$ compared to $2^{0.47t}$ in BG's estimation algorithm. This results in a substantial run-time saving for modest T counts.

The algorithm we present here is a work in progress. In Sec. 5.2, we commence with an overview describing the scope of the algorithm and results. Then we first describe a simple algorithm that showcases the main techniques and ideas that are employed. This is covered in four steps in Secs. 5.3 to 5.6. In Sec. 5.7 we discuss the run-time of this algorithm. In Sec. 5.8, we discuss a modification to this algorithm aimed at sharpening the run-time followed by the run-time analysis. In Sec. 5.9 we discuss methods for choosing two particular algorithmic parameters in order to sharpen run-time. In Sec. 5.10 we conclude with an outlook.

5.2 Overview of the estimation algorithm

We consider a system composed of n computational qubits, initially prepared in a computational zero state $|0\rangle^{\otimes n}$, which we will denote by $|0^n\rangle$. The system then evolves to the final state $|\psi_U\rangle$ according to a unitary transformation U synthesized from the gate-set consisting of T, H, S, CX and CZ gates (see Sec. 3.2 for definitions).

$$|\psi_U\rangle := U\,|0^n\rangle \tag{5.1}$$

Given some ordered subset $J = \{\, j_1, \ldots, j_w \,\} \subseteq [n]$ of qubits to be measured in the computational basis and some outcome $\boldsymbol{x} = (x_1, \ldots, x_w) \in \{0,1\}^w$, our aim is to estimate the probability $p = p(J, \boldsymbol{x})$ of observing the outcome $\boldsymbol{x}$ when measuring the final state, i.e., we want to estimate

$$p := \|P\,|\psi_U\rangle\,\|^2, \tag{5.2}$$

where

$$P = \bigotimes_{i=1}^{w} \frac{I_{j_i} + (-1)^{x_i} Z_{j_i}}{2} \bigotimes_{k \notin J} I_k \tag{5.3}$$

is the projector onto the target outcome $\boldsymbol{x}$. Without loss of generality, we will assume that the first w qubits are measured and hence $J = \{\, 1, \ldots, w \,\}$. Since it is known that a general n-qubit unitary circuit can be approximated arbitrarily well by a circuit composed only of Clifford and T gates, we will focus only on estimating the probability for circuits composed of t instances of the T gates and c Clifford gates. Then, our main result is captured by the following theorem.

Theorem 4. *Suppose an n-qubit unitary U can be written as a product consisting of c one- or two-qubit Clifford gates from H, S, CX and CZ; and t single-qubit T gates. Then, there exists a classical algorithm that can output an estimate $\hat{p}$ of the outcome probability p such that for all $\epsilon_{\text{tot}}, \delta_{\text{tot}} > 0$:*

$$\Pr\left(|\hat{p} - p| \geq \epsilon_{\text{tot}}\right) \leq \delta_{\text{tot}} \tag{5.4}$$

with the dominant exponential component of run-time scaling as $\tau_{\text{exp}} \sim \tilde{O}\left(2^{\gamma t} t^3 \epsilon_{\text{tot}}^{-4}\right)$, where $\gamma \approx 0.228$ is a constant and the tilde hides logarithmic factors in the run-time.

In the above theorem, we have omitted logarithmic components of run-time and components that do not scale exponentially in t. We have also omitted additional run-time components that scale like $\tilde{O}\left(2^{\gamma t/2} t^3 \epsilon_{\text{tot}}^{-4}\right)$ or more slowly. Not omitting these exponential components, the run-time scales like $\tilde{O}\left((2^{\gamma t/2} + 1)^2 t^3 \epsilon_{\text{tot}}^{-4}\right)$. These simplifications have been applied in order to focus on what is likely to be the most significant component of run-time. A more detailed discussion of run-time is presented in Sec. 5.8.

In Thm. 4, we have provided a pessimistic upper bounds to the run-time. However, as per our discussion in Sec. 5.9, we expect that our algorithm will perform with run-time that is closer to:

$$\tau_{\text{exp}} \sim \tilde{O}\left(2^{\gamma t} t^3 p^3 \epsilon_{\text{tot}}^{-4}\right); \quad \text{when} \quad \epsilon_{\text{tot}} \ll p. \tag{5.5}$$

resulting in significantly shorter run-time for instances that have small Born rule probabilities.

Eq. (5.5) is a simplification of a more complicated expression (see eqs. (5.61) and (5.62)) in the regime $\epsilon_{\text{tot}} \ll p$. Thus one should exercise caution in interpreting this result. In particular, it is not valid to take the limit $p \to 0$ while holding ϵ_{tot} constant. By simplifying the more complicated expressions in eqs. (5.61) and (5.62) under this limit, it is easy to show that as $p \to 0$, $\tau_{\text{exp}} \to \tilde{O}(2^{\gamma t} t^3 \epsilon_{\text{tot}}^{-1})$.

We note that the run-time of the BG sampling algorithm is $\tilde{O}(2^{\gamma t} t^3 w^3 \epsilon^{-4})$ where we have again ignored logarithmic factors and all run-time components that do not scale exponentially in t. To convert this into an estimation algorithm that is with high probability accurate to within ϵ additive error generically requires $O(\epsilon^{-2})$ samples. This gives a total run-time of $O(2^{\gamma t} t^3 w^3 \epsilon^{-6})$ for estimation using the BG sampling algorithm. Thus the run-time for estimation based on the BG sampling algorithm is larger than the dominant exponential component of our algorithm's expected run-time by a factor of $\tilde{O}(w^3 \epsilon^{-2} p^{-3})$. This is likely to result in a large reduction in run-time particularly for circuits where multiple qubits are measured.

The algorithm we will use to prove Thm. 4 is an application of the general framework presented in Ch. 4. This algorithm is based on the pure state formalism using the set of qubit stabilizer states as the choice of frame. We will use magic state injection post-selected on the all zero outcome to inject the T gates. At a conceptual level, the basic steps of the simple algorithm we present in Secs. 5.3 to 5.6 are as follows:

1. Re-express the circuit, from Eq. (5.1), as a post-selected Clifford circuit with a magic state input (see Eq. (5.7)).

2. Express the input magic state as a (typically exponential) linear combination of stabilizer states (see Eq. (5.14)).

3. Iterate the following procedure:

   (a) uniformly sample from the stabilizer states appearing in the (typically exponential) linear combination;

   (b) propagate the sampled stabilizer state through the Clifford circuit;

   (c) project this onto the relevant measurement and post-selection outcome.

   Using the results from all of the iterations, form an equally weighted superposition (sample average) as per Eq. (5.25).

4. Use the fast norm estimation algorithm from Ref. [38] to compute the norm of the result. The square of this quantity is an estimate of the target probability.

The details of this simple algorithm will be modified in Sec. 5.8 to sharpen run-time. This modification introduces an additional step after step 1 and makes step 3(b) unnecessary. We now proceed to a step-by-step description of the protocol.

5.3 Step 1: Gadgetization and post-selection

It is well known that a T gate acting on a qubit j can be replaced by its gadgetized version [62]. More precisely, one can prepare an ancillary qubit in a magic state

$$|T\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \exp(i\pi/4)\,|1\rangle), \tag{5.6}$$

couple it to qubit j by a CNOT gate (with the qubit j acting as the control) and measure in the computational basis. Then, if the outcome is $|1\rangle$, one also needs to apply a correction Clifford phase gate S to qubit j. The effect of the above procedure is the same as direct application of the T gate to qubit j. We can gadgetize each of the t occurrences of the T gate in this way. Hence, we can replace a general unitary circuit U on n qubits in a state $|0^n\rangle$ by a circuit on $n+t$ qubits in a state $|0^n\rangle_c \otimes |T^t\rangle_m$. Here, we use a subscript m to denote the magic states register prepared in the state $|T\rangle^{\otimes t}$. Similarly, we use a subscript c to denote the computational register of n qubits. The variable c representing the number of Clifford gates in U should not be confused with this subscipt. This gadgetized circuit is composed of Clifford gates and classically controlled Clifford gates that depend on computational basis measurement of the magic register.

Now, one can easily show that each of the 2^t measurement outcomes arising through gadgetization is equally likely. Therefore, we can focus on just one particular outcome – for simplicity chosen to be the all zeros outcome (as no correction gates are then needed) – and estimate the target probability using this simplified post-selected circuit. More precisely, let us denote by V the unitary acting on $n+t$ qubits and composed of $c+t$ Clifford gates: the original c Clifford unitaries appearing in the decomposition of U into Cliffords and T gates, plus t CNOT gates between computational and ancillary qubits arising from gadgetization of T gates. Then, the evolution induced by V on $|0^n\rangle_c \otimes |T^t\rangle_m$ with the post-selection on the all zero outcome (for ancillary qubits) has the same effect (up to a normalization factor) on the computational qubits as the original unitary U applied to $|0^n\rangle_c$, i.e.,

$$|\psi_U\rangle := 2^{t/2}\left(I^n \otimes \langle 0^t|_m\right) V \left(|0^n\rangle_c \otimes |T^t\rangle_m\right). \tag{5.7}$$

5.4 Step 2: Pure stabilizer decomposition of magic states

Now that we have expressed $|\psi_U\rangle$ as Clifford evolution of computational and magic states, the next step is to decompose the state of the magic register $|T^t\rangle_m$ into a superposition of stabilizer states. This will allow us to employ the known results on classical simulation of the evolution of stabiliser states under Clifford gates (the Gottesman-Knill theorem) in the next step of our estimation algorithm. We will then see that the crucial quantity responsible for the scaling of the run-time will be given by the *stabilizer extent* ξ [38],

$$\xi(|\phi\rangle) := \min_{\boldsymbol{c}} \left\{ \|\boldsymbol{c}\|_1^2 \;\middle|\; |\phi\rangle = \sum_j c_j\,|s_j\rangle\,, |s_j\rangle \in \mathrm{Stab} \right\}. \tag{5.8}$$

An optimal decomposition of the $|T\rangle$ state, i.e., one that achieves a square l_1 norm equal to the stabilizer extent $\xi(|T\rangle)$, is given by

$$|T\rangle = \alpha\,|\tilde{0}\rangle + \alpha^*\,|\tilde{1}\rangle\,, \quad \alpha = \frac{1+i(\sqrt{2}-1)}{2} \tag{5.9}$$

where we introduced the following non-standard notation for $|+\rangle$ and $|+i\rangle$ states:

$$|\tilde{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |\tilde{1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle). \tag{5.10}$$

The stabilizer extent is thus equal to

$$\xi(|T\rangle) = 4 - 2\sqrt{2} \approx 1.17, \tag{5.11}$$

and we also define its logarithm in base 2 (which will be a crucial quantity appearing in the scaling of the run-time of our algorithm):

$$\gamma := \log_2 \xi(|T\rangle) \approx 0.228. \tag{5.12}$$

Moreover, as proven in Ref. [38], the stabilizer extent for products of single-qubit states is multiplicative, so

$$\xi\left(|T\rangle^{\otimes t}\right) = \xi(|T\rangle)^t = 2^{\gamma t}, \tag{5.13}$$

and thus the optimal decomposition of the magic register is simply given by

$$|T^t\rangle_m = \left(\alpha\,|\tilde{0}\rangle + \alpha^*\,|\tilde{1}\rangle\right)^{\otimes t} = \sum_{y\in\{0,1\}^t} \alpha^{t-|y|}(\alpha^*)^{|y|}\,|\tilde{y}\rangle_m\,, \tag{5.14}$$

where $|y|$ denotes the Hamming weight of y.

5.5 Step 3: Sampling from stabilizer decomposition

The first two steps allow us to express $|\psi_U\rangle$ as a superposition of stabilizer states evolved under Clifford circuit V and projected (with a normalization factor) on the all zero state of the magic register. More precisely, combining Eqs. (5.7) and (5.14), we get

$$|\psi_U\rangle = 2^{t/2}\left(I^n \otimes \langle 0^t|_m\right) \sum_{y\in\{0,1\}^t} \alpha^{t-|y|}(\alpha^*)^{|y|} V\left(|0^n\rangle_c \otimes |\tilde{y}\rangle_m\right). \tag{5.15}$$

Contracting this state with $\langle x|_J$, representing the measurement outcome of interest, produces the vector:

$$|\mu\rangle := 2^{t/2}\left(\langle x|_J \otimes \langle 0^t|_m\right) \sum_{y\in\{0,1\}^t} \alpha^{t-|y|}(\alpha^*)^{|y|} V\left(|0^n\rangle_c \otimes |\tilde{y}\rangle_m\right). \tag{5.16}$$

The square of the l_2 norm of $|\mu\rangle$ is the target probability from Eq. (5.2)

$$p = \||\mu\rangle\|_2^2\,. \tag{5.17}$$

Note that each term of the sum from Eq. (5.16) can be efficiently calculated using Gottesman-Knill theorem. However, there are 2^t of those terms, and so the exact calculation of p scales as 2^t with the number of t gates. Instead, we will now explain how to estimate p with additive error, but with the run-time scaling considerably better in t, as $2^{\gamma t}$. This will be achieved in two further steps (steps 3 and 4). In step 3, we will use a probabilistic sampling procedure to sample s unnormalized stabilizer vectors. The sample average of these vectors, denoted by $\overline{|\Psi\rangle}$ (see Eq. (5.25)) will be used to estimate $|\mu\rangle$. In step 4, we will use BG's fast norm estimation algorithm to estimate the l_2 norm of $\overline{|\Psi\rangle}$ giving us an estimator for $\|\mu\|_2 = \sqrt{p}$.

The main technical tool that we will employ in step 3 is Lem. 1, the generalization of the Hoeffding's inequality [52] from estimating real parameters to estimating complex vectors. Lem. 1 and its proof can be found in Sec. 4.2.3. The original theorem of Hoeffding, can be found in Appendix A.

For the reader's convenience we restate Lem. 1 here.

Lemma 5 (Restated lemma). *Let $\chi, s \in \mathbb{N}$ and $m > 0$. Let $\mathcal{FB} := \{\ |\ \phi_j\rangle\ \}_{j\in[\chi]}$ be a set of D-dimensional vectors over $\mathbb{C}$ with lengths $\||\phi_j\rangle\|_2 \leq m$. Let $\mathcal{P} := \{\,\mathcal{P}_j\,\}_{j\in[\chi]}$ be a probability distribution over $[\chi]$ and define $|\mu\rangle$ as the D-dimensional vector over $\mathbb{C}$ that is the expectation of $|\phi_X\rangle$ with respect to the random variable X with probability distribution $\mathcal{P}$:*

$$|\mu\rangle = \mathop{\mathbb{E}}_{X\sim\mathcal{P}}[|\phi_X\rangle] = \sum_{j\in[\chi]} \mathcal{P}_j\, |\phi_j\rangle\,. \tag{5.18}$$

For $j \in [s]$, let $x_j \in [\chi]$ be independently sampled from the probability distribution $\mathcal{P}$. We define a vector sample mean over s samples by:

$$\overline{|\phi\rangle} = \frac{1}{s}\sum_{j=1}^{s} |\phi_{x_j}\rangle. \tag{5.19}$$

Then for all $\epsilon > 0$:

$$\Pr\left(\left\|\overline{|\phi\rangle} - |\mu\rangle\right\|_2 \geq \epsilon\right) \leq 2e^2 \exp\left(\frac{-s\epsilon^2}{2(m+p)^2}\right) \tag{5.20}$$

where $p := \||\mu\rangle\|_2$.

The following procedure will be repeated s times, with s depending on the circuit and the estimation accuracy we want to achieve. First, with a uniform probability $q(y) = 2^{-t}$ we sample a bit string y of length t, corresponding to the stabilizer states $|\tilde{y}\rangle_m$ appearing in Eq. (5.16). Next, we combine the sampled state with the initial state of the computational qubits, $|0^n\rangle_c$, and use the Gottesmann-Knill algorithm to evolve the joint stabilizer state $|0^n\rangle_c \otimes |\tilde{y}\rangle_m$ under a Clifford circuit V into the final stabilizer state $|\psi(y)\rangle$.

$$|\psi(y)\rangle = V\,|0^n\rangle_c \otimes |\tilde{y}\rangle_m \tag{5.21}$$

We then compute the inner product between the $(w+t)$ qubit bra vector $\langle x|_J \otimes \langle 0^t|_m$ and the

$(n+t)$ qubit ket vector $|\psi(y)\rangle$. This outputs the $(n-w)$ qubit unnormalized state:

$$|\Psi(y)\rangle := 2^{3t/2}\alpha^{t-|y|}(\alpha^*)^{|y|} \ \left(\langle x|_J \otimes \langle 0^t|_m\right)|\psi(y)\rangle \,. \tag{5.22}$$

Note that since $|x\rangle_J \langle x|_J \otimes |0^t\rangle_m \langle 0^t|_m$ is a projector onto a stabiliser codespace and $|\psi(y)\rangle$ is a stabilizer state, $|\Psi(y)\rangle$ is itself an unnormalzed stabilizer state. Our algorithm will require s independent samples $|\Psi_1\rangle, \ldots, |\Psi_s\rangle$ with $|\Psi_i\rangle$ for each $i \in [s]$ selected by uniformly sampling $y_i \in \{0,1\}^t$ then computing $|\Psi(y_i)\rangle$.

Now, comparing with eqs. (5.16), we see that the expectation[1] of the random vector $|\Psi\rangle$, which is constructed by uniformly sampling y then computing $|\Psi(y)\rangle$, is given by:

$$\begin{aligned}
\mathop{\mathbb{E}}_{|\Psi\rangle\sim\mathcal{U}}[|\Psi\rangle] :&= \sum_{y\in\{0,1\}^t} 2^{-t}|\Psi(y)\rangle = 2^{t/2}\sum_{y\in\{0,1\}^t}\alpha^{t-|y|}(\alpha^*)^{|y|}\left(\langle x|_J \otimes \langle 0^t|_m\right)|\psi(y)\rangle \\
&= 2^{t/2}\sum_{y\in\{0,1\}^t}\alpha^{t-|y|}(\alpha^*)^{|y|}\left(\langle x|_J \otimes \langle 0^t|_m\right)V\left(|0^n\rangle_c \otimes |\tilde{y}\rangle_m\right) \\
&= |\mu\rangle \,,
\end{aligned} \tag{5.23}$$

and so, by Eq. (5.17), we see that

$$p = \left\| \mathop{\mathbb{E}}_{|\Psi\rangle\sim\mathcal{U}}[|\Psi\rangle] \right\|_2^2 . \tag{5.24}$$

For some finite sample size s, we will aim to use the sample average $\overline{|\Psi\rangle}$,

$$\overline{|\Psi\rangle} = \frac{1}{s}\sum_{i=1}^{s}|\Psi_i\rangle, \tag{5.25}$$

as an approximation to the expectation value. We will thus employ Lem. 1 to bound the departure of the sample average vector from the expectation vector. Before that, let us first bound the norm of $|\Psi(y)\rangle$. We have

$$\begin{aligned}
\langle\Psi(y)|\Psi(y)\rangle &= 2^{3t}\ |\alpha|^{2t}\ \langle\psi(y)|\left(P\otimes|0^t\rangle_m\langle 0^t|_m\right)|\psi(y)\rangle \\
&\le 2^{3t}\ |\alpha|^{2t}\ 2^{-t} = (2|\alpha|)^{2t} = \xi(|T\rangle)^t = 2^{\gamma t},
\end{aligned} \tag{5.26}$$

where the inequality comes from the fact that the qubits in the magic register of $|\psi(y)\rangle$ were initially prepared in states $|\tilde{0}\rangle$ and $|\tilde{1}\rangle$ and, under the evolution by V, were only acted on by CX gates as target qubits, so they did not change the overlap with the computational basis states, which initially was $1/\sqrt{2}$ per magic qubit. Using Lem. 1 we now get:

$$\Pr\left(\left\|\overline{|\Psi\rangle} - \mathop{\mathbb{E}}_{|\Psi\rangle\sim\mathcal{U}}[|\Psi\rangle]\right\|_2 \ge \varepsilon\right) \le 2e^2\exp\left(\frac{-s\varepsilon^2}{2(2^{\gamma t/2}+p)^2}\right). \tag{5.27}$$

As per the derivation of Eq. (4.47), we first apply the reverse triangle inequality, and use Eq. (5.24)

[1] this is a vector with components given by expectation values of the components of the input vector

to get:

$$\Pr\left(\left|\left\|\overline{|\Psi\rangle}\right\|_2 - \sqrt{p}\right| \geq \varepsilon\right) \leq 2e^2 \exp\left(\frac{-s\varepsilon^2}{2(2^{\gamma t/2}+p)^2}\right). \tag{5.28}$$

Then, by noting that for $\lambda, p \in \mathbb{R}$, if $\left|\sqrt{\lambda} - \sqrt{p}\right| \leq \epsilon$ then $|\lambda - p| \leq \epsilon(\epsilon + 2\sqrt{p})$, we find:

$$\Pr\left(\left|\left\|\overline{|\Psi\rangle}\right\|_2^2 - p\right| \geq \varepsilon(\varepsilon + 2\sqrt{p})\right) \leq 2e^2 \exp\left(\frac{-s\varepsilon^2}{2(2^{\gamma t/2}+p)^2}\right). \tag{5.29}$$

We will denote $\left\|\overline{|\Psi\rangle}\right\|_2^2$ by λ. Thus for all $\epsilon > 0$:

$$\Pr\left(|\lambda - p| \geq \epsilon\right) \leq 2e^2 \exp\left(\frac{-s\left(\sqrt{p+\epsilon} - \sqrt{p}\right)^2}{2(2^{\gamma t/2}+p)^2}\right) =: \delta. \tag{5.30}$$

We conclude that the squared l_2 norm of an unnormalized state $\overline{|\Psi\rangle}$ created by an equal superposition (with coefficients $1/s$) of s states sampled according to the described procedure can get ϵ close to p, with probability greater than $1 - \delta$, as long as s is sufficiently large:

$$s \geq \frac{2(2^{\gamma t/2}+p)^2}{\left(\sqrt{p+\epsilon} - \sqrt{p}\right)^2} \log \left(\frac{\delta}{2e^2}\right)^{-1}. \tag{5.31}$$

We note that s, the number of samples required to satisfy Eq. (5.30) depends on the unknown quantity p. For now, we note that a conservative choice of s can always be made by setting $p = 1$. However, this reduces sharpness in the run-time. Instead, for now we will treat s as a variable and we will discuss how s is to be chosen in Sec. 5.9.

Let us now point out a slight complication that is encountered in computing the inner product in Eq. (5.22). We note that the standard technique for computing the inner product between two stabilizer states Ref. [63] produces the inner product up to a global phase. We will be concerned with computing the sum of multiple such inner products. Thus, for our purposes, the relative global phase of each term will be important. There exist a number of techniques for efficiently computing stabilizer inner products while remaining sensitive to global phase information Ref. [64]. For concreteness, we will use the CH-form to represent and manipulate stabilizer states. This was developed by Bravyi et. al. in Ref. [38].

5.6 Step 4: Fast norm estimation

Eq. (5.30) shows that the target Born rule probability can be estimated by λ, the norm of the sample mean given in Eq. (5.25). The last remaining problem then is to calculate λ.

BG developed an algorithm for producing an estimate $\hat{\phi}$ of the norm of any linear combination $|\phi\rangle$ of s stabilizer states for n-qubit system. The norm of such a state can be estimated with multiplicative precision, i.e., for any desired error level $\tilde{\epsilon} > 0$ and confidence parameter $\tilde{\delta} > 0$ we get

$$\Pr\left(\left|\hat{\phi} - \||\phi\rangle\|_2^2\right| \geq \tilde{\epsilon}\||\phi\rangle\|_2^2\right) \leq \tilde{\delta}, \tag{5.32}$$

and the run-time of the algorithm scales as:

$$O(sn^3\ \tilde{\epsilon}^{-2}\log\tilde{\delta}^{-1}). \tag{5.33}$$

The fast norm estimation algorithm works as follows. As an input, the algorithm is given an n qubit state $|\phi\rangle$ as a linear combination of s stabilizer states:

$$|\phi\rangle = \frac{1}{s}\sum_{j=1}^{s}|\phi_j\rangle \tag{5.34}$$

where $|\phi_j\rangle$ are n qubit stabilizer states. To proceed, we generate L randomly sampled n qubit stabilizer states $|\theta_1\rangle, \ldots, |\theta_L\rangle$. The estimate $\hat{\phi}$ is computed by first computing the quantities:

$$\tilde{\phi}_i := \langle\theta_i|\,\phi\rangle = \frac{1}{s}\sum_{j=1}^{s}\langle\theta_i|\,\phi_j\rangle \tag{5.35}$$

for each $i \in [L]$. Then, the estimate $\hat{\phi}$ is computed using:

$$\hat{\phi} = \frac{2^n}{L}\sum_{k=1}^{L}\left|\tilde{\phi}_k\right|^2. \tag{5.36}$$

The run-time of this algorithm is $O(sn^3L)$ and by choosing:

$$L = \left\lceil\tilde{\epsilon}^{-2}\log\tilde{\delta}^{-1}\right\rceil, \tag{5.37}$$

we ensure that Eq. (5.32) is satisfied.

Let us denote the estimate of $\lambda := \left\|\overline{|\Psi\rangle}\right\|_2^2$ produced using the fast norm algorithm by $\hat{p}$. The estimate $\hat{p}$ is the final output of our algorithm and is an estimate of the target probability p. We now combine the two sources of error (arising from steps 3 and 4) to show that, for an appropriate choice of s and L, our estimate satisfies Eq. (5.4). First, we can employ the triangle inequality to obtain

$$|\hat{p} - p| = |\hat{p} - \lambda + \lambda - p| \leq |\hat{p} - \lambda| + |\lambda - p|. \tag{5.38}$$

From Eq. (5.32) we have that with probability larger than $1 - \tilde{\delta}$ the following holds:

$$|\hat{p} - \lambda| \leq \tilde{\epsilon}\lambda \leq \tilde{\epsilon}\left(|\lambda - p| + p\right). \tag{5.39}$$

Then, from Eq. (5.30) we get that with probability larger than $1 - \delta$ we have

$$|\lambda - p| \leq \epsilon. \tag{5.40}$$

Since both algorithms (estimating the expectation vector and estimating its norm) are independent, we get that with probability larger than $(1-\delta)(1-\tilde{\delta})$ we have

$$|\hat{p}-p| \leq \tilde{\epsilon}(\epsilon+p)+\epsilon. \tag{5.41}$$

We can thus write

$$\Pr\left(|\hat{p}-p| \geq \tilde{\epsilon}(\epsilon+p)+\epsilon\right) \leq \tilde{\delta}+\delta. \tag{5.42}$$

For any choice of $\epsilon_{\text{tot}}, \delta_{\text{tot}} > 0$, $\hat{p}$ satisfies:

$$\Pr\left(|\hat{p}-p| \geq \epsilon_{\text{tot}}\right) \leq \delta_{\text{tot}}, \tag{5.43}$$

for every choice of $\epsilon \in (0, \epsilon_{\text{tot}})$ and $\delta \in (0, \delta_{\text{tot}})$ whenever:

$$s \geq \frac{2(2^{\gamma t/2}+p)^2}{\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2} \log\left(\frac{\delta}{2e^2}\right)^{-1}; \tag{5.44}$$

with

$$\tilde{\epsilon} = \frac{\epsilon_{\text{tot}}-\epsilon}{p+\epsilon}; \quad \text{and} \quad \tilde{\delta} = \delta_{\text{tot}}-\delta. \tag{5.45}$$

By using Eq. (5.37), this gives the number of iterations, L, required for the fast norm estimation:

$$L \geq \left(\frac{p+\epsilon}{\epsilon_{\text{tot}}-\epsilon}\right)^2 \log \tilde{\delta}^{-1}. \tag{5.46}$$

As per the case for the required number of samples s, we see that the choice of L also depends on the unknown p. For now, we note that a conservative choice of L can always be made by setting $p = 1$. However, this reduces sharpness in the run-time. Instead, for now we will treat L as a variable and we will discuss how L is to be chosen in Sec. 5.9.

5.7 Analysis of run-time

We now discuss the performance of this algorithm before considering possible modifications to improve run-time. The key computational steps in this algorithm are:

1. Compute V, the gadgetized Clifford circuit.
2. For each of the s independent trials, uniformly sample a bit-sting $y \in \{0,1\}^t$.
3. For each of the s independent trials, compute the evolution of $|0^n\rangle_c \otimes |\tilde{y}\rangle_m$ to $|\psi(y)\rangle$ as per Eq. (5.21).
4. For each of the s independent trials, compute the contraction of $|\psi(y)\rangle$ to $|\Psi(y)\rangle$ as per Eq. (5.22).
5. Compute the estimate $\hat{p}$ using L iterations of BG's fast norm estimation algorithm.

We point out the this algorithm is not fully specified as we have not specified a means for determining s and L which depend on (the unknown) p. For now we will assume that these take the minimal values specified by eqs. (5.44) and (5.46) and return to this issue in Sec. 5.9.

Let us denote the run-time associated with each of these steps as $\tau_1, \ldots, \tau_5$ respectively. The τ_1 component of run-time is $O(c+t)$ assuming the circuit is specified as a length $c+t$ list of gates with associated registers that they act on. The τ_2 component of run-time is $O(st)$ assuming uniform bit sampling is done in time $O(1)$. Here, we note that τ_2 and all other run-time components that involve s repetitions are parallelizable.

The τ_3 component of run-time is $O(s[(c+t-h)(n+t)+h(n+t)^2])$ where h is the number of Hadamard gates in V and $c+t-h$ is the number of remaining gates (these must be non-Hadamard Clifford gates). To see this, we note that the state initially starts off as $|0^n\rangle_c \otimes |\tilde{y}\rangle_m$ represented in CH-form and is subsequently updated by each gate in V as per Eq. (5.21). The CH-form can be used to update an n qubit stabilizer state by the action of a Clifford gate in time $O(n)$ if the gate is a CX, CZ or phase gate and, in time $O(n^2)$ if it is a Hadamard gate. Applying these runtimes to a system with $n+t$ qubits, we arrive at the claim.

We expect the τ_4 component of run-time to be $O(s(n+t)^2(w+t))$. For an n qubit stabilizer state in CH-form acted on by a projector of the form $(I+P)/2$, where P is a Pauli operator, the resultant n qubits stabilizer state can be computed in CH-form in time $O(n^2)$. The formalism outlined in Ref. [38] does not specify how to contract an n qubit stabilizer state in CH-form with a $k<n$ qubit stabilizer state in CH-form to attain a $n-k$ qubit stabilizer state in CH-form. Due to the non-uniqueness of the CH-form, this computation is non-trivial and our proposed method for computing the outcome has not been rigorously analyzed for run-time. We nevertheless expect that this run-time will be $O(n^2k)$.

Conjecture 1. *Given an n qubit stabilizer state $|\psi\rangle$ in CH-form with a $k<n$ qubit stabilizer state $|\phi\rangle$ in CH-form, there is a classical procedure that outputs the $n-k$ qubit stabilizer state $\langle\phi|\psi\rangle = |\tilde{\psi}\rangle$ in CH-form in time $O(n^2k)$ assuming that the action of each Clifford gate on a Pauli operator can be computed in time $O(1)$.*

Assuming Conjecture 1 and noting that we will be contracting an $n+t$ qubit state with an $w+t$ qubit state to get a $n-w$ qubit state, we arrive at our expected run-time τ_4.

The τ_5 component of run-time is $O(s(n-w)^3L)$. This can be attained by setting the number of qubits to $n-w$ in the run-time of BG's fast norm estimation algorithm.

Putting all of the steps together, we arrive at an overall run-time for the simple algorithm given by:

$$\begin{aligned} &O\left(\tau_1+\tau_2+\tau_3+\tau_4+\tau_5\right) \\ &\quad = O\left(c+t+s\left[t+(c+t-h)(n+t)+h(n+t)^2+(n+t)^2(w+t)+(n-w)^3L\right]\right) \end{aligned} \tag{5.47}$$

From Eq. (5.44) we see that s contains the exponential dependence on t. Thus, it is important to minimise the polynomial run-time factor associated with s. Instead of the direct approach resulting in the run-time of Eq. (5.47), in Sec. 5.8 we discuss a modification that significantly reduces this run-time in most cases of interest.

Using Eqs. (5.44), (5.45) and assuming Conjecture 1, the total time of the estimation algorithm that satisfies Eq. (5.43), scales as

$$O\left(c+t+\frac{2(2^{\gamma t/2}+p)^2\Gamma}{\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2}\log\left(\frac{\delta}{2e^2}\right)^{-1}\right), \tag{5.48}$$

with,

$$\Gamma := \left(t+(c+t-h)(n+t)+h(n+t)^2+(n+t)^2(w+t)+(n-w)^3\left(\frac{p+\epsilon}{\epsilon_{\text{tot}}-\epsilon}\right)^2\log(\delta_{\text{tot}}-\delta)^{-1}\right), \tag{5.49}$$

where $\epsilon \in (0, \epsilon_{\text{tot}})$ and $\delta \in (0, \delta_{\text{tot}})$ are free parameters. This gives the final run-time of the protocol ignoring the modifications to the algorithm discussed in Sec. 5.8. Here, we have also ignored the issue of how to choose the parameters s and L instead assuming that a choice within a factor of the optimal choice (see eqs. (5.44), (5.46)) can be made. To aid comparison to competing algorithms, we will now simplify this expression. First, we set $\delta = \delta_{\text{tot}}/2$ as this achieves near minimal run-time for small δ_{tot}. The regime where $\epsilon_{\text{tot}}/p \ll 1$ is interesting and natural to consider because here, the additive error in our estimate is likely to be significantly less than the Born rule probability allowing our estimate to "resolve" the target. Thus, we also assume that $\epsilon_{\text{tot}}/p \ll 1$. This results in the simplifications:

$$\frac{1}{\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2} \approx \frac{4p}{\epsilon^2} \tag{5.50}$$

and also achieves near minimal run-time for $\epsilon = \epsilon_{\text{tot}}/2$. We note that the regime $\epsilon_{\text{tot}}/p \ll 1$ results in the same order runtime as $\epsilon_{\text{tot}}/p \approx 1$ and is slower than the regime of $\epsilon_{\text{tot}}/p \gg 1$ and small p. Ignoring the relatively smaller $\tau_1 \sim O(c+t)$ component and the non-quadratic terms in $2^{\gamma t/2}$ in the expansion of $(2^{\gamma t/2}+p)^2$, these approximations result in a simplified run-time given by:

$$O\left(\frac{2^{\gamma t}p}{\epsilon_{\text{tot}}^2}\left(t+(c+t-h)(n+t)+h(n+t)^2+(n+t)^2(w+t)+(n-w)^3\frac{p^2}{\epsilon_{\text{tot}}^2}\ \left(4+\log\delta_{\text{tot}}^{-1}\right)^2\right)\right). \tag{5.51}$$

In interpreting this run-time, it is helpful to think of factors such as $p/\epsilon_{\text{tot}}^2$ as $(\epsilon_{\text{tot}}/p)^{-1} \times \epsilon_{\text{tot}}^{-1}$. Here, ϵ_{tot}/p can be though of as the relative error in contrast to the additive error ϵ_{tot}. For example, if the run-time for an estimation algorithm scaled purely in the relative error, then the algorithm is in fact multiplicative precision. Thus we see that in the above algorithm is a hybrid of additive and multiplicative precision, depending on both the relative and additive errors. As such, we expect that for a fixed run-time and all else being equal, the accuracy will improve due to a reduction in the target probability but not proportionally to the reduction.

We note that the number of Clifford gates in the input circuit can in practice be quite large. In these cases, the τ_3 component of run-time may become the bottle neck for this algorithm. In Sec. 5.8 we modify this algorithm such that the Clifford evolution component is applied to the final projection instead of the initial state. This modification has two key consequences. First, the Clifford evolution can be computed only once. This is in contrast to the pre-modification algorithm where the Clifford evolution is applied to each of the s samples representing the initial state. Secondly, the number of qubits in the state immediately prior to the application of the fast norm estimation algorithm is changed from $n - w$ to t. Another consequence, related to this point is that in the modified algorithm, the step associated with τ_4 is incorporated into the fast norm estimation component.

5.8 Improvements

We now consider a useful modification to the basic algorithm presented above. This modification involves rewriting the Born rule probability as the expected value of a projector Π with respect to the magic state $|T^t\rangle$. After this initial modification, the algorithm proceeds in a similar fashion to that described above involving each of the key steps including the decomposition of the magic state, sampling stabilizer states based on this decomposition, updating these samples (here, they are updated by the application of the projector Π), constructing a vector average from the updated samples and finally running the fast norm estimation algorithm to estimate the l_2 norm of this average vector.

We now sketch an argument showing that there exists an integer u and a t-qubit projector Π onto a stabilizer code such that $p = 2^{t-u} \langle T^t| \, \Pi \, |T^t\rangle$. From Eq. (5.7) we see that the target probability p can be computed by:

$$\begin{aligned} p &= \langle \psi_U | \left(|x\rangle\langle x| \otimes I^{n-w} \right) |\psi_U\rangle && (5.52) \\ &= 2^t \, \langle 0^n|_c \otimes \langle T^t|_m \, V^\dagger \left(|x\rangle\langle x| \otimes I^{n-w} \otimes \left|0^t\right\rangle\left\langle 0^t\right|_m \right) V \, |0^n\rangle_c \otimes |T^t\rangle_m \, . && (5.53) \end{aligned}$$

The $n + t$ qubit stabilizer projector $\tilde{\Pi} := V^\dagger \left(|x\rangle\langle x| \otimes I^{n-w} \otimes \left|0^t\right\rangle\left\langle 0^t\right|_m \right) V$ can be easily computed using the Gottesman Knill theorem. This involves evolving the set of stabilizers $\{ (-1)^{x_1} Z_1, \ldots, (-1)^{x_w} Z_w, Z_{n+1}, \ldots, Z_{n+t} \}$ by the Clifford circuit V. Further, using now standard techniques [2, 36, 39], this can be contracted to a t qubit projector:

$$2^{-u} \Pi = \langle 0^n|_c \, \tilde{\Pi} \, |0^n\rangle_c \, , \qquad (5.54)$$

where $u \in \mathbb{N}$ and the generating set of Π can be computed in time $O((w+t)(c+t) + (n+t)^3)$. Then, as claimed, the target probability is given by $p = 2^{t-u} \langle T^t| \, \Pi \, |T^t\rangle$. If the projector Π is zero, then $p = 0$ and we are done. Otherwise, Π is a rank R projector onto a stabilizer sub-space where $1 \geq R \geq \min \{ n - w, t \}$.

The remainder of this modified algorithm follows in close analogy to the algorithm we presented earlier. In the remainder of this section, we present each of the key steps and discuss the impact on run-time.

Having computed the projector Π and the integer u, we write p as a square norm,

$$p = 2^{t-u} \langle T^t| \Pi \, \Pi |T^t\rangle \tag{5.55}$$

$$= \left\| 2^{(t-u)/2} \Pi |T^t\rangle \right\|_2^2 \tag{5.56}$$

$$= \left\| |\mu'\rangle \right\|_2^2 \tag{5.57}$$

where we have defined the vector $|\mu'\rangle := 2^{(t-u)/2}\Pi |T^t\rangle$ in analogy to $|\mu\rangle$ from Eq. (5.16). We now proceed to decompose of the magic state into stabilizers as per Eq. (5.14) and then sample from this decomposition similarly to the method from Sec. 5.5. We define the vector $|\Psi'(y)\rangle$ analogous to $|\Psi(y)\rangle$ in Eq. (5.22) as follows:

$$|\Psi'(y)\rangle = 2^{(3t-u)/2} \alpha^{t-|y|} (\alpha^*)^{|y|} \Pi |\tilde{y}\rangle \, . \tag{5.58}$$

Using Eq. (5.14) it is easy to show that the expectation value of $|\Psi'(y)\rangle$ over uniform samples of $y \in \{0,1\}^t$ is equal to $|\mu'\rangle$.

Additionally, by using eqs. (5.54) and (5.26) it is easy to show that the square l_2 norm of $|\Psi'(y)\rangle$ is upper bounded by $2^{\gamma t}$. Thus, as before, we will use the s sample average:

$$\overline{|\Psi'\rangle} = \frac{1}{s} \sum_{i=1}^{s} |\Psi'_i\rangle \tag{5.59}$$

to approximate $|\mu'\rangle$. In the above $|\Psi'_i\rangle$ represents $|\Psi'(y)\rangle$ computed using the i^{th} independent uniform sample of $y \in \{0,1\}^t$. Since the l_2 norm upper bounds for $|\Psi'\rangle$ and $|\Psi\rangle$ are the same, the number of samples needed under this modified algorithm is given by Eq. (5.31) as before. The final step is to apply the fast-norm estimation algorithm to $\overline{|\Psi'\rangle}$ to approximate p as previously. The number of iterations, L, in the fast-norm estimation algorithm is also unchanged and given by Eq. (5.46).

We note that the run-time of the algorithm depends on the number of qubits. In particular the fast norm estimation run-time scales as the cube of the number of qubits. This modification has resulted in a potential change in the number of qubits. In particular, the number of qubits associated with $|\Psi'(y)\rangle$, $\overline{|\Psi'\rangle}$ and $|\mu'\rangle$ is now t compared to $n-w$ for the corresponding states in the original algorithm. This modification also removes the need to evolve the post sampling state through a Clifford circuit saving time. The run-time of the modified algorithm will be:

$$O\left(c + t + (w+t)(c+t) + (n+t)^3 + st^3 L\right) . \tag{5.60}$$

Assuming s and L can be chosen within a constant factor of the minimal values in eqs. (5.31) and (5.46), the run-time becomes:

$$O\left(c + t + (w+t)(c+t) + (n+t)^3 + \frac{2(2^{\gamma t/2} + p)^2 \Gamma'}{\left(\sqrt{p+\epsilon} - \sqrt{p}\right)^2} \log\left(\frac{\delta}{2e^2}\right)^{-1}\right) , \tag{5.61}$$

with,

$$\Gamma' := \left(t^3 \left(\frac{p+\epsilon}{\epsilon_{\text{tot}} - \epsilon} \right)^2 \log(\delta_{\text{tot}} - \delta)^{-1} \right), \tag{5.62}$$

where $\epsilon \in (0, \epsilon_{\text{tot}})$ and $\delta \in (0, \delta_{\text{tot}})$ are free parameters. To simplify this run-time, we set $\delta = \delta_{\text{tot}}/2$ and $\epsilon = \epsilon_{\text{tot}}/2$. We also assume that $\epsilon_{\text{tot}}/p \ll 1$ as previously. Ignoring the component that does not scale exponentially with t and the non-quadratic terms in $2^{\gamma t/2}$ in the expansion of $(2^{\gamma t/2}+p)^2$, this results in a simplified run-time given by:

$$O\left(\frac{2^{\gamma t} p^3}{\epsilon_{\text{tot}}^4} t^3 \ \log^2 \delta_{\text{tot}}^{-1} \right). \tag{5.63}$$

We summarize the main steps in the modified algorithm as follows:

1. We first gadgetize the circuit to produce the $n+t$ qubit Clifford gate-sequence V.
2. We compute $u \in \mathbb{N}$ and the generators of the t qubit projector Π.
3. For each $i \in [s]$, we:
   - uniformly sample $y_i \in \{0,1\}^t$
   - compute $|\Psi'_i\rangle = |\Psi'(y_i)\rangle$ as per Eq. (5.58)
4. Compute the average vector $\overline{|\Psi'\rangle}$ as per Eq. (5.59).
5. Apply the fast norm estimation algorithm using L iterations to produce the estimate $\hat{p}$.

5.9 On the choice of algorithm parameters

We have previously noted that the minimal choice of the number of samples s and the number of iterations L is given by eqs. (5.31) and (5.46). With this minimal choice, our algorithm produces estimates of Born rule probabilities that satisfy Eq. (5.4) and achieves a run-time specified by eqs. (5.61) and (5.62). However, we note that computing the minimal choice of s and L is non-trivial as these depend on the unknown parameter p. This does not present the possibility of the algorithm running forever since the parameters s and L can always be chosen conservatively as the RHS of eqs. (5.31) and (5.46), evaluated at $p = 1$.

In this section we argue that, in the cases when p is small, it is reasonable to expect that we can substantially improve on the conservative choice of $p = 1$. We do this by noting that one can run our algorithm for some initial choice of s and L to produce an estimate $\hat{p}$. This estimate can then be used to guide the choice of s and L in another round of execution. As this is a work in progress, we have yet to fine tune the ideal strategy for this procedure. Nevertheless, in Appendix B, we show that with high probability p can be upper bounded by the following function of the initial choice of s, L and the resulting initial estimate $\hat{p}$:

$$p \leq \hat{p} \frac{1}{1 - \sqrt{b/L}} + \frac{1 + \sqrt{b/L}}{1 - \sqrt{b/L}} \left(2\sqrt{\frac{a}{s}} + \frac{a}{s} \right), \tag{5.64}$$

where $a := 2(2^{\gamma t/2}+1)^2 \log\left(\delta_{\mathrm{tot}}/8e^2\right)^{-1}$ and $b := \log\left(\delta_{\mathrm{tot}}/4\right)^{-1}$. Thus, by making an initial choice of s and L that is sufficiently large relative to a and b respectively, we can with probability greater than $1-\delta_{\mathrm{tot}}/2$ non-trivially upper bound p.

Now, let us consider a small increase on the minimal choices of s and L given by Eqs. (5.44) and (5.46) combined with the choice $\delta = \tilde{\delta} = \delta_{\mathrm{tot}}/4$:

$$s = \frac{2(2^{\gamma t/2}+1)^2}{\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2} \log\left(\delta_{\mathrm{tot}}/8e^2\right)^{-1} \qquad L = \left(\frac{p+\epsilon}{\epsilon_{\mathrm{tot}}-\epsilon}\right)^2 \log\left(\delta_{\mathrm{tot}}/4\right)^{-1}. \tag{5.65}$$

We have now chosen δ and $\tilde{\delta}$ so that the total failure probability from this round sums to $\delta_{\mathrm{tot}}/2$ allowing a non-zero failure tolerance for future rounds. We note that since both s and L in Eq. (5.65) are increasing functions of p, our upper bound can be used to determine a better choice of s and L in the next round.

Our algorithm can be executed in such a way as to allow us to also incorporate the sample data used to produce the initial estimate $\hat{p}$ into future estimates. In doing so, one can introduce selection bias into the estimation procedure but this may be outweighed by the increased sample data.

5.10 Conclusions and outlook

In this chapter, we have motivated the need for a fast additive precision Born rule probability estimation algorithms. We have presented an algorithm that promises to achieve this specific task significantly faster than the BG algorithm.

Our algorithm is a work in progress. The key aspects of this algorithm that still need to be finalized relate to the content of Sec. 5.9. Here, we aim to identify a precise near optimal iterative strategy for computing the choices of the s and L parameter. We aim to analyze the run-time performance of the algorithm and compare this to our predicted performance given in Eq. (5.5). We also aim to incorporate refinements to our algorithm based on the results from Ref. [38]. Finally, we aim to explore the possibility of reducing the number of qubits from t to R, the rank of the projector Π. This will result in a reduced run-time in the fast norm estimation component but it will incur a run-time cost associated with applying a sequence of Clifford gates U_c to the sampled stabilizer states. We hope to mitigate the latter deficiency by developing a technique that allows us to evolve the set of all stabilizer states of the form $|\tilde{y}\rangle$ while treating y as a variable. This can be used to uniformly sample from the post evolution set $\{\, U_c \mid \tilde{y}\rangle \,\}_y$ of states without repeating the evolution separately for each sample.

Chapter 6

Discussion: Part 1

Research into the classical simulation of quantum systems has produced important conceptual leaps in our understanding of quantum computation. These insights have been influential in numerous developments. They have contributed to the birth of quantum computation and the development of early quantum algorithms, as well as ubiquitous tools used in their construction. More recently, these ideas have motivated the study of intermediate models of quantum computation and demonstrations of quantum advantage. Classical algorithms for simulating quantum systems have produced and continue to produced practical solutions to important problems. For example, algorithms for the classical simulation of Clifford circuits [17, 18] and classical simulation of non-interacting fermionic quantum systems [19, 20, 21] have been crucial in solving many research problems. More recently, classical analogs of the HHL quantum matrix inversion algorithm [65] have provided practical solutions to a wide variety of data analysis problems such as finding the optimal investment portfolio given many stocks [66] and providing optimal recommendations based on a user's preferences [32].

In Ref. [1], we presented a method that can be used to construct classical simulation algorithms for estimating Born rule probabilities. Born rule estimation with controllable precision represents a widely applicable notion of classical simulation. Our result has found a number of important applications in: assessing the performance of certain error correcting codes in a more physically realistic setting [33], proving optimality of certain gate synthesis protocols [34] and the use of noisy quantum circuits to simulate ideal ones [35]. In addition, Ref. [1] has inspired a number of important developments in the classical simulation literature [36, 2, 34, 3, 37, 38, 39].

In Part I of this thesis, we substantially extend our earlier work. The general framework we present here provides a flexible mathematical structure that can be combined with a number of free parameter choices (the model) to produce a Born rule probability estimation algorithm. This framework generalizes a number of recent works on simulation including Ref. [1]. By reformulating simulation techniques used in these works into the general framework, we show how the estimation algorithm's performance is influenced by the choice of model. In this thesis, we make progress by refining the mathematical formalism, increasing the scope of the framework and enriching it through examples and applications, as follows.

Our present work substantially relaxes the mathematical structure of the framework, removing unnecessarily restrictive conditions relating to Hermiticity, normalization and duality of frames.

The benefits of this improvement are threefold. Firstly, the increased flexibility of the mathematical formalism permits more freedom to incorporate additional techniques into the simulation algorithm. Secondly, the simpler mathematical structure makes the set of possible algorithms easier to study, making it more amenable to no-go theorems, theorems proving conditional optimality etc. Finally, we note that the removal of these mathematical restrictions has significantly broadened the scope of estimation models; for example our general framework includes the Weyl-Heisenberg frame (i.e. the Pauli frame for qubits).

Compared to Ref. [1], our generalized framework has also provided a significant broadening in scope of application through the inclusion of the pure state formalism. While the techniques presented in Ref. [1] permit the simulation of pure states within the mixed state mathematical formalism; simulation within the pure state formalism can be achieved with a significantly improved run-time. In addition to this practical benefit, our generalization provides a deeper insight into how changes to the state space of a quantum system and its associated norms influence the run-time. We believe this will be important in future work.

The abstract general framework that we present in this thesis is complemented with multiple examples. We explore a range of possibilities by presenting multiple examples of estimation models as well as example applications such as the Clifford plus T algorithm presented in Ch. 5 and the reformulation of other works. We have seen important phenomenological differences that arise from model choice. For example, unlike the discrete Wigner function frame, overcomplete frames such as the stabilizer frames can be optimized to improve performance but the optimization process can be inefficient. Through these examples we have provided an enhanced appreciation for the scope of possibilities within this framework.

The utility of our general framework is twofold. Firstly, it provides an avenue towards the discovery of algorithms that offer conceptual insight and/or practical solutions to important problems. As a tool for the development of novel estimation algorithms, the general framework makes it easier to recognize common structure and generalize patterns in known examples to form testable hypotheses. Secondly; it provides a unifying mathematical formalism that can be used to study the opportunities and limitations of this powerful approach to classically estimating Born rule probabilities.

Our framework provides a vast and, with the exception of a handful of examples, unexplored space of models. There is significant opportunity to better characterize the space of models. For example, there is opportunity to show no-go results for efficient protocols using known results characterizing the isometry groups of l_p spaces. There is also vast potential for developing techniques for sharpening run-times of algorithms fitting in this framework. For example, it is promising to explore generalizations to the run-time reduction phenomena we observed when simulating a system in the pure state formalism instead of the mixed state formalism. Can we more generally use known constraints imposed on the quantum state space to improve run-time? In a similar vein, it is promising to explore these questions in the context of maps between quantum state spaces.

Through the common lens of our general framework, promising research directions are revealed. In the general framework, we have seen three distinct applications using Wigner frames, Weyl-Heisenberg frames and stabilizer frames. From these examples, other prominent possibilities become self-evident. For example, tensor networks with restricted bond dimension offer a promising avenue of investigation. Another possibility is presented in frames generated from a computational basis state under the action of matchgates. Recent work [67] showed that gadgetization and magic state injection in the context of Clifford circuits have analogs in the context of matchgate circuits. The application of our general framework towards simulations based on magic state injected matchgate circuits is an alluring direction of research.

In this thesis, we also presented a classical additive $1/poly$ precision Born rule probability estimation algorithm for Clifford plus T circuits. This algorithm is still a work in progress, but has the potential for being best-in-class for estimation of this important computational model, and a state-of-the-art algorithm for application in the characterization and verification of near term quantum devices.

Part II

From estimation to sampling

Chapter 7

Introduction: From estimation to sampling

Which quantum processes can be efficiently simulated using classical resources is a fundamental and longstanding problem [7, 18, 19, 20, 68, 69]. 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 [13], instantaneous quantum polynomial (IQP) circuits [16, 70], various translationally invariant spin models [71, 72], quantum Fourier sampling [73], one clean qubit (also known as DQC1) circuits [74, 27], chaotic quantum circuits [75] and conjugated Clifford circuits [76] 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 [77] (see also [19, 20, 21] for earlier works in this direction), circuits with positive quasi-probabilistic representations [22, 24, 23], stabilizer circuits supplemented with a small number of T gates [2], stabilizer circuits with small coherent local errors [33], noisy IQP circuits [78], noisy boson sampling circuits [79], low negativity magic state injection in the fault tolerant circuit model [34], quantum circuits with polynomial bounded negativity [1], Abelian-group normalizer circuits [80, 81] and certain circuits with computationally tractable states and sparse output distributions [82]. In addition, there has been some work on using small quantum systems to simulate larger quantum systems [36] as well as using noisy quantum systems to simulate ideal ones [35].

An important motivation for showing efficient classical simulability or hardness thereof for a given (possibly non-universal) quantum computer is understanding what properties of a quantum computer give rise to super-classical computational power. In this context, we desire classical simulability to imply that the computational power of the target quantum computer is "contained in classical", and the hardness of classical simulablility to imply that the target computational device can achieve at least some computational task beyond classical. Achieving these desiderata hinges crucially on the strength of the notion of simulation that is employed. As an extreme example, if one uses a notion of simulation that is too weak, then efficient classical "simulation" of universal quantum circuits may be possible (even if BQP $\not\subseteq$ BPP). In such a case, the existence of a “simulator” does not imply that the computational power of the simulated system is contained within classical. As an opposite extreme, if one uses a notion of simulation that is too strong, then efficient classical “simulation” of even classical circuits may be impossible [83]. In this case, the non-existence of such a simulator does not imply that the computational power of the “un-simulable” system is outside of classical. Once we establish the notion of simulation that is neither “too strong” nor “too weak”, it will become evident that both too strong and too weak notions of simulations have been commonly used in the literature. To this end, we require a clear mathematical statement about which notion of simulation minimally preserves the computational power of the system it simulates.

From a computer science perspective, the computational power of a device can be characterized by the set of problems such a device can solve. However, when it comes to quantum devices that produce probabilistic output from an exponentially growing space, even the question of what problems these devices solve or what constitutes a solution is subtle. Given an efficient description of a quantum circuit, the exact task performed by a quantum computer is to output a sample from the probability distribution associated with the measurement outcomes of that quantum circuit. This suggests that for ideal quantum computers, sampling from the *exact* quantum distribution is what constitutes a solution. On the other hand, it is unclear what well justified necessary requirement fail to be met by an arbitrarily small departure from exact sampling. Perhaps due to these subtleties, the choice of notion of “classical simulation” for sampling problems lacks consensus and, under the umbrella term of *weak simulation*, a number of different definitions have been used in the literature. We will argue that some of these notions are too strong to be minimal and others are too weak to capture computational power. The cornerstone of this argument will be the concept of *efficient indistinguishability*; the ability of one agent to remain indistinguishable from another agent under the scrutiny of any interactive test performed by a computationally powerful referee whilst simultaneously employing resources that are polynomially equivalent.

Examples of definitions that we argue are too strong include simulators required to sample from exactly the target distribution or sample from a distribution that is exponentially close (in L_1-norm) to the target distribution [25]. These also include a notion of simulation based on approximate sampling where the accuracy requirement is the very strong condition that every outcome probability is within a small *relative* error of the target probability [26, 16, 27]. From the perspective of efficient indistinguishibility, these notions of simulation are not minimal since they rule out weaker notions of simulation that are nonetheless efficiently indistinguishable from the target quantum system.

An example of a notion of approximate weak simulation that we argue is too weak requires that the classical algorithm sample from a distribution that is within some small fixed constant L_1-norm of the target distribution [78, 79, 70, 71, 72, 84, 76]. We argue that such a notion does not capture the full computational power of the target, since it cannot perform a task that can be performed by the target device, namely of passing some sufficiently powerful distinguishibility test.

The focus of Part II of this thesis will be on a notion of approximate weak simulation we call EPSILON-simulation. This has been used in prior works including Refs. [13, 85, 73, 2]. We will advocate for this notion of simulation (over other definitions of weak simulation) by showing that an epsilon-simulator of a quantum computer achieves efficient indistinguishablity and any simulator that achieves efficient indistinguishablity satisfies the definition of an epsilon-simulator. Thus epsilon-simulation minimally captures computational power. The notion of epsilon-simulation is also closely related to the definition of a sampling problem from Ref. [85] (where the definition includes an exact statement of what constitutes a solution to the sampling problem). In this language, an epsilon-simulator of a family of quantum circuits can be exactly defined as an efficient classical algorithm which can solve all sampling problems defined by the family of quantum circuits in the natural way. Thus, our result shows that a device can solve all sampling problems defined by a quantum computer if and only if the device is efficiently indistinguishable from the quantum computer.

The conceptual significance of epsilon-simulation as a notion that minimally captures computational power motivates the study of its relation to other notions of simulation. This is particularly important for translating the existing results on simulability and hardness into statements about computational power relative to classical. Such a comparison to the above-mentioned approximate weak simulators is clear but a comparison to simulators defined in terms of Born probability estimation can be significantly more involved. Simulators which output sufficiently accurate Born probability estimates can be called as subroutines in an efficient classical procedure in order to output samples from a desired target distribution. Such a procedure can be used to "lift" these simulators to an epsilon-simulator implying that the computational power of all families of quantum circuits simulable in this way is contained within classical.

Some commonly used notions of simulation such as *strong simulation* and multiplicative precision simulation require the ability to estimate Born probabilities extremely accurately. These simulators can be lifted to epsilon-simulators [19, 20, 26]. We focus on another notion of simulation that has been prominent in recent literature [1, 34, 33] which we call a *poly-box* . Compared to strong or multiplicative precision simulators, a poly-box has a much less stringent requirement on the accuracy of Born probability estimates that it produces. We discuss the significant conceptual importance of poly-boxes owing to the fact that they capture the computational power with respect to decision problems while simultaneously being weak enough to be admitted by IQP circuits, unconditioned magic-state injected Clifford circuits and possibly other intermediate models for quantum computation.

Assuming some complexity theoretic conjectures, we show that a poly-box is a strictly weaker notion of simulation than epsilon-simulation. However, if we impose a particular sparsity restriction on the target family of quantum circuits, then we show that a poly-box can be lifted to an epsilon-simulator, implying that the two notions are, up to efficient classical computation, equivalent under this sparsity restriction.

7.1 Outline of our main results

7.1.1 Indistinguishability and ϵ-simulation.

In Sec. 8, we motivate the use of a particular notion of efficient simulation, which we call efficient polynomially small in L_1-norm simulation (epsilon-simulation or ϵ-simulation for short). Essentially, we say that an algorithm can ϵ-*simulate* a family of quantum circuits, if for any $\epsilon > 0$, it can sample from a distribution that is ϵ-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 ϵ-simulator" for a family of circuits is equivalent to demanding that even a computationally omnipotent referee cannot distinguish the simulator's outputs from that of the target circuit family. Further, any simulator that satisfies efficient distinguishability also satisfies the definition of an ϵ-simulator. This is captured by the following theorem presented in Sec. 8.

Result 1. *Bob has an ϵ-simulator of Alice's quantum computer if and only if given the hypothesis testing scenario considered in Sec. 8.3 there exists a strategy for Bob which jointly achieves indistinguishability and efficiency.*

7.1.2 Efficient outcome estimation: the poly-box.

A family of binary outcome quantum circuits, where each circuit is indexed by a bit-string, defines a decision problem as follows: Given a bit-string indexing a quantum circuit, decide which of the circuit's two possible outcomes is more likely[1].

Here, the only quantity relevant to the computation is the probability associated with the binary measurement outcome (decision). Hence, in this setting, simulation can be defined in terms of the accuracy to which these probabilities can be estimated. A commonly used notion of simulation known as *strong simulation* requires the ability to estimate Born probabilities extremely accurately. In Sec. 4.1.5, we defined a much weaker notion of simulation (*poly-box*[2]) which is a device that computes an additive polynomial precisions estimate of the quantum probability (or marginal probability) associated with a specific outcome of a quantum circuit.

We show that families of quantum circuits must admit a poly-box in order to be ϵ-simulable.

Result 2. *If $\mathcal{C}$ is a family of quantum circuits that does not admit a poly-box algorithm, then $\mathcal{C}$ is not ϵ-simulable.*

We advocate the importance of this notion on the grounds that whether or not some given family of quantum circuits admits a poly-box informs our knowledge of the computational power of that family relative to classical. In particular:

- if a (possibly non-universal) quantum computer can be efficiently classically simulated in the sense of a poly-box, then such a quantum computer cannot solve decision problems outside of classical
- if a (possibly non-universal) quantum computer cannot be efficiently classically simulated in the sense of a poly-box, then such a quantum computer can solve a sampling problem outside of classical (Thm. 7)

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. [1],

[1] Technically, one is also promised that the given bit-string will only ever index a circuit where the probability of the two outcomes are bounded away from 50%.

[2] This notion is similar to a notion introduced by Ref. [86] where it was (using a terminology inconsistent with the present paper) referred to as weak simulation.

where the it was found that the efficiency of this estimator depends on the amount of "negativity" in the quasiprobability description of the quantum circuit. As a second example, we consider the family of circuits $\mathcal{C}_{\mathrm{PROD}}$, for which the n-qubit input state ρ 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 poly-box for $\mathcal{C}_{\mathrm{PROD}}$ in Sec. 9. As a third example, we also outline a construction of a poly-box for Instantaneous Quantum Polynomial-time (IQP) circuits $\mathcal{C}_{\mathrm{IQP}}$ based on the work of Ref. [87].

7.1.3 From estimation to simulation.

For the case of very high precision probability estimation algorithms, prior work has addressed the question of how to efficiently lift these to algorithms for high precision approximate weak simulators. In particular, Refs. [19, 20, 26] (see also Appendix C) lift estimation algorithms with small *relative* error. In Appendix C, we also present a potentially useful algorithm for lifting small additive error estimators. In Sec. 10 we focus on the task of lifting an algorithm for a poly-box, to an ϵ-simulator. Since a poly-box is a much less precise probability estimation algorithm (in comparison to strong simulation), achieving this task in the general case is implausible (see Sec. 11). In Sec. 10, we will show that a poly-box can efficiently be lifted to an ϵ-simulator if we restrict the family of quantum distributions to those possessing a property we call *poly-sparsity.* This sparsity property measures "peakedness versus uniformness" of 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 formalize this notion in Sec. 10.

Result 3. *Let $\mathcal{C}$ be a family of quantum circuits with a corresponding family of probability distributions $\mathbb{P}$. Suppose there exists a poly-box over $\mathcal{C}$, and that $\mathbb{P}$ is poly-sparse. Then, there exists an ϵ-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. [33], are ϵ-simulable.

7.1.4 Hardness results.

Finally, in Sec. 11, we prove that the poly-box requirements of Result 3 is on its own not sufficient for ϵ-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 ϵ-simulation is impossible.

We prove that the family $\mathcal{C}_{\mathrm{PROD}}$ described above, which violates the poly-sparsity requirement, admits a poly-box. Then, by assuming a now commonly used "average case hardness" conjecture [13, 70, 71, 73, 84, 76], we show that the ability to perform ϵ-simulation of $\mathcal{C}_{\mathrm{PROD}}$ implies the unlikely result that the polynomial hierarchy collapses to the third level. Loosely, this result suggests that there exist quantum circuits where the probability of any individual outcome (and marginals) can be efficiently estimated, but the system cannot be ϵ-simulated. Our hardness result closely follows the structure of several similar results, and in particular that of the IQP circuits result of Ref. [70].

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 can be shown by applying the result of Refs. [88, 57, 46, 89, 45] and follows from an analogous argument to Thm. 5.1 of Ref. [76].

Result 4. *If there exists an ϵ-simulator of $\mathcal{C}_{\text{PROD}}$ and Conjecture 2 holds, then the polynomial hierarchy collapses to the third level.*

We note that our hardness result is implied by the hardness results presented in Refs. [71, 72], however; our proof is able to use a more plausible average case hardness conjecture than these references due to the fact that we are proving hardness of ϵ-simulation rather than proving the hardness of the yet weaker notion of approximate weak simulation employed by these references.

In Appendix E we also present Thm. 18. This theorem shows that the properties of poly-sparsity and anti-concentration are mutually exclusive.

The flow chart in Fig. 7.1 summarizes the main results in Part II of this thesis by categorizing any given family of quantum circuits in terms of its computation power based on whether or not the circuit family admits certain properties related to simulability.

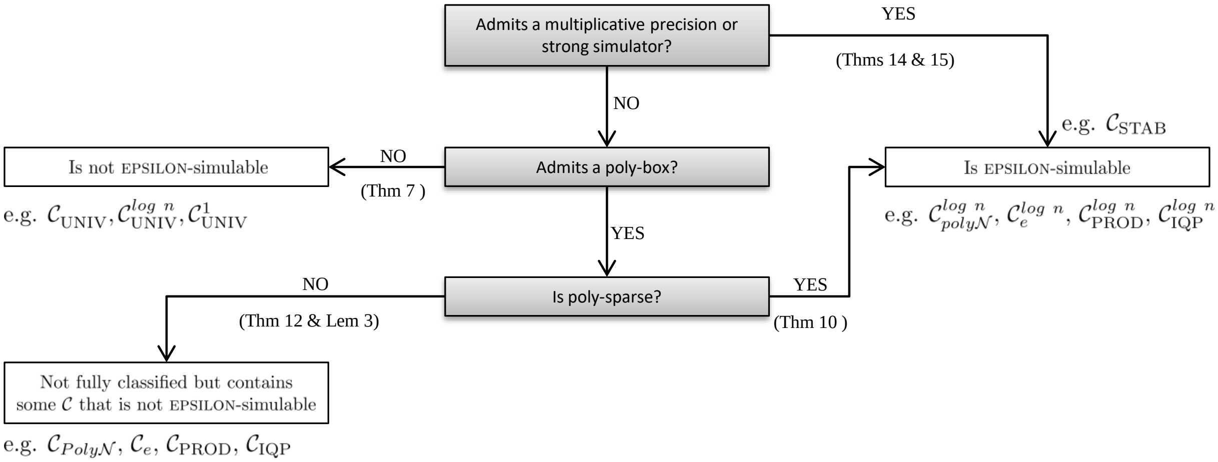
Admits a multiplicative precision or strong simulator?
YES
(Thms 14 & 15)
NO
e.g. $\mathcal{C}_{\text{STAB}}$
Is not EPSILON-simulable
NO
Admits a poly-box?
Is EPSILON-simulable
(Thm 7)
e.g. $\mathcal{C}_{\text{UNIV}}, \mathcal{C}_{\text{UNIV}}^{log\ n}, \mathcal{C}_{\text{UNIV}}^{1}$
e.g. $\mathcal{C}_{poly\mathcal{N}}^{log\ n}, \mathcal{C}_{e}^{log\ n}, \mathcal{C}_{\text{PROD}}^{log\ n}, \mathcal{C}_{\text{IQP}}^{log\ n}$
YES
NO
Is poly-sparse?
YES
(Thm 12 & Lem 3)
(Thm 10)
Not fully classified but contains some $\mathcal{C}$ that is not EPSILON-simulable
e.g. $\mathcal{C}_{Poly\mathcal{N}}, \mathcal{C}_{e}, \mathcal{C}_{\text{PROD}}, \mathcal{C}_{\text{IQP}}$

Figure 7.1: An overview of the main results. An arbitrary family of quantum circuits $\mathcal{C}$ is partially classified by its computational power relative to universal classical computers. The unclassified category (admits a poly-box and is not poly-sparse) is known to contain circuit families that are hard to EPSILON-simulate assuming some plausible complexity theoretic conjectures. We give examples of circuits families in these categories. Here, $\mathcal{C}_{\text{UNIV}}^{*}, \mathcal{C}_{\text{STAB}}^{*}, \mathcal{C}_{poly\mathcal{N}}^{*}$ and $\mathcal{C}_{\text{IQP}}^{*}$ refer to the following families of circuits: universal circuits, stabilizer circuits, circuits with polynomially bounded negativity and IQP circuits respectively. The circuit families $\mathcal{C}_e$ and $\mathcal{C}_{\text{PROD}}$ are discussed in some detail in Secs. 10.1 and 4.3.5 respectively. The presence of a superscript represents an upper bound on the number of qubits to be measured.

Chapter 8

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 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 [13, 70]. As another example, Ref. [27] 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) with a probability 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. [1, 33] 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 discuss a meaningful notion of approximate weak simulation, which we call ϵ-simulation. This notion of simulation is a natural mathematical relaxation of exact weak simulation and has been used in prior works, e.g., in Refs. [13, 73, 2]. Further, this notion of simulation is closely related to the class of problems in complexity theory known as sampling problems [85]. Here, we define ϵ-simulation and prove that up to polynomial equivalence, an ϵ-simulator of a quantum computer is effectively a perfect substitute for any task that can be performed by the quantum computer itself. In particular, we will show that ϵ-simulators satisfy *efficient indistinguishability* meaning that they can remain statistically indistinguishable from (according to a computationally unbounded referee) and have a polynomially equivalent runtime to the quantum computer that they simulate. We argue that efficient indistinguishability is a natural choice of a rigorously defined global condition which minimally captures the concept of computational power. The accuracy requirements of ϵ-simulation are rigorously defined at the local level of each circuit and correspond to solving a sampling problem (as defined in [85]) based on the outcome distribution of the circuit. Thus our result shows that the ability to solve all sampling problems solvable by a quantum computer $\mathcal{C}$ is a necessary and sufficient condition to being efficiently indistinguishable from $\mathcal{C}$ or "computationally as powerful as $\mathcal{C}$".

8.1 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. 8.1 below depicts a quantum circuit. The output of running this circuit is a classical random variable $X = (X_1, \ldots, X_k)$ that is distributed according to the quantum probability distribution.

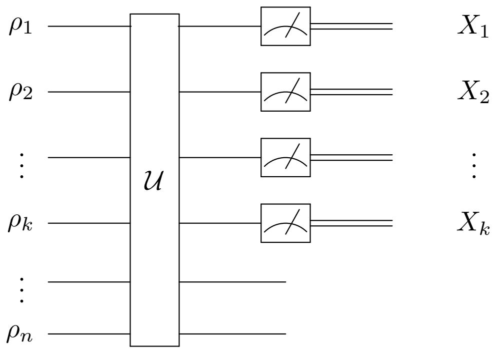

Figure 8.1: An example of a quantum circuit. This circuit acts on n qubits (or, in general, qudits). The initial state is a product state. The unitary operation $\mathcal{U}$ must be constructed out of a sequence of local unitary gates. The first k qubits in this example are each measured in a fixed basis, yielding outcome $(X_1, X_2, \ldots, X_k)$. 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.

The term *weak simulation* has also been used in reference to classical algorithms which sample from distributions which *approximate* the target probability distribution. There exist at least four distinct notions of approximate weak simulation. As background, we give a brief description of these here although the focus of this paper will be on only one of these and will be discussed in some detail later in this section.

1. The strongest[1] of these requires that the classical algorithm sample from a distribution

[1] We note that the relative strength of these notions of approximate weak simulation depends on the specifics

that is exponentially close (in L_1-norm) to the target distribution. This notion was used in Ref. [25, 83].

2. A generally weaker notion requires that the sampled distribution be sufficiently close to the target distribution so as to ensure that for every outcome x, the sampled distribution satisfies $|P_{sampled}(x) - P_{target}(x)| \leq \epsilon P_{target}(x)$ for some fixed $\epsilon > 0$. See Ref. [26] and also [16, 27] for related variants.

3. A yet weaker notion of approximate weak simulation requires that the classical algorithm sample from a distribution that is inverse polynomially close (in L_1-norm) to the target distribution. This notion of simulation has been used in prior works, e.g., in Refs. [13, 85, 73, 2] both in the context of hardness of classical simulation and existence of classical simulators. We call this ϵ-simulation.

4. The final prominent example of approximate weak simulation, the weakest of all four approximate notions, requires that the classical algorithm sample from a distribution that is within some small fixed constant L_1-norm of the target distribution. This definition has predominantly featured in hardness proofs [70, 71, 72, 84, 76]. It has also feature in proofs of efficient classical simulability of noisy boson sampling circuits [79] and noisy IQP circuits [78].

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[2] 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. 8.1 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).

Ref. [83] provides a compelling argument advocating for the use of weak simulation in place of strong simulation by showing that there exist classically efficiently weak simulable probability distributions that are #P-hard to strong simulate, thus showing that aiming to classically strong simulate is an unnecessarily challenging goal. In a similar vein, here we will advocate for the notion of ϵ-simulation over other notions of simulation including the alternative notions of approximate weak simulation.

of the target probability distributions. Our ordering of strongest to weakest is only a rough guide.

[2] A distribution $P(x)$ over bit-strings $x \in \{0,1\}^n$ is said to have a marginal distribution $P_{\{i_1,\ldots,i_m\}}(\tilde{x}) = \sum_{x_{i_1}} \cdots \sum_{x_{i_m}} P(x)$ (marginalized over the bits $\{i_1,\ldots,i_m\}$ where $\tilde{x} \in \{0,1\}^{n-m}$ is given by modifying the vector x by removing the entries $\{i_1,\ldots,i_m\}$.

8.2 ϵ-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. [25, 83, 13, 85, 73, 2, 70, 71, 84, 76, 79, 78] . Here, we define the notion of ϵ-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 ϵ 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 ϵ-sampling of a probability distribution $\mathcal{P}$ as follows:

Definition 3. *Let $\mathcal{P}$ be a discrete probability distribution. We say that a classical device or algorithm can* ϵ-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-distance upper bounds on the one-shot success probability of distinguishing between two distributions. More details can be found in the proof of Thm. 6 in Sec. 8.4.

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 ϵ 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 ϵ in order to output a sample, hence the efficiency requirement ensures that these resources scale at most polynomially in the precision $1/\epsilon$.

Definition 4. *We say that a classical device or algorithm can ϵ-simulate a quantum circuit if it can ϵ-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 ϵ) 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[3]. As an example, consider a family of circuits described by a mapping from $\mathcal{A}^*$ (finite strings over some finite 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[4] 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

[3] 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 $poly(n)$.

[4] 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.

the case of ϵ-simulation, we require that the simulator be able to sample a distribution within ϵ distance of the quantum distribution efficiently in both n and $1/\epsilon$.

Definition 5. *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 $O(poly(n, \frac{1}{\epsilon}))$.*

8.3 ϵ-simulation and efficient indistinguishability

As noted earlier, this definition ensures that ϵ-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 ϵ-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 ϵ-simulation requires a polynomial scaling between the precision $(1/\epsilon)$ of the approximate sampling and the time taken to produce a sample. Below (Thm. 6), 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 ϵ-simulator is to be decided). Further, suppose that a referee with unbounded computational power and with full knowledge of the specifications of $\mathcal{C}$, 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 8.2). 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 $P_{correct} \geq \frac{1}{2} + \delta$ where $P_{correct}$ is the probability of deciding correctly. Bob's* *goal is to come up with a (δ-dependent) strategy for responding to the referee's requests such that it jointly achieves:*

- indistinguishablity: *for any $\delta > 0$ and for any test that the referee applies, $P_{correct} < \frac{1}{2} + \delta$ and*
- efficiency: *for every choice of circuit request sequence α, Bob must be able to execute his strategy using resources which are $O(poly(N(\alpha), \frac{1}{\delta}))$ 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 $P_{correct} = \frac{1}{2}$ simply by randomly guessing H_a or H_b. Importantly, the referee has complete control over the number of rounds in the test and additionally does not have any upper bound imposed on the number of rounds. Hence, $P_{correct}$ is the ultimate one shot probability of the referee correctly deciding between H_a or H_b and in no sense can this probability be amplified through more rounds of information requests. As such, we will say that the referee achieves distinguishability between Alice and Bob if $\forall\delta > 0$, there exists a test that the referee can apply ensuring that $P_{correct} \geq 1-\delta$ (independent of Bob's strategy). Alternatively, we will say that Bob achieves indistinguishability (from Alice) if $\forall\delta > 0$, there exists a response strategy for Bob such that $P_{correct} \leq \frac{1}{2} + \delta$ (independent of what test the referee can apply). We will show that if Bob has an ϵ-simulator then there exists a strategy for Bob such that he jointly achieves indistinguishablity (i.e. the referee cannot improve on a random guess by any fixed probability $\delta > 0$) and efficiency. In this case, Bob can at the outset choose any $\delta > 0$ and ensure that $P_{correct} < \frac{1}{2} + \delta$ for all strategies the referee can employ.

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 $c_1, \ldots, c_m \in \mathcal{C}$. Thus we define the resource cost for Alice to meet this request by $N := n_1 + \ldots + n_m$.

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 ϵ. Thus, Bob's strategy is defined by the rules by which he chooses ϵ_j, the accuracy parameter for his response in the j^{th} round[5]. Thus, for a given sequence of circuit requests $a_1, \ldots, a_m \in \mathcal{A}^*$, Bob will incur a resource cost $T = t_1 + \ldots + t_m$ where $t_j \sim poly(n_{a_j}, 1/\epsilon_j)$ is Bob's resource in the j^{th} round. Thus the efficiency condition requires that there exists some polynomial $f(x, y)$ such that for all $\delta > 0$ and for all

[5] 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.

possible request sequences $\alpha = (a_1, \ldots, a_m)$, $T(\alpha) \leq f(N(\alpha), \frac{1}{\delta})$. 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 6. *Bob has an ϵ-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 indistinguishablity and efficiency.*

The proof for this theorem can be found in Sec. 8.4. The proof uses the fact that the L_1 distance between Alice and Bob's output distributions over the entire interactive protocol can be used to upper bound the probability of correctly deciding between H_a and H_b. Further, we show that the total L_1 distance between Alice and Bob's output distributions over the entire interactive protocol grows at most additively in the L_1 distance of each round of the protocol. We also note that an ϵ-simulator allows Bob to ensure that the L_1 distance of each round decays like an inverse quadratic ensuring that the sum of the L_1 distances converges to the desired upper bound. The convergence of the inverse quadratic series, which is an inverse polynomial, thus motivates the significance of ϵ-simulators i.e. simulators with run-time $O(poly(n, 1/\epsilon))$.

We note that the "if" component of the theorem says that meeting the definition of ϵ-simulator is necessary for achieving efficient indistinguishability, thus the notion of simulation cannot be weakened any further without compromising efficient indistinguishability.

Throughout this paper we view a quantum computer as a uniform family of quantum circuits $\mathcal{C} = \{\, c_a \mid a \in \mathcal{A}^* \,\}$. We note that by committing to the circuit model of quantum computation, our language including important definitions such as ϵ-simulation are not necessarily well suited to other models of computation unless these are first translated to the circuit model. For example, in a quantum computational model that makes use of intermediate measurements, such as the measurement based quantum computing (MBQC) model, consider a procedure where a part of the state is measured then conditioned on the outcome, a second measurement is conducted. This procedure (consisting of 2 rounds of measurement) can be described as a single circuit in the circuit model, but cannot be broken up into two rounds involving two separate circuits. This limitation becomes apparent when we consider the hypothesis testing scenario. If the referee is performing a multi-round query, expecting the candidate to possess an MBQC-based quantum computer, then even Alice with a quantum computer may be unable to pass the test unless her computer operates in an architecture that can maintain quantum coherence between rounds. In the setting we consider, such a query by the referee in not allowed.

8.4 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:

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

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, α^* and β^* 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:

$$\begin{aligned} P_{correct} &= 1 - \Pr(X \in A_2 \mid X \sim \mathcal{P}_1) \Pr(X \sim \mathcal{P}_1) - \Pr(X \in A_1 \mid X \sim \mathcal{P}_2) \Pr(X \sim \mathcal{P}_2) \\ &= 1 - \alpha^* \Pr(H_1) - \beta^* \Pr(H_2) \\ &= \frac{1}{2} + \frac{||\mathcal{P}_1 - \mathcal{P}_2||_1}{4} \,. \end{aligned} \tag{8.1}$$

The interactive protocol between the referee and the candidate will proceed as follows (see Figure 8.2):

1. Initially, the referee will fix a test by choosing a function $a(\cdot)$ 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 baring 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 $j = 1, 2, \ldots$, 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 $m \in \mathbb{N}$.

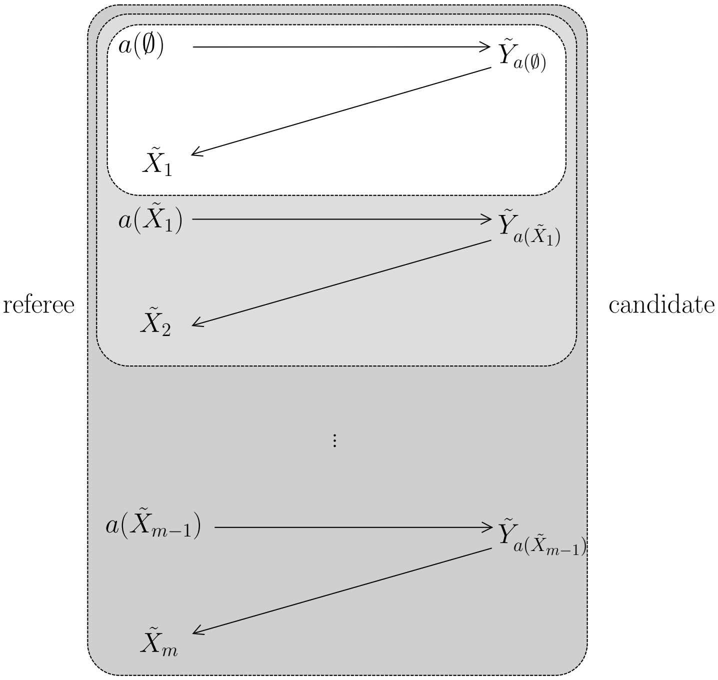

Figure 8.2: The figure above shows the interactive protocol between the referee and the candidate. In each round of the protocol, the referee send a circuit description to the candidate. This circuit description is in general given by applying any fixed (possibly stochastic) map to all of the prior data collected by the referee. The candidate's 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.

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. $\tilde{X}, \tilde{Y} \to X, Y$) and alternatively when the candidate is fixed to be Bob, we will replace the tilde with a prime (i.e. $\tilde{X}, \tilde{Y} \to X', Y'$).

The set of all possible data collected by the referee (based on all probabilistic choices including the choice of the candidate) over the course of the entire test can be viewed as a tree where each branch corresponds to a distinct observed value of the random variable $\tilde{X}_m$. The two probability distribution $\mathcal{P}_1$ and $\mathcal{P}_2$ discussed above will each correspond to a distribution over all of the branches of this tree conditioned on the choice of candidate. We note that one can easily incorporate probabilistic choices by the referee into the formalism which will only result in an increase in the number of branches of the tree. Equation (8.1) shows that Bob can ensure suppression of $P_{correct}$ by suppressing the L_1 distance $||\mathcal{P}_1 - \mathcal{P}_2||_1$. However, if Bob has an ϵ-simulator, he can only directly control the L_1 distance between his output and that of Alice for a given circuit request. The proof below culminating in Eq. (8.5) demonstrates that $||\mathcal{P}_1 - \mathcal{P}_2||_1$ is sub-additive in the L_1 distance of each circuit request thus Bob can upper bound $P_{correct}$ by bounding each round's L_1 distance in such a way as to ensure the sum converges to the desired bound for $||\mathcal{P}_1 - \mathcal{P}_2||_1$.

Proof. We now prove each direction of the "if and only if" statement of Thm. 6.
"⇒" Here, we assume that Bob's simulation scheme is an ϵ-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 ϵ-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 as outlined above is fixed and independent of the referee's hypothesis test. Further, we note that for all $m \in \mathbb{N}$:

$$\begin{aligned} \sum_{j=1}^{m} \epsilon_j &\leq \sum_{j=1}^{\infty} \epsilon_j \\ &= 4\delta \end{aligned} \tag{8.2}$$

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. (8.2) upper bounds $\mathcal{E}[X_m, X'_m]$. Hence:

$$\mathcal{E}[X_m, X'_m] \leq 4\delta \tag{8.3}$$

$$\begin{aligned} \mathcal{E}[X_{j+1}, X'_{j+1}] &= \sum_{\alpha,\beta} \Big|\Pr(Y_{a(X_j)} = \beta | X_j = \alpha) \Pr(X_j = \alpha) - \Pr(Y'_{a(X'_j)} = \beta | X'_j = \alpha) \Pr(X'_j = \alpha)\Big| \\ &= \sum_{\alpha,\beta} | \Pr(Y_{a(X_j)} = \beta | X_j = \alpha) \Pr(X_j = \alpha) - \Pr(Y_{a(X_j)} = \beta | X_j = \alpha) \Pr(X'_j = \alpha) \\ &\qquad\qquad + \Pr(Y_{a(X_j)} = \beta | X_j = \alpha) \Pr(X'_j = \alpha) - \Pr(Y'_{a(X'_j)} = \beta | X'_j = \alpha) \Pr(X'_j = \alpha)| \\ &\leq \sum_{\alpha,\beta} \Pr(Y_{a(X_j)} = \beta | X_j = \alpha) \big|\Pr(X = \alpha) - Pr(X' = \alpha)\big| \\ &\qquad + \sum_{\alpha,\beta} \Pr(X'_j = \alpha) \Big|\Pr(Y_{a(X_j)} = \beta | X_j = \alpha) - \Pr(Y'_{a(X'_j)} = \beta | X'_j = \alpha)\Big| \\ &\leq \mathcal{E}[X_j, X'_j] + \sum_{\alpha} \mathcal{E}[Y_{a(\alpha)}, Y'_{a(\alpha)}] \Pr(X'_j = \alpha) \end{aligned} \tag{8.4}$$

where the sums are taken over α in the support of $\tilde{X}_j$ and β in $\underset{a \in \mathcal{A}^*}{\cup} supp(\tilde{Y}_a)$.

We note that the precision of Bob's response in any round only depends on the round number. Thus, Eq. (8.4) can be simplified by replacing $\mathcal{E}[Y_{a(\alpha)}, Y'_{a(\alpha)}]$ with the upper bound ϵ_{j+1}. Combined with the observation that $\mathcal{E}[X_1, X'_1] = \epsilon_1$, we have shown that:

$$\begin{aligned}\mathcal{E}[X_m, X'_m] &\leq \sum_{j=1}^{m} \epsilon_j \\ &\leq 4\delta\end{aligned} \tag{8.5}$$

This proves that Bob's strategy meets the indistinguishibility property. We now consider the efficiency of the strategy. We recall that given a circuit request sequence α, Alice's and Bob's resource costs are represented by $N(\alpha)$ and $T(\alpha)$ respectively. Further, Alice's resource costs is lower bounded by m, the number of rounds of the Hypothesis test α.

By definition of ϵ-simulation, there exists $\kappa, c_1, c_2 \in \mathbb{N}$ such that for a given circuit index a, and precision ϵ, $T(a) \leq c_1 \left(\frac{N(a)}{\epsilon}\right)^{\kappa} + c_2$. 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 $\alpha = (a_1, \ldots, a_m)$, with $m \geq 2$ we have:

$$T(\alpha) = \sum_{j=1}^{m} T(a_j) \tag{8.6}$$

$$\leq \sum_{j=1}^{m} \left(\frac{N(a_j)}{\epsilon_j}\right)^{\kappa} \tag{8.7}$$

$$= \sum_{j=1}^{m} \left(\frac{\pi^2 j^2 N(a_j)}{24\delta}\right)^{\kappa} \tag{8.8}$$

$$\leq \left(\frac{\pi^2}{24\delta}\right)^{\kappa} \left[\sum_{j=1}^{m-1} j^{2\kappa} + m^{2\kappa}[N(\alpha) - (m-1)]^{\kappa}\right] \tag{8.9}$$

$$\leq \left(\frac{\pi^2}{24\delta}\right)^{\kappa} \left[\left(\frac{m - 0.5}{2\kappa + 1}\right)^{2\kappa+1} + m^{2\kappa} N(\alpha)^{\kappa} - m^{2\kappa}(m-1)^{\kappa}\right] \tag{8.10}$$

$$\leq \left(\frac{\pi^2 m^2 N(\alpha)}{24\delta}\right)^{\kappa} \tag{8.11}$$

$$\leq \left(\frac{\pi^2 N(\alpha)^3}{24\delta}\right)^{\kappa} \tag{8.12}$$

$$\in O\left(poly(N(\alpha), \frac{1}{\delta})\right) \tag{8.13}$$

where:

- in Eq. (8.9) we have used the fact that $N(a_j) \geq 1$ for all j hence the expression is maximized when α is chosen such that $N(a_j) = 1$ for $j = 1, \ldots, m-1$ and $N(a_m) = N(a_1) + \ldots + N(a_m) - (m-1)$;
- in Eq. (8.10) 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. (8.10) 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. (8.11) 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 α 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 $\in O\left(poly(N(a), \frac{1}{\delta})\right) \subseteq O\left(poly(n, \frac{1}{\epsilon})\right)$. □

8.5 ϵ-simulation and computational power

In addition to the technical contribution of Thm. 6, we wish to make an argument for the conceptual connection between computational power and efficient indistinguishability. Intuitively, we wish to say that an agent A is at least as computationally powerful as agent B if A can "do" every task that B can do using an equivalent amount of resources. In our setting, we can restrict ourselves to polynomially equivalent resources and the most general task of sampling from a target probability distribution given an efficient description of it. However, defining what constitutes an acceptable solution to the sampling task is not only of central importance but also difficult to conceptually motivate. Given a description of a probability distribution, can anything short of sampling *exactly* from the specified distribution constitute success? An answer in the negative seems unsatisfactory because very small deviations[6] from exact sampling are ruled out. However, an answer in the positive presents the subtlety of specifying the exact requirement for achieving the task. It is easy to offer mathematically reasonable requirements for what constitutes success at the local level of each task but significantly more difficult to conceptually justify these as precisely the right notion. In our view, this difficulty arises because a well formed conceptually motivated requirement at the local level of each task must be inherited from a global requirement imposed at the level of the agent across their performance on any possible task.

We advocate for efficient indistinguishability as the right choice of global requirement for defining computational power and implicitly defining what constitutes a solution to a sampling task. If an agent is efficiently indistinguishable from another then, for any choice of $\delta > 0$ chosen at the outset, the referee cannot assign any computational task to the candidate to observe a consequence that will improve (over randomly guessing) their ability to correctly decide between H_a and H_b by a probability δ. Thus, there is no observable consequence[7] to substituting an agent

[6] For example consider the scenario that whenever an agent is asked to sample from some distribution $\mathcal{P}$, they output samples from exactly $\mathcal{P}$ every time, possibly with one exception. In particular, a memory bit stores if the exception has ever taken place. If it has occurred, then forever more, when the agent is asked to sample from $\mathcal{P}$, an exact sample is produced. If the exception has not yet taken place then with some very small probability, the agent will output the most likely outcome instead of an exact sample from $\mathcal{P}$.

[7] Since the observer is the computationally unbounded referee, then any event is an observable consequence i.e.

with another efficiently indistinguishable agent. For these reasons, we argue that in the setting where the agents are being used as computational resources, an agent's ability to (efficiently and indistinguishably) substitute another naturally defines containment of computational power. In light of this, the "only if" component of Thm. 6 says that, the computational power of Bob (given an ϵ-simulator of $\mathcal{C}$) contains that of Alice (given $\mathcal{C}$) and the "if" component says that an ϵ-simulator is the *minimal* simulator that achieves this since any simulator to achieve efficient indistinguishibility is an ϵ-simulator.

The referee can be seen as a mathematical tool for bounding the adversarial ability of any natural process to distinguish an agent from an efficiently indistinguishable substitute. As such one may argue for further generalization of the concept of efficient indistinguishability from one which is defined with respect to (w.r.t.) a computationally unbounded referee to a notion dependent on the computational power of the referee. If we take the view that the computational power of all agents within this universe is bounded by universal quantum computation, then a particularly interest generalization is efficiently indistinguishability w.r.t. a referee limited to universal quantum computation. We return to this generalization in the discussion, elsewhere focusing on efficient indistinguishability w.r.t. a computationally unbounded referee.

if we let S be the set of all possible responses across all rounds from both Alice and Bob, then any element of the power set of S is an observable consequence.

Chapter 9

Probability Estimation

As described in the previous section, an exact (or approximate in the sense of Ref. [25]) weak simulator produces outcomes sampled from the exact (or exponentially close to the exact) Born rule probability distribution associated with a quantum circuit. The notion of ϵ-simulation is a weaker notion of simulation, a fact we aim to exploit by constructing algorithms for ϵ-simulation that would not satisfy the above-mentioned stronger notions of simulation. In Ch. 10, we describe an approach to ϵ-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 Part I of this thesis, we focussed on this first component presenting a general framework for the construction of Born rule probability estimation algorithms. In Ch. 5 we also discussed an additive $1/poly$ precision Born rule probability estimation algorithms that is not a poly-box because its run-time is exponential in the number of T gates. In this chapter we will focus specifically on poly-boxes. First we will discuss the conceptual significance of a poly-box from the perspective of computational power. Then we will present a poly-box over the family of Instantaneous Quantum Polynomial-time (IQP) quantum circuits [15, 90, 16, 70].

9.1 Conceptual significance of a poly-box

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 a useful concept. We first note that the existence of a poly-box is a necessary condition for ϵ-simulation.

Theorem 7. *If $\mathcal{C}$ is a family of quantum circuits that does not admit a poly-box algorithm, then $\mathcal{C}$ is not ϵ-simulable.*

Proof. We note that given an ϵ-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 ϵ-simulator outputs outcomes in S and using this observed frequency as the estimator for $\mathcal{P}(S)$. □

A poly-box over $\mathcal{C}$ is not only necessary for the existence of an ϵ-simulator over $\mathcal{C}$, but as we will show in Thm. 10, 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 ϵ-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 (ϵ, δ)-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 an alternative easily computable choice of a' such that the probability that running the circuit $c_{a'} \in \mathcal{C}_{\text{UNIV}}$ results in an outcome in $(1, \bullet, \dots, \bullet)$ is identical to the probability that running the circuit $c_a \in \mathcal{C}_{\text{UNIV}}$ results in an outcome in S. However, since we are considering the general case of not necessarily universal families of quantum circuits, it is feasible that a poly-box over $\mathcal{C}$ will be computationally more powerful than a poly-box over $\mathcal{C}$ restricted to only estimating probabilities of events of the form $S = (1, \bullet, \dots, \bullet)$. On the other hand, we do not wish to make poly-boxes exceedingly powerful. If we view a poly-box over $\mathcal{C}$ as a black box containing an agent with access to $\mathcal{C}$ and processing an estimation algorithm as per the aforementioned example, then by restricting the allowable events as above and choosing such a simple method of indexing these, we are able to limit the additional computational power given to the agent and/or poly-box.

9.2 Example: A poly-box over $\mathcal{C}_{\text{IQP}}$

IQP circuits [15, 90, 16, 70] consist of computational basis preparation and measurements with all gates diagonal in the X basis. This family of circuits is a well studied intermediate model for quantum computing.

We show that IQP circuits admit a poly-box. One can construct such a poly-box over $\mathcal{C}_{\text{IQP}}$ by noting that Proposition 5 from Ref. [87] gives a closed form expression for all Born rule probabilities and marginals of these circuits [91]. This expression:

$$\mathcal{P}_P(S) = \mathbb{E}_{r \in \text{span}\{\, \vec{e}_i \,\mid\, i \in \{\, i_1, \dots, i_k \,\}\,\}} \left[(-1)^{r \cdot s} \alpha \left(P_r, \frac{\pi}{4} \right) \right] \tag{9.1}$$

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 [92] and Ref. [87]. As such, one can construct an additive polynomial precision estimator for all Born rule probabilities and marginals simply by evaluating the expression:

$$\hat{p}_1 = \mathrm{Re}\left[(-1)^{r \cdot s} \alpha\left(P_r, \frac{\pi}{4}\right)\right] \tag{9.2}$$

for polynomially many independent uniformly randomly chosen $r \in \mathrm{span}\left\{\, \vec{e}_i \;\middle|\; i \in \{\, i_1, \ldots, i_k \,\}\right\}$ and computing the average over all choices. This can be shown to produce a poly-box by application of the Hoeffding inequality.

Chapter 10

From estimation to simulation

Given the significance of ϵ-simulation as the notion that minimally preserves computational power, here we turn our attention to the construction of an efficient algorithms for lifting a poly-box to an ϵ-simulators. We give strong evidence that in the general case, such a construction is not possible. This suggests that a poly-box is statistically distinguishable from an ϵ-simulator and hence computationally less powerful. However, by restricting to a special family of quantum circuits, we show an explicit algorithm for lifting a poly-box to an ϵ-simulator. Combined with Thm. 7 this shows that within this restricted family a poly-box is computationally equivalent to an ϵ-simulator.

The significance of ϵ-simulation also motivates the need to understand the relationship to other simulators defined in terms of Born probability estimation. At the end of this section and in Appendices C and D we present two algorithms which lift an estimator of probabilities and marginals to a sampler.

10.1 A poly-box is not sufficient for ϵ-simulation

This section focuses on the relation between poly-boxes and ϵ-simulation. With a poly-box, one can efficiently estimate Born rule probabilities of outcomes of a quantum circuit with additive precision. However, assuming BQP$\neq$ BPP, a poly-box alone is not a sufficient computational resource for ϵ-simulation. We illustrate this using a simple but somewhat contrived example, wherein an encoding into a large number of qubits is used to obscure (from the poly-box) the computational power of sampling [93].

Define a family of quantum circuits $\mathcal{C}_e$ 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 $Z = (X \oplus \mathrm{Par}(Y), Y) \in \{0,1\}^{n+1}$, where $\mathrm{Par}(Y)$ is the parity function on the input bit-string Y.

We note that $\mathcal{C}_e$ cannot admit an ϵ-simulator unless BQP⊆BPP, since simple classical post processing reduces the ϵ-simulator over $\mathcal{C}_e$ to an ϵ-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 ϵ 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 ϵ case: if $\epsilon < 1/2^n$, explicitly compute the quantum probability $p := \Pr(X = 1)$;
   (b) large ϵ 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 8. *For all $a \in \mathcal{A}^*$, $\epsilon > 0$ and $S \in \{0,1,\bullet\}^n$, the above poly-box can output estimates within ϵ additive error of the target probability using $O(poly(n, 1/\epsilon))$ 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 $O(poly(n, 1/\epsilon))$. Since in the case of small ϵ it is $O(poly(2^n)) \subseteq O(poly(1/\epsilon))$ and in the case of large ϵ 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 ϵ case as the estimates are exact in the alternate case.

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

$$\Pr(Z = z) = \Pr(Z_0 = z_0 \mid Y = z') \Pr(Y = z') = \Pr(X = z_0 \oplus \mathrm{Par}(z')) 2^{-n} \,. \tag{10.1}$$

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

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

$$p(S) = \sum_{z_i=0}^{1} \Pr(Z = z) = \sum_{z_i=0}^{1} \Pr(X = z_0 \oplus \mathrm{Par}(z')) 2^{-n} = p2^{-n} + (1-p)2^{-n} = 2^{-k} \,, \tag{10.2}$$

where $z_j := S_j$ for $j \neq i$. This implies 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 $\leq 2^{-(n+1)}$.

□

This example clearly demonstrates that the existence of a poly-box for a class of quantum circuits is not sufficient for ϵ-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 ϵ-simulation.

10.2 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 ϵ-simulator for $\mathcal{C}$, for some quantum circuit families, a poly-box suffices. Here, we show that one can construct an ϵ-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 [82] regarding sparse distributions. In Ref. [82], they define the following property of discrete probability distributions:

Definition 6. *(ϵ-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 ϵ-approximately t-sparse if it has a L_1 distance less than or equal to ϵ from some t-sparse distribution.*

The lemma below is a (slightly weakened) restatement of Thm. 11 from Ref. [82].

Lemma 9. *(Thm. 11 of Ref. [82]). Let $\mathcal{P}$ be a distribution on $\{0,1\}^k$ that satisfies the following conditions:*

1. *$\mathcal{P}$ is promised to be ϵ-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, ϵ^{-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 ϵ^{-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 ϵ^{-1}. This fact motivates the following definition:

Definition 7. *(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 ϵ-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 ϵ-approximately t-sparse whenever $t \geq P(k_a/\epsilon)$.

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

$$2^{H^{\epsilon}_{\max}(\mathcal{P})} \leq P\left(\frac{n}{\epsilon}\right) \tag{10.3}$$

where $H^{\epsilon}_{\max}(\mathcal{P}) := \inf_{\mathcal{P}'} \log_2 |Supp(\mathcal{P}')|$, $|Supp(\mathcal{P}')|$ is the cardinality of the support of the distribution $\mathcal{P}'$ and the infimum is taken over all distributions $\mathcal{P}'$ subject to $\frac{1}{2}||\mathcal{P}' - \mathcal{P}||_1 \leq \epsilon$. This notion was first defined in Ref. [94] where it corresponds to the ϵ-smooth Rényi entropy of order $\alpha = 0$.

10.3 Conditions for ϵ-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 10. *Let $\mathcal{C}$ be a family of quantum circuits with a corresponding family of probability distributions $\mathbb{P}$. Suppose there exists a poly-box over $\mathcal{C}$, and that $\mathbb{P}$ is poly-sparse. Then, there exists an ϵ-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 ϵ-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 Lem. 9, it is possible to classically sample from a probability distribution $\mathcal{P}^{\epsilon}_a \in B(\mathcal{P}_a, \epsilon)$ efficiently in ϵ^{-1}, t and k_a. We note that here we have removed the dependence on δ since we can make $\delta \leq \epsilon$ whilst remaining efficient in ϵ^{-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. □

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 $poly(n)$. Then $\mathcal{C}$ is poly-sparse (even if the output distributions are uniform over the $poly(n)$ sized support). Hence, if $\mathcal{C}$ also admits a poly-box, then by Thm. 10 one can with high probability repeatedly sample from this space of $poly(n)$ outcomes hidden within a exponentially large space of bit-strings.

We have shown that having a poly-box and a poly-sparsity guarantee for a family of quantum circuits gives us a ϵ-simulator. We emphasize that this approach allows for the ϵ-simulation of families of quantum circuits for which no known weak simulation method exists. Specifically, by combining the results from Ref. [1] 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 ϵ-simulated.

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. [33], are ϵ-simulable.

Thm. 10 requires a promise of poly-sparsity. Since this is a property of infinite families of probability distributions, one cannot hope to algorithmically verify (or even falsify) it through sampling from member distributions. Nevertheless, for distributions generated by some particular family of quantum circuits, a proof that this property holds may be possible.

In summary, the results of Thms. 10 and 7 imply that in order to construct an ϵ-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. 10.1, 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 ϵ-simulation. In the next section, we show that this phenomenon also occurs with much more natural families of quantum circuits.

10.4 On lifting stronger estimators to approximate samplers

In contrast to poly-boxes, certain stronger nations of simulation based on Born rule probability estimation can be lifted to ϵ-simulators (or even stronger approximate weak simulators). In Appendices C and D we present two such efficient classical algorithms.

The algorithm presented in Appendix D uses an estimator with multiplicative precision to construct an ϵ-simulator (it can in fact construct an approximate weak simulator based on the stronger notion from Ref. [26]). This algorithm exploits the fact that ratios of multiplicative precision estimators are multiplicative precision in order to sequentially, one qubit's measurement outcome at a time, sample from the marginal probability of the next qubit's measurement conditioned on the sampled measurement outcomes of the prior meaurements. This algorithm and its variants have been presented in [19, 20, 26] and are well known within the simulation-of-quantum-circuits community.

The algorithm presented in Appendix C uses an estimator with exponentially small additive precision to construct an ϵ-simulator (it can in fact construct an approximate weak simulator based on the stronger notion from Ref. [25]). This algorithm aims to map a bit-string r (approximately representing a uniformly sampled point from the unit interval) to a bit-string representing the outcome of running the circuit. Such a mapping is defined for every ordering of the measurement outcomes. This algorithm makes intuitive use of marginal probability estimates to do a binary search for the measurement outcome corresponding to r. This technique avoids computing ratios of probability estimates making it useful in regimes where additive errors are small but larger than some of the probabilities in the target distribution. Hence, this algorithm has some advantages compared to that of Appendix D. In particular, it can be used to lift an additive ε precision estimator to a sampler from within L_1 distance $O(2^n\varepsilon)$. This can be used to construct a ϵ-simulator in certain cases where the algorithm in Appendix D would fail. An example is when one has access to an estimator with additive precision $\varepsilon = 2^{-n}\kappa$ where $\kappa > 0$ can be made arbitrarily small in run-time $O(poly(n, 1/\kappa))$.

Chapter 11

Hardness results

In the previous section, we have shown that one can construct an ϵ-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 ϵ-simulable (unless BQP=BPP). In this section, we provide strong evidence (dependent only on standard complexity assumptions and a variant of the now somewhat commonly used [13, 70, 71, 72, 73, 84, 76] "average case hardness" conjecture) that a condition such as poly-sparsity is necessary even for natural families of quantum circuits. One such family has already been identified by noting that $\mathcal{C}_{\mathrm{IQP}}$ admits a poly-box and is hard to ϵ-simulate [70]. Here, we also show the hardness of ϵ-simulating the non-poly-sparse Clifford circuit family $\mathcal{C}_{\mathrm{PROD}}$ (defined in Sec. 9). These results meant that at least two (and possibly more) of the intermediate models of quantum computing have the property that the probability of individual outcomes and marginals can be estimated to $1/poly(n)$ additive error but due to non-sparsity, their ϵ-simulability is implausible.

Our hardness result for classical ϵ-simulation of $\mathcal{C}_{\mathrm{PROD}}$ closely follows the structure of several similar results, and in particular that of the IQP circuits result of Ref. [70]. We note that this hardness result is implied by the hardness results presented in Refs. [71, 72], however; our proof is able to use a more plausible average case hardness conjecture than these references due to the fact that we are proving hardness of ϵ-simulation rather than proving the hardness of the yet weaker notion of approximate weak simulation employed by these references.

Despite the existence of a poly-box over $\mathcal{C}_{\mathrm{PROD}}$, we show that there cannot exist a classical ϵ-simulator of this family unless the average case hardness conjecture fails or the polynomial hierarchy collapses to the third level. We note that the hardness of *exact* weak simulation of $\mathcal{C}_{\mathrm{PROD}}$ was shown in Ref. [57]. In contrast here we show the hardness of ϵ-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 $\mathcal{C}_{\mathrm{PROD}}$. This average case hardness conjecture is a strengthening of the worst case hardness of multiplicative precision estimation of probabilities associated with circuits from $\mathcal{C}_{\mathrm{PROD}}$.

The hardness of $\epsilon-$simulating $\mathcal{C}_{\mathrm{PROD}}$ circuits is shown by first noting that the existence of a classical ϵ-simulator implies, via the application of the Stockmeyer approximate counting algorithm [95], the existence of an algorithm (in the third level of the PH) for estimating the probabilities associated with the output distribution of the ϵ-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 ϵ-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 ϵ-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 2 thus implying that an ϵ-simulator does not exist.

11.1 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 2. *There exist an input product state ρ over n qubits such that given a uniformly random Clifford unitary U acting on n qubits, estimating $p := \operatorname{tr}\left(U\rho U^\dagger|0\rangle\langle 0|\right)$ to within a multiplicative error of $1/poly(n)$ for 49% or more of the sampled Clifford unitaries is #P-hard.*

We note that this average case hardness conjecture has an analogous worst case hardness version[1]. The worst case hardness can be proven by applying the result of Refs. [88, 57, 46, 89, 45] and by an argument essentially identical to the proof of Thm. 5.1 in Ref. [76]. 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 that use a gate set with algebraic entries, to within any multiplicative factor in the open interval $(1, \sqrt{2})$ is #P-hard [45, 46] ;

2. for gate sets with algebraic entries, all non-zero output probabilities are lower bounded by some inverse exponential [89];

3. that $\mathcal{C}_{\text{PROD}}$ circuits with post-selection (or adaptivity) are universal for quantum computation [88, 57].

We emphasize that similar conjectures are commonly used in related hardness proofs, such as Refs. [13, 70, 71, 73, 84, 76].

[1] This is the same statement as per Conjecture 2 but with "49% or more" replaced by "100%".

11.2 Anti-concentration of outcomes for $\mathcal{C}_{\mathrm{PROD}}$

Next, we prove that Clifford circuits chosen uniformly at random from the family $\mathcal{C}_{\mathrm{PROD}}$ satisfy an anti-concentration property.

Lemma 11. *Let d be a prime. For each $n \in \mathbb{N}$, let c_n be an n-qudit Clifford circuit chosen by fixing an arbitrary n qudit input state ρ, applying a uniformly random Clifford unitary U acting on n qudits and doing a computational basis measurement on all qudits. Then for all $\alpha \in (0,1)$ and for any fixed choice of $x \in \{0, \ldots, d-1\}^n$:*

$$\Pr_U \left(p_x \geq \frac{\alpha}{d^n} \right) > \frac{(1-\alpha)^2}{2}, \tag{11.1}$$

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

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

$$\begin{aligned}
\mathbb{E}(p_x) &= \mathrm{tr}\left(\mathbb{E}(\mathrm{U}\rho\mathrm{U}^\dagger)|\mathrm{x}\rangle\langle\mathrm{x}| \right) = \mathrm{tr}\left(\mathbb{1}/\mathrm{d}^\mathrm{n} |\mathrm{x}\rangle\langle\mathrm{x}| \right) = \frac{1}{\mathrm{d}^\mathrm{n}} &(11.2)\\
\mathbb{E}(p_x^2) &= \mathrm{tr}\left(\mathbb{E}\big(\mathrm{U}\otimes\mathrm{U}(\rho\otimes\rho)\mathrm{U}^\dagger\otimes\mathrm{U}^\dagger\big)|\mathrm{x}\rangle|\mathrm{x}\rangle\langle\mathrm{x}|\langle\mathrm{x}| \right)\\
&= \frac{\mathrm{tr}\big(\mathrm{P}_{\mathrm{Sym}}(\rho\otimes\rho)\big)}{tr P_{\mathrm{Sym}}}\,\mathrm{tr}\left(\mathrm{P}_{\mathrm{Sym}}|\mathrm{x}\rangle|\mathrm{x}\rangle\langle\mathrm{x}|\langle\mathrm{x}|\right)\\
&= \frac{2\mathrm{tr}\big(\mathrm{P}_{\mathrm{Sym}}(\rho\otimes\rho)\big)}{d^n(d^n+1)}\\
&= \frac{\mathrm{tr}\left(\rho^2\right) + (\mathrm{tr}\ \rho)^2}{d^n(d^n+1)}\\
&\leq \frac{2}{d^n(d^n+1)}, &(11.3)
\end{aligned}$$

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

$$\Pr\left(R \geq \alpha \mathbb{E}[R] \right) \geq (1-\alpha)^2 \frac{\mathbb{E}^2[R]}{\mathbb{E}[R^2]}, \qquad \text{(Paley-Zygmund inequality)} \tag{11.4}$$

Application of this inequality with Eqs. (11.2-11.3) then gives the desired result. □

We point out that the property of anti-concentration is inconsistent with poly-sparsity. This result is shown in Thm. 18 of Appendix E.

11.3 Hardness theorem

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

Theorem 12. *If there exists an ϵ-simulator of $\mathcal{C}_{\text{PROD}}$ and Conjecture 2 holds, then the polynomial hierarchy collapses to the third level.*

Proof. Assuming there exists an ϵ-simulator of $\mathcal{C}_{\text{PROD}}$, we can treat the ϵ-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), $r \in \{0,1\}^{poly(n/\epsilon)}$ (representing the random bit-string) and $d_c \in \mathcal{A}^{poly(n)}$ (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 ϵ in L_1 distance in time $\text{poly}(n, 1/\epsilon)$. That is, the output satisfies:

$$||p - p^\epsilon||_1 := \sum_{x \in \{0,1\}^k} |p_x - p^\epsilon_x| \leq \epsilon \tag{11.5}$$

where $p_x := \Pr(X = x)$ is the probability of observing outcome x on a single run of the quantum circuit c and $p^\epsilon_x := \Pr\limits_{r \sim unif}(X^\epsilon = x)$ 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 ϵ, d_c inputs.

We now note that the problem of computing the proportion p^ϵ_x 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^ϵ_x to within a multiplicative error in the complexity class FBPP^{NP}.

More precisely, there exists an algorithm in FBPP^{NP} which will output an estimate $\tilde{p}^\epsilon_x$ such that:

$$|p^\epsilon_x - \tilde{p}^\epsilon_x| \leq \frac{p^\epsilon_x}{poly(n)} \tag{11.6}$$

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

$$\begin{aligned} |p_x - \tilde{p}^\epsilon_x| &\leq |p_x - p^\epsilon_x| + |p^\epsilon_x - \tilde{p}^\epsilon_x| \\ &\leq |p_x - p^\epsilon_x| + \frac{p^\epsilon_x}{poly(n)} \\ &\leq |p_x - p^\epsilon_x| + \frac{p_x + |p_x - p^\epsilon_x|}{poly(n)} \\ &= |p_x - p^\epsilon_x| \left(1 + \frac{1}{poly(n)}\right) + \frac{p_x}{poly(n)} \end{aligned} \tag{11.7}$$

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

$$\mathop{\mathbb{E}}_x[|p_x - p^\epsilon_x|] = \frac{1}{2^k} \sum_x |p_x - p^\epsilon_x| = \frac{1}{2^k} ||p - p^\epsilon||_1 \leq \frac{\epsilon}{2^k}$$

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

$$\mathop{\mathbb{E}}_x[|p_x - p^\epsilon_x|] \leq \frac{\epsilon}{2^n} \tag{11.8}$$

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

$$\Pr\left(R \geq \frac{\mathbb{E}[R]}{\gamma}\right) \leq \gamma\,, \qquad \text{(Markov's inequality)} \tag{11.9}$$

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

$$\Pr_x\left(|p_x - p_x^\epsilon| \geq \frac{\mathbb{E}_x[|p_x - p_x^\epsilon|]}{\beta}\right) \leq \beta \tag{11.10}$$

That is:

$$\Pr_x\left(|p_x - p_x^\epsilon| < \frac{\epsilon}{\beta 2^n}\right) > (1-\beta) \tag{11.11}$$

Applying this to the upper bound in Eq. (11.7), we find that for all $\beta > 0$:

$$\Pr_x\left(|p_x - \tilde{p}_x^\epsilon| < \frac{\epsilon}{\beta 2^n}\left(1 + \frac{1}{poly(n)}\right) + \frac{p_x}{poly(n)}\right) > (1-\beta) \tag{11.12}$$

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

By Eq. (11.1), we have $\Pr_U(A) > \frac{(1-\alpha)^2}{2}$ and by Eq. (11.12), 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:

$$\Pr_{U,x}\left(|p_x - \tilde{p}_x^\epsilon| < \frac{\epsilon p_x}{\alpha\beta}\left(1 + \frac{1}{poly(n)}\right) + \frac{p_x}{poly(n)}\right) > \frac{(1-\alpha)^2 - 2\beta}{2} \tag{11.13}$$

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:

$$\begin{aligned} \Pr_{U,U_x}(U_x U = V) &= \Pr_{U,U_x}(U = U_x V) && (11.14)\\ &= \Pr_U(U = V) && (11.15)\\ &= \Pr_U(U_y U = V) && (11.16) \end{aligned}$$

where probabilities over U_x are chosen uniformly over all $x \in \{0,1\}^n$. Applying this to Eq. (11.13) we find that for all $y \in \{0,1\}^n$ and for all n qubit product states ρ;

$$\Pr_U\left(|p_y - \tilde{p}_y^\epsilon| < \frac{\epsilon p_y}{\alpha\beta}\left(1 + \frac{1}{poly(n)}\right) + \frac{p_y}{poly(n)}\right) > \frac{(1-\alpha)^2 - 2\beta}{2} \tag{11.17}$$

We recall that for an ϵ-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 α, β, ϵ:

$$\alpha = \frac{1}{n}, \qquad \beta = \frac{1}{2n^2}, \qquad \epsilon = \frac{\alpha\beta}{n}. \tag{11.18}$$

This argument shows that the existence of an ϵ-simulator of $\mathcal{C}_{\text{PROD}}$ implies that there exists and algorithm in Δ_3^p that can for any fixed product states ρ and measurement outcomes $x \in \{0,1\}^n$, output an $O(1/n)$ multiplicative precision estimate of $p_x := \operatorname{tr}\left(U\rho U^\dagger |x\rangle\langle x|\right)$ for almost 50% of randomly uniformly chosen Clifford unitaries U acting on n qubits. That is:

$$\Pr_U\Big[|p_x - \tilde{p}_x^\epsilon| < p_x O(1/n)\Big] > \frac{1}{2} - \frac{1}{n} \tag{11.19}$$

By conjecture 2, this is #P-hard. This implies that a #P-hard problem is solved in FBPP^{NP}. By Toda's theorem [96], this collapses the polynomial hierarchy to its third level. □

Chapter 12

Discussion: Part 2

There is a substantial and growing body of results showing the classical "simulability" of some quantum computers and the hardness of "simulability" of others. We hope that the results presented here will significantly inform the interpretation of this literature in relation to the comparison of the computational power of the relevant quantum computer to the computational power of a universal classical computer. For some family of quantum circuits $\mathcal{C}$, these results typically make statements of the form either:

- Simulability: $\mathcal{C}$ can be classically "simulated" or
- Hardness: $\mathcal{C}$ can be classically "simulated" implies some implausible outcome

In the case of simulability proofs, our results show that whenever the notion of simulation used is stronger or equivalent to ϵ-simulation, the useful computational power of $\mathcal{C}$ is contained within classical. Further, if the notion of simulation is a poly-box (a weaker notion then ϵ-simulation), this still applies provided that $\mathcal{C}$ is poly-sparse. If $\mathcal{C}$ is not known to be poly-sparse but admits a poly-box then, we can still conclude that without non-trivial classical post-processing, $\mathcal{C}$ is incapable of solving decision problems outside of the complexity class BPP.

In the case of hardness proofs, our results show that whenever the notion of simulation used is weaker or equivalent to ϵ-simulation, it is plausible that the useful computational power of $\mathcal{C}$ is beyond classical. However, for proofs of hardness based on weaker notions of simulation, it may be possible to alter the proof such that it shows the hardness of ϵ-simulation (rather than a yet weaker notion) with the added benefit that now the hardness is more plausible.

Some hardness results show the implausibility of classically simulating $\mathcal{C}$ with respect to a notion of simulation much stronger than ϵ-simulation. Even if quantum computers can reliably achieve such a notion of simulation, these results cannot be seen as showing the implausibility of the existence of efficient classical devices that can be used as a perfectly good computational substitute to $\mathcal{C}$.

The perspective of efficient indistinguishability gives us a natural avenue to defining the set of all problems solvable by a quantum device. We have seen that the minimal notion of simulation to achieve this is ϵ-simulation; a significantly weaker notion of simulation than many of the notions used in literature [25, 26, 16, 27]. Thus, the gap between classical and quantum computational power can be closed not only by the development of more powerful classical simulation algorithms but also by significantly reducing the computational hurdle classical devices must overcome in order to act as efficient substitutes to quantum computers. Our results exploit this feature in order to show that any family of quantum circuits that both admits a poly-box and satisfies the poly-sparsity condition can be ϵ-simulated. The existence of multiple known constructions of poly-boxes (see Refs. [18, 43, 1, 2, 33]) over restricted families of quantum circuits, and in particular Ref. [1], demonstrates the significant advantages offered by weakening the minimal requirements on classical simulators from the stronger notions of weak simulation to that of ϵ-simulation.

For any given family of quantum circuits, poly-sparsity can be trivially guaranteed by upper bounding the number of measured qubits by $\log\ n$. However, the condition of poly-sparsity permits significantly more complex probability distribution families (including families with exponentially growing support). Future exploration of how to non-trivially guarantee poly-sparsity offers yet more potential for identifying interesting families of quantum circuits that are ϵ-simulable using the techniques outlined here.

In this paper, we have argued that ϵ-simulation minimally captures computational power. However, the term "minimally" is with respect to the computational power of the referee, which is unbounded in the setting we considered. This raises the importance of future work aimed at defining the notion of simulation which minimally captures the computational power of a quantum computer with respect to a referee that is computationally bounded to universal quantum computation (or equivalent). In light of this observation, our work suggests that even requiring a simulator to be capable of solving all sampling problems (as defined in Ref. [85]) solvable by the quantum device is too strong to be minimal (w.r.t. a universal quantum bounded referee). Future results in this direction would inform us on precisely how to further weaken the notion of sampling problems and to define a yet weaker complexity class than SampBQP (or more generally Samp$\mathcal{C}$) that (w.r.t. a universal quantum bounded referee) minimally captures computational power.

In an experimental setting where there is a constant lower bound to the noise present in the quantum device, the minimal requirements for efficient indistinguishability become yet weaker. In this setting, it is plausible that for IQP circuits and boson sampling circuits, classical computation can achieve the minimal requirements for efficient indistinguishability w.r.t. a universal quantum bounded referee. This possibility is supported by the existence of classical algorithms for simulating noisy IQP circuits [78] and noisy boson sampling circuits [79]. In the constant lower bounded noise setting, these algorithms fail to achieve efficient indistinguishability w.r.t. a computationally unbounded referee. However, whether or not they achieve efficient indistinguishability w.r.t. a universal quantum bounded referee remains a question to be resolved.

Aiming to tighten the separation between simulability and hardness is an important goal toward a deeper understanding of the computational power of quantum verses classical circuits. Specifically, the aim is to move towards a full classification of simulablity by gradually reducing the "unclassified" space (of parameters describing a quantum computer that are both outside the range to ensure simulabilty and outside the range to ensure hardness of simulability). By focusing on the tension between anti-concentration and poly-sparsity our work has made modest progress in this direction with potential for further consolidation and progress with respect to this aim.

We have shown the poly-sparsity and anti-concentration properties to be mutually exclusive. If we assume that the polynomial hierarchy does not collapse and restrict to quantum computers that admit a poly-box and average case hardness (some plausible candidates being $\mathcal{C}_{\mathrm{IQP}}$, $\mathcal{C}_{\mathrm{PROD}}$, and their poly-sparse restricted counterparts) we see that either poly-sparsity holds ensuring ϵ-simulability or anti-concentration holds ensuring hardness.

For general quantum circuit families, poly-sparsity and anti-concentration are not exhaustive. Future work directed towards finding interesting spaces of quantum computers where the two notions are exhaustive would help to classify more of the yet unclassified computers in Fig. 7.1 (admits a poly-box and not poly-sparse). Restricted to this setting, all quantum computers that admit the appropriate average case hardness property would admit a hardness proof. Further, such work can give a much needed new perspective on the peculiar nature of the transition from ϵ-simulable to hardness that IQP and magic state injected Clifford circuit families undergo as they transition from poly-sparse to non-poly-sparse. In particular, this may shed light on whether this behavior (shared by IQP, magic state injected Clifford circuit and possibly others) is common to intermediate models of quantum computing for a good reason or simply a coincidence.

Our work establishes the conceptual importance of a poly-box as a notion of simulation. Through the Hoeffding inequality and powerful sampling techniques such as Monte Carlo simulations, we inherit a number of important examples of poly-boxes including IQP circuits, magic state injected Clifford circuit and circuits with polynomially bounded negativity (see also Refs. [18, 43, 1, 2, 33]). This is of immediate practical interest as admitting a poly-box is sufficient for many useful problems such as finding certain expectation values or estimating the probability associated with certain events.

Whether or not a family of quantum circuits $\mathcal{C}$ admits a poly-box significantly informs our understanding of the computational power of $\mathcal{C}$ relative to classical. Simulability of a family of quantum circuits $\mathcal{C}$ according to the notion of a poly-box, implies that, without additional classical computational resources, $\mathcal{C}$ cannot solve decision problems outside of classical. If $\mathcal{C}$ is also poly-sparse (with binary outcome circuits being a very special case) then even an agent with universal classical computational power and access to the quantum computer $\mathcal{C}$ is confined to universal classical computational power. However, when supplemented with a universal classical computer, if $\mathcal{C}$ admits a poly-box but is not poly-sparse then it may be capable of solving decision problems beyond BPP. This possibility is not ruled out by our analysis and is consistent with the fact that $\mathcal{C}_{\mathrm{PROD}}$ and $\mathcal{C}_{\mathrm{IQP}}$ circuits both admit hardness proofs.

There is something conceptually unclear about circuit families that admit poly-boxes and a hardness proof of the type presented in Sec. 11. In particular, it is unclear if these admit a poly-box purely due to the restriction placed on the types of events that a poly-box can be queried about, or if hardness of ϵ-simulation could manifest even in circuit families which allow efficient classical polynomial precision estimation of probabilities associated with any family of events decidable in BPP. In the latter case, an agent with access to such circuits cannot solve any decision problem outside of BPP even given access to a universal classical computer. The former case leaves open the possibility that these families of circuits will behave like $\mathcal{C}_e$ (introduced in Sec. 10.1) where some appropriate classical post-processing of outcome samples will render them more powerful than BPP (assuming BQP$\not\subseteq$ BPP). This question is closely related to an open question raised by Aaronson in Ref. [85].

It is surprising that examples of families of circuits that admit a poly-box and a plausible hardness proof are far from rare and in fact may be typical among intermediate models of quantum computing. In addition to the families we have shown to be in this category ($\mathcal{C}_{\text{PROD}}$ and $\mathcal{C}_{\text{IQP}}$), we note that linear optical networks $\mathcal{C}_{\text{LON}}$ and circuits with polynomially bounded negativity $\mathcal{C}_{poly\mathcal{N}}$ are also plausible candidates. We note that due to an algorithm by Gurvits [69] (see also Ref. [97]), the family of linear optical quantum circuits considered in the boson sampling setting of Ref. [13] 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 *all* marginal probabilities. Alternatively, $\mathcal{C}_{poly\mathcal{N}}$ is known to admit a poly-box [1]. Also, for odd prime d it contains the qudit generalization of $\mathcal{C}_{\text{PROD}}$ which is both universal under post-selection and anti-concentrates. Hence an average case hardness conjecture is also plausible implying that $\mathcal{C}_{poly\mathcal{N}}$ admits a proof of hardness essentially identical to that of $\mathcal{C}_{\text{PROD}}$. In light of these considerations we are optimistic that useful and computationally interesting applications can be found for intermediate models of quantum computation.

Appendices

Appendix A

Hoeffding's inequality

The original Hoeffding's Inequality reads [52]:

Theorem 13 (Hoeffding's Inequality)**.** *Let $\{X_s\}_{s=1}^S$ be independent real valued random variables such that $X_s \in [a_s, b_s]$ for all $s \in [S]$. Then, for all $\epsilon > 0$ the following holds:*

$$\Pr\left(\bar{X} - \mu \geq \epsilon\right) \leq \exp\left(\frac{-2S^2\epsilon^2}{\sum_{s=1}^S (a_s - b_s)^2}\right) \tag{A.1}$$

where $\bar{X} = \frac{1}{S}\sum_{i=1}^S X_i$ and $\mu = \langle \bar{X} \rangle$.

Appendix B

Upper bounding Born rule probabilities from estimates

In this section we derive Eq. (5.64). This equation provides a probabilistic upper bound for the Born rule probability p in terms of algorithmic parameters s and L and an estimate $\hat{p}$ produced using our estimation algorithm from Ch. 5 with these parameters.

Let us consider a small increase on the minimal choices of s and L given by Eqs. (5.44) and (5.46) combined with the choice $\delta = \tilde{\delta} = \delta_{\text{tot}}/4$:

$$s = \frac{2(2^{\gamma t/2}+1)^2}{\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2} \log\left(\delta_{\text{tot}}/8e^2\right)^{-1}, \tag{B.1}$$

$$= \frac{a}{\left(\sqrt{p+\epsilon}-\sqrt{p}\right)^2} \tag{B.2}$$

and

$$L = \left(\frac{p+\epsilon}{\epsilon_{\text{tot}}-\epsilon}\right)^2 \log\left(\delta_{\text{tot}}/4\right)^{-1}, \tag{B.3}$$

$$= \left(\frac{p+\epsilon}{\epsilon_{\text{tot}}-\epsilon}\right)^2 b. \tag{B.4}$$

where we have defined $a := 2(2^{\gamma t/2}+1)^2 \log\left(\delta_{\text{tot}}/8e^2\right)^{-1}$ and $b := \log\left(\delta_{\text{tot}}/4\right)^{-1}$.

Eqs. (B.2) and (B.4) can be rewritten as follows:

$$\epsilon = \left(\sqrt{p}+\sqrt{\frac{a}{s}}\right)^2 - p,$$

$$= 2\sqrt{\frac{pa}{s}} + \frac{a}{s} \tag{B.5}$$

and

$$\begin{aligned}\epsilon_{\text{tot}} &= \epsilon + (p+\epsilon)\sqrt{\frac{b}{L}} \\ &= p\sqrt{\frac{b}{L}} + \epsilon\left(1+\sqrt{\frac{b}{L}}\right). \end{aligned} \tag{B.6}$$

By substituting Eq. (B.5) into Eq. (B.6), it is easy to show that:

$$\begin{aligned}\epsilon_{\text{tot}} &= p\sqrt{\frac{b}{L}} + \left(2\sqrt{\frac{ap}{s}} + \frac{a}{s}\right)\left(1+\sqrt{\frac{b}{L}}\right), && \text{(B.7)} \\ &\leq p\sqrt{\frac{b}{L}} + \left(2\sqrt{\frac{a}{s}} + \frac{a}{s}\right)\left(1+\sqrt{\frac{b}{L}}\right). && \text{(B.8)}\end{aligned}$$

We note that since this is an increasing function of p on the domain of interest, we can probabilistically upper bound the ϵ_{tot} by replacing p with a probabilistic upper bound, namely $p \leq \hat{p} + \epsilon_{\text{tot}}$ with high probability (w.h.p.) i.e. with probability of at least $1 - \delta_{\text{tot}}/2$. This gives that w.h.p.:

$$\epsilon_{\text{tot}} \leq (\hat{p} + \epsilon_{\text{tot}})\sqrt{\frac{b}{L}} + \left(2\sqrt{\frac{a}{s}} + \frac{a}{s}\right)\left(1+\sqrt{\frac{b}{L}}\right). \tag{B.9}$$

We now simplify to get:

$$\epsilon_{\text{tot}} \leq \hat{p}\frac{\sqrt{b/L}}{1-\sqrt{b/L}} + \frac{1+\sqrt{b/L}}{1-\sqrt{b/L}}\left(2\sqrt{\frac{a}{s}} + \frac{a}{s}\right). \tag{B.10}$$

Adding $\hat{p}$ to both sides and re-applying the probabilistic upper bound $p \leq \hat{p} + \epsilon_{\text{tot}}$, we find that the LHS upper bounds p giving the desired probabilistic upper bound:

$$p \leq \hat{p}\frac{1}{1-\sqrt{b/L}} + \frac{1+\sqrt{b/L}}{1-\sqrt{b/L}}\left(2\sqrt{\frac{a}{s}} + \frac{a}{s}\right). \tag{B.11}$$

Appendix C

Strong simulation implies EPSILON-simulation

In this appendix , we will show that the existence of a classical strong simulator of a family of quantum circuits implies the existence of an ϵ-simulator (it can in fact construct an approximate weak simulator based on the stronger notion from Ref. [25]). This algorithm aims to map a bit-string (representing the outcome of running the circuit) to r, which is sampled uniformly from [0,1]. While such a mapping is defined for every ordering of the measurement outcomes, it cannot be efficiently computed. This algorithm makes intuitive use of marginal probability estimates to do a binary search for the bit-string corresponding to r. This technique avoids computing ratios of probability estimates making it useful in regimes where additive errors are small but larger than some of the probabilities in the target distribution.

We start by giving a more precise definition of a strong simulator (than was presented in Sec. 8.1).

Definition 8. (strong simulator). *A strong simulator of a uniform 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, for all $a \in \mathcal{A}^*, \epsilon, \delta > 0$ and $S \in \{\, 0, 1, \bullet \,\}^{k_n}$, can be used to compute an estimate $\hat{p}$ of $\mathcal{P}_a(S)$ such that $\hat{p}$ satisfies the accuracy requirement:*

$$\Pr\big(|p - \hat{p}| \geq \epsilon\big) \leq \delta \tag{C.1}$$

and, the run-time required to compute the estimate $\hat{p}$ is $O(poly(n, \log\ \epsilon^{-1}, \log\ \delta^{-1}))$.

We point out that much like a poly-box, a strong simulator outputs estimates of Born probabilities. The key difference is that the precision of a strong simulator is exponential compared to the polynomial precision of a poly-box. In particular, for any polynomial f, a strong simulator can (efficiently in n) output estimates such that Eq. (C.1) is satisfied for $\epsilon \in \Omega(2^{-f(n)})$ (as opposed to a poly-box which generally requires $\epsilon \in \Omega(1/f(n))$). Hence, we note that the only difference between the definition of a strong simulator and that of a poly-box is the scaling of run-time in ϵ.

Theorem 14. *Let $\mathcal{C}$ be a uniform family of quantum circuits. If $\mathcal{C}$ admits a strong simulator, then $\mathcal{C}$ admits an ϵ-simulator.*

In fact we will prove an even stronger statement; that a strong simulator implies approximate weak simulation in the much stronger sense of approximate weak simulation used in Ref. [25] (exponentially small error in L_1 norm).

Before proving this theorem, we introduce an algorithm that uses output from a strong simulator to approximately sample from the output distribution of a circuit i.e. to produce output consistent with the definition of an ϵ-simulator. Without loss of generality, let $c \in \mathcal{C}$ be an arbitrary n qubit circuit with all n qubits measured. We will denote the quantum probabilities by p_S and the output of the strong simulator by $p_S^{\epsilon,\delta}$ suppressing the dependence on c.

To give a rough intuition, the algorithm will first sample a polynomial length bit-string $\tilde{r}$ which will be mapped to a probability $r \in [0,1]$. This value will remain fixed and be used throughout the algorithm until a sample $\tilde{X}$ is generated from the approximate output distribution. This sample will be the output of the ϵ-simulator upon a single execution with the input (ϵ', c). The sample $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_n)$ will be generated by sampling one bit at a time starting with $\tilde{X}_1$. The choice of the j^{th} bit $\tilde{X}_j$ is based on the comparisons between the output of the strong simulator $p_S^{\epsilon,\delta}$ and the probability r. This n step process will require n calls to the strong simulator where in each call, the only variation in the inputs is the events S_j. Each event S_j will be chosen based on the previously sampled values $\tilde{X}_1, \ldots, \tilde{X}_{j-1}$.

The algorithm will proceed as follows:

1. Fix $m \in \mathbb{N}$ and $\epsilon, \delta > 0$ based on $\mathcal{C}$ and the desired L_1 error upper bound, ϵ' (see later).
2. Sample $\tilde{r}$ uniformly from $\{0,1\}^m$.
3. Compute $r = \sum_{i=1}^{m} \tilde{r}_i 2^{-i}$
4. Set $S := (s_1, \ldots, s_n) = (\bullet, \ldots, \bullet)$.
5. Set $j = 1$.
6. Set $s_j = 0$.
7. Set $S_j = S$.
8. Request $p_{S_j}^{\epsilon,\delta}$ from the strong simulator.
9. If $p_{S_j}^{\epsilon,\delta} \geq r$, then set $\tilde{X}_j = 0$ otherwise, set $\tilde{X}_j = 1$.
10. Set $s_j = \tilde{X}_j$.
11. If $j = n$, output the string $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_n)$ and end.
12. Reset $j \rightarrow j+1$ and go to step 6.

We now prove Thm. 14.

Proof. We wish to show that for all acceptable families of quantum circuits $\mathcal{C}$, choices of $c \in \mathcal{C}$ and $\epsilon' > 0$:

- there exist a polynomially bounded function $f(\epsilon', n)$ which determines m and

- there exist functions for determining ϵ, δ

such that given a strong simulator of $\mathcal{C}$, the above algorithm can be executed in run-time $O(poly(n, \epsilon'^{-1}))$ and produce output $\tilde{X}$ from a distribution $\tilde{\mathcal{P}}$ satisfying $\tilde{\mathcal{P}} \in B(\mathcal{P}, \epsilon')$.

We note that the probability distribution over $x \in \{0,1\}^n$ defines a partitioning (up to sets of measure zero) of the unit interval into 2^n intervals[1] V_x labeled by x such that the uniform measure on these intervals corresponds to the quantum probability of outcome x. That is, we fix the partitioning such that for all $x \in \{0,1\}^n$:

$$\mu(V_x) = p_x. \tag{C.2}$$

To be specific, we can define $V_x = [v_x^-, v_x^+]$ where:

$$v_x^- = \sum_{x' < x} p_{x'} \tag{C.3}$$

$$v_x^+ = \sum_{x' \leq x} p_{x'} \tag{C.4}$$

$$\tag{C.5}$$

where, the above order on bit strings x' and x is defined by lexicographical ordering.

We note that given a uniform sample p from the unit interval, p will, up to measure zero, be strictly identified with an outcome $x \in \{0,1\}^n$ through the mapping $o : [0,1] \setminus D \to \{0,1\}^n$ implicitly defined by $p \in V_{o(p)}$ for all $p \in [0,1] \setminus D$ where $D := \{\, p_S \mid S \in \{\, 0, 1, \bullet \,\}^n \,\}$. Further, in the ideal case where the strong simulator produces output which is deterministically exact i.e. $p_S^{\epsilon,\delta} = p_S$ for all S, we note that the above algorithm would, for a given r, produce output $\tilde{X} = o(r)$. For r distributed uniformly on the unit interval, this ensures $\tilde{X}$ is sampled from exactly the quantum distribution. We thus note that two sources of error arise. The first is from the inaccuracies introduced by the strong simulator's output. The second is from having to approximate a uniform sample over $[0,1]$ by a uniform sample over $\{0,1\}^m$.

Let $\tilde{p}_x^{\epsilon,\delta}$ denote the probability $\Pr(\tilde{X} = x)$. Then, we have:

$$\tilde{p}_x^{\epsilon,\delta} = \sum_{\tilde{r} \in \{0,1\}^m} 2^{-m} \Pr(\tilde{X} = x \mid r) \tag{C.6}$$

Given an interval $V = [v^-, v^+]$ and $\alpha \in \mathbb{R}$, we define:

$$V^\alpha = \begin{cases} [v^- - \alpha, v^+ + \alpha], & \text{if } \alpha \geq 0 \text{ or } v^+ - v^- \geq 2\alpha \\ [\frac{v^- + v^+}{2}, \frac{v^- + v^+}{2}], & \text{otherwise} \end{cases} \tag{C.7}$$

If $p_x \geq 2\epsilon$ and $r \in V_x^{-\epsilon}$ then:

$$\Pr(\tilde{X} = x | r) \geq (1 - \delta)^n. \tag{C.8}$$

This can be seen by noting that with probability $\geq 1 - \delta$, each requested probability estimate in

[1] Here, we use a looser notion of interval by allowing points $p \in \mathbb{R}$ to constitute an intervals $[p, p]$.

step 8 will be within ϵ of the corresponding quantum probability resulting in $\tilde{X}_j = o(r)_j$.

Thus, we have:

$$\begin{aligned} \tilde{p}_x^{\epsilon,\delta} &\geq \sum_{\substack{\tilde{r}\in\{0,1\}^m \\ r\in V_x^{-\epsilon}}} 2^{-m}(1-\delta)^n && \text{(C.9)} \\ &\geq l_x 2^{-m}(1-\delta)^n && \text{(C.10)} \\ &\geq [p_x - 2\epsilon - 2^{-m}](1-\delta)^n && \text{(C.11)} \end{aligned}$$

where

$$\begin{aligned} l_x :&= \left\lfloor \frac{|V_x^{-\epsilon}|}{2^{-m}} \right\rfloor && \text{(C.12)} \\ &= \left\lfloor \frac{p_x - 2\epsilon}{2^{-m}} \right\rfloor && \text{(C.13)} \end{aligned}$$

is a lower bound on the number of bit strings $\tilde{r}$ which under the map in step 3 must be contained in the interval $V_x^{-\epsilon}$.

If $r \in \mathbb{R} \setminus V_x^{+\epsilon}$ then:

$$\Pr(\tilde{X} = x|r) \leq 1 - (1-\delta)^n \tag{C.14}$$

since estimates within ϵ of the target probability at each of the n iterations of step 8 will result in $\tilde{X} \neq x$.

Thus, we also have:

$$\begin{aligned} \tilde{p}_x^{\epsilon,\delta} &\leq \sum_{\substack{\tilde{r}\in\{0,1\}^m \\ r\in V_x^{+\epsilon}}} 2^{-m} \Pr(\tilde{X} = x \mid r) + \sum_{\substack{\tilde{r}\in\{0,1\}^m \\ r\in \mathbb{R}\setminus V_x^{+\epsilon}}} 2^{-m} 1 - (1-\delta)^n && \text{(C.15)} \\ &\leq u_x 2^{-m} + [1-(1-\delta)^n] && \text{(C.16)} \\ &\leq \left[p_x + 2\epsilon + 2^{-m}\right] + [1-(1-\delta)^n] && \text{(C.17)} \end{aligned}$$

where

$$\begin{aligned} u_x :&= \left\lfloor \frac{|V_x^{+\epsilon}|}{2^{-m}} \right\rfloor + 1 && \text{(C.18)} \\ &= \left\lfloor \frac{p_x + 2\epsilon}{2^{-m}} \right\rfloor + 1 && \text{(C.19)} \end{aligned}$$

is an upper bound on the number of bit strings $\tilde{r}$ which under the map in step 3 must be contained in the interval $V_x^{+\epsilon}$.

Thus $p_x \geq 2\epsilon$:

$$[-2\epsilon - 2^{-m}](1-\delta)^n - p_x\left[1-(1-\delta)^n\right] \leq \tilde{p}_x^{\epsilon,\delta} - p_x \leq \left[2\epsilon + 2^{-m}\right] + \left[1-(1-\delta)^n\right] \tag{C.20}$$

i.e.

$$\left|\tilde{p}_x^{\epsilon,\delta} - p_x\right| \leq \left[2\epsilon + 2^{-m}\right] + \left[1 - (1-\delta)^n\right] \tag{C.21}$$

Also, if $p_x \leq 2\epsilon$, then:

$$-p_x \leq \tilde{p}_x^{\epsilon,\delta} - p_x \leq \left[2\epsilon + 2^{-m}\right] + \left[1 - (1-\delta)^n\right] \tag{C.22}$$

thus, the bound from Eq. (C.21) also applies in this case.

This implies that:

$$\epsilon' \leq 2^n \left[2\epsilon + 2^{-m} + 1 - (1-\delta)^n\right]. \tag{C.23}$$

Clearly there exist choices of polynomials f_1, f_2, f_3 such that for $\epsilon' = \frac{1}{poly(n)}$ or even $\epsilon' = 2^{-poly(n)}$, Eq. (C.23) can be satisfied by choosing $\epsilon \leq 2^{-f_1(n)}$, $\delta \leq 2^{-f_2(n)}$ and $m \geq f_3(n)$. We complete the proof by noting that these choices ensure that the run-time of the strong simulator and the above algorithm are efficient in n and $1/\epsilon'$. □

Appendix D

Multiplicative precision simulation implies EPSILON-simulation

In this section, we present an algorithm (very similar to Ref. [26]) which uses an estimator with multiplicative precision to construct an ϵ-simulator (it can in fact construct an approximate weak simulator based on the stronger notion from Ref. [26]). This algorithm exploits the fact that ratios of multiplicative precision estimators are multiplicative precision in order to sequentially, one qubit's measurement outcome at a time, sample from the marginal probability of the next qubit's measurement conditioned on the sampled outcomes of the prior measurements. This algorithm and its variants have also been presented in [19, 20] and are well known within the simulation-of-quantum-circuits community.

Here, we claim without proof that this algorithm lifts a classical multiplicative precision simulator of a family of quantum circuits to an approximate weak simulator based on the stronger notion from Ref. [26]. This result has been shown in Ref. [26], but we discuss it here for completeness.

We start by giving a definition of a multiplicative precision simulator.

Definition 9. (multiplicative precision simulator). *A multiplicative precision simulator of a uniform 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, for all $a \in \mathcal{A}^*, \epsilon, \delta > 0$ and $S \in \{\, 0, 1, \bullet \,\}^{k_n}$, can be used to compute an estimate $\hat{p}$ of $\mathcal{P}_a(S)$ such that $\hat{p}$ satisfies the accuracy requirement:*

$$\Pr\big(|p - \hat{p}| \geq \epsilon p\big) \leq \delta \tag{D.1}$$

and, the run-time required to compute the estimate $\hat{p}$ is $O(poly(n, \epsilon^{-1}, \delta^{-1}))$.

We claim that a multiplicative precision simulator can be used to construct an ϵ-simulator.

Theorem 15. *Let $\mathcal{C}$ be a uniform family of quantum circuits. If $\mathcal{C}$ admits a multiplicative precision simulator, then $\mathcal{C}$ admits an ϵ-simulator.*

We omit a complete proof of this theorem as it makes straightforward use of standard techniques. However, we outline the algorithms which uses output from a multiplicative precision simulator to approximately sample from the output distribution of a circuit. Without loss of generality, let $c \in \mathcal{C}$ be an arbitrary n qubit circuit with all n qubits measured. We will denote the quantum probabilities by p_S and the output of the multiplicative precision simulator by $p_S^{\epsilon,\delta}$ suppressing the dependence on c.

The algorithm will proceed as follows:

1. Fix $m \in \mathbb{N}$ and $\epsilon, \delta > 0$ based on $\mathcal{C}$ and the desired L_1 error upper bound, ϵ' (see later).

2. Set $S := (s_1, \ldots, s_n) = (\bullet, \ldots, \bullet)$.

3. Set $p_{S_0}^{\epsilon,\delta} := 1$.

4. Set $j = 1$.

5. Set $s_j = 0$.

6. Set $S_j = S$.

7. Request $p_{S_j}^{\epsilon,\delta}$ from the multiplicative precision simulator.

8. Compute $c_j := p_{S_j}^{\epsilon,\delta} / p_{S_{j-1}}^{\epsilon,\delta}$

9. Sample $\tilde{r}$ uniformly from $\{0,1\}^m$.

10. Compute $r = \sum_{i=1}^{m} \tilde{r}_i 2^{-i}$

11. If $c_j \geq r$, then set $\tilde{X}_j = 0$ otherwise, set $\tilde{X}_j = 1$.

12. Set $s_j = \tilde{X}_j$.

13. If $j = n$, output the string $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_n)$ and end.

14. Reset $j \to j + 1$ and go to step 5.

We note that multiplicative precision estimate can divide each other and still produce a multiplicative precision estimate. Hence c_j computed in step 8 is a multiplicative precision estimate of the quantum conditional probability $p_{S_j}/p_{S_{j-1}} = \Pr(X_j = x_j \mid X_1 = x_1, \ldots, X_{j-1} = x_{j-1})$. This ensures that for $\epsilon' = \frac{1}{poly(n)}$, there exist polynomials f_1, f_2, f_3 such that $\epsilon \leq 1/f_1(n)$, $\delta \leq 1/f_2(n)$ and $m \geq f_3(n)$ satisfy the desired accuracy.

Appendix E

On poly-sparsity and anti-concentration

In this section we will prove that poly-sparsity and anti-concentration cannot simultaneously be satisfied by any family of quantum circuits. This result is proven in Thm. 18.

The condition of poly-sparsity forces the output distributions over exponentially many outcomes to concentrate on polynomially many outcomes. Alternatively, the property of anti-concentration forces the probabilities of observing any particular outcome, over random choices of circuits, to be low. Intuitively these properties do appear to oppose each other. However, since these properties are statements with respect to different probability spaces, we must first translate each property into a statement about a common probability space and with a common measure. This is done for anti-concentration and poly-sparsity in Lem. 16 and 17 respectively. We then state and prove our main claim in Thm. 18.

First let us restate the relevant definitions is some detail.

Definition 10. *(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 ϵ-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 ϵ-approximately t-sparse whenever $t \geq P(k_a/\epsilon)$.

Definition 11. *(anti-concentration) Let $\mathcal{C}$ be a family of quantum circuits with $\mathbb{P}$ its associated family of probability distributions. For all $n \in \mathbb{N}$ let σ_n be a probability measure over $\mathcal{A}^n$. We say that $\mathcal{C}$ anti-concentrates with respect to the set of measures $\Sigma := \{\sigma_n\}_{n\in\mathbb{N}}$ iff $\forall n \in \mathbb{N}$, $\forall x \in \{0,1\}^n$ and $\forall \alpha \in (0,1)$:*

$$\Pr_{a\in\mathcal{A}^n}\left(\mathcal{P}_a(x) \geq \frac{\alpha}{2^n}\right) > \frac{(1-\alpha)^2}{2} \tag{E.1}$$

where the probability is with respect to the measure σ_n.

Lemma 16. *For each $n \in \mathbb{N}$, let σ_n be a probability measure over $\mathcal{A}^n$, let ν_n be any probability measure over over $\{0,1\}^n$ and let τ_n be a probability measure over $\mathcal{A}^n \times \{0,1\}^n$ defined as the* *product measure $\sigma_n \times \nu_n$. Then $\mathcal{C}$ anti-concentrates with respect to $\{ \sigma_n \}_{n\in\mathbb{N}}$ implies that $\forall n \in \mathbb{N}$ and $\forall \alpha \in (0,1)$:*

$$\Pr_{(a,x)\in\mathcal{A}^n\times\{0,1\}^n}\left(\mathcal{P}_a(x) \geq \frac{\alpha}{2^n}\right) > \frac{(1-\alpha)^2}{2} \tag{E.2}$$

where the probability is taken with respect to τ_n.

Proof. For each $n \in \mathbb{N}$, we first define the sets $S_x := \{ (a,x) \in \mathcal{A}^n \times \{0,1\}^n \mid \mathcal{P}_a(x) \geq \frac{\alpha}{2^n} \}$ and $S'_x := \{ a \in \mathcal{A}^n \mid \mathcal{P}_a(x) \geq \frac{\alpha}{2^n} \}$. Let $S := \bigcup\limits_{x\in\{0,1\}^n} S_x$.

By the definition of anti-concentration, we have that $\forall n \in \mathbb{N}$, $\forall x \in \{0,1\}^n$ and $\forall \alpha \in (0,1)$:

$$\sigma_n(S_x) > \frac{(1-\alpha)^2}{2}.$$

Let us fix ν_n and note that:

$$\sum_{x\in\{0,1\}^n} \nu_n(\{ x \}) \times \sigma_n(S'_x) > \sum_{x\in\{0,1\}^n} \nu_n(\{ x \}) \times \frac{(1-\alpha)^2}{2} \tag{E.3}$$

The LHS of Eq. (E.3) simplifies as follows:

$$\begin{aligned}\sum_{x\in\{0,1\}^n} \nu_n(\{ x \}) \times \sigma_n(S'_x) &= \sum_{x\in\{0,1\}^n} \tau_n(S_x) \\ &= \tau_n(S).\end{aligned}$$

By the fact that ν_n is a probability measure, the RHS simplifies to $\frac{(1-\alpha)^2}{2}$ thus proving the claim. □

Lemma 17. *For each $n \in \mathbb{N}$, let σ_n be any probability measure over $\mathcal{A}^n$, let u_n be the uniform probability measure over over $\{0,1\}^n$ and let τ_n be a probability measure over $\mathcal{A}^n \times \{0,1\}^n$ defined as the product measure $\sigma_n \times \mu_n$. Then $\mathcal{C}$ is poly-sparse implies that $\forall \beta, \epsilon \in (0,1]$, $\exists n_0 \in \mathbb{N}$ such that $\forall n \geq n_0$ and $\forall \gamma > 0$:*

$$\Pr_{(a,x)\in\mathcal{A}^n\times\{0,1\}^n}\left(\mathcal{P}_a(x) \geq \frac{\epsilon}{2^n\gamma(1-\beta)}\right) \leq \gamma + \beta \tag{E.4}$$

where the probability is taken with respect to τ_n.

Proof. For any family of quantum circuits $\mathcal{C}$, $\forall n \in \mathbb{N}$ and for each $a \in \mathcal{A}^n$, a minimal function $t_a : [0,1] \to \mathbb{N}$ can be uniquely defined such that $\forall \epsilon \in [0,1]$, the probability of observing an outcome to be one of the $t_a(\epsilon)$ most likely outcomes (when circuit $c_a \in \mathcal{C}$ is run) is $\geq 1-\epsilon$. Poly-sparsity implies that there exists a polynomial P such that $\forall \epsilon \in (0,1]$, $\forall n \in \mathbb{N}$, $\forall a \in \mathcal{A}^n$, $t_a(\epsilon) \leq P(n/\epsilon)$.

We apply Markov's inequality, which states that for R a non-negative random variable and

$\gamma > 0$:

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

For any fixed $n \in \mathbb{N}$, $a \in \mathcal{A}^n$, let us consider uniformly randomly sampling from one of the $2^n - t_a(\epsilon)$ least likely outcomes. In this case:

$$\begin{aligned}\mathbb{E}[\mathcal{P}_a(x)] &= \frac{\epsilon^+}{2^n - t_a(\epsilon)} \\ &\leq \frac{\epsilon}{2^n - t_a(\epsilon^+)} \\ &= \frac{\epsilon}{2^n - t_a(\epsilon) - 1}\end{aligned}$$

where $\epsilon^{\pm}$ are the two extremal points of the interval $V := [\epsilon^-, \epsilon^+)$ containing ϵ such that $\forall \kappa \in V$ $t_a(\kappa) = t_a(\epsilon)$.

This implies that $\forall \epsilon \in (0,1]$, $\forall n \in \mathbb{N}$, $\forall a \in \mathcal{A}^n$, $\forall \gamma > 0$, if we uniformly randomly sample from one of the $2^n - t_a(\epsilon)$ least likely outcomes, then:

$$\begin{aligned}\Pr_{x \in \{0,1\}^n}\left(\mathcal{P}_a(x) \geq \frac{\mathbb{E}[\mathcal{P}_a(x)]}{\gamma}\right) &\leq \Pr_{x \in \{0,1\}^n}\left(\mathcal{P}_a(x) \geq \frac{\epsilon}{\gamma(2^n - t_a(\epsilon) - 1)}\right) \\ &\leq \gamma\end{aligned}$$

Hence, for x uniformly sampled over *all* bit-strings:

$$\Pr_{x \in \{0,1\}^n}\left(\mathcal{P}_a(x) \geq \frac{\epsilon}{\gamma(2^n - t_a(\epsilon) - 1)}\right) \leq \gamma + \frac{t_a(\epsilon)}{2^n}$$

i.e.

$$\Pr_{x \in \{0,1\}^n}\left(\mathcal{P}_a(x) \geq \eta\right) \leq c \tag{E.5}$$

where $\eta := \frac{\epsilon 2^{-n}}{\gamma(1 - t_a(\epsilon)2^{-n} - 2^{-n})}$, $c := \gamma + \frac{t_a(\epsilon)}{2^n}$ and the probability is over the uniform measure u_n over $\{0,1\}^n$.

Let us note that by the fact that poly-sparsity requires that there is a polynomial P such that $t_a(\epsilon) \leq P(n/\epsilon)$, we have that $\forall \beta, \epsilon \in (0,1]$, $\exists n_0 \in \mathbb{N}$ such that $\forall n \geq n_0$, $\forall a \in \mathcal{A}^n$ and $\forall \gamma > 0$:

$$\beta \geq \frac{t_a(\epsilon) + 1}{2^n}$$

implying that $\eta \leq \frac{\epsilon 2^{-n}}{\gamma(1-\beta)}$ and $c < \gamma + \beta$.

For all $n \in \mathbb{N}$, we define the sets:

$$T_a := \left\{ (a, x) \in \mathcal{A}^n \times \{0,1\}^n \ \middle|\ \mathcal{P}_a(x) \geq \eta \right\},$$

$$T'_a := \{\, x \in \{0,1\}^n \quad | \; \mathcal{P}_a(x) \geq \eta \,\},$$

and

$$T := \underset{a \in \mathcal{A}^n}{\cup} T_a.$$

We now note that Eq. (E.5) can be rewritten as:

$$u_n(T'_a) \leq c.$$

For any fixed sequence of measures σ_n over $\mathcal{A}^n$, we can define $\tau_n = \sigma_n \times u_n$ to get:

$$\sum_{a \in \mathcal{A}^n} \sigma_n(a) \times u_n(T'_a) \leq \sum_{a \in \mathcal{A}^n} \sigma_n(a) \times c. \tag{E.6}$$

The LHS of this equation can be simplified as follows:

$$\begin{aligned}\sum_{a \in \mathcal{A}^n} \sigma_n(a) \times u_n(T'_a) &= \sum_{a \in \mathcal{A}^n} \tau_n(T_a) \\ &= \tau_n(T).\end{aligned}$$

By the fact that σ_n is a probability measure, the RHS of Eq. (E.6) simplifies to c thus proving the claim. □

Theorem 18. *For each $n \in \mathbb{N}$ let σ_n be a measures over $\mathcal{A}^n$ and let $\mathcal{C} = \underset{n \in \mathbb{N}}{\cup} \{\, c_a \,\}_{a \in \mathcal{A}^n}$ be a family of quantum circuits such that $\mathcal{C}$ anti-concentrates with respect to $\Sigma = \{\, \sigma_n \,\}_{n \in \mathbb{N}}$. Then $\mathcal{C}$ is not poly-sparse.*

Proof. We will show that the two conditions together give rise to a contradiction and hence are inconsistent. We apply Lem. 16 and set $\alpha = 1/8$ and for each $n \in \mathbb{N}$, $\nu_n = u_n$, the uniform measure over $\{0,1\}^n$ giving:

$$\Pr_{(a,x) \in \mathcal{A}^n \times \{0,1\}^n} \left(\mathcal{P}_a(x) \geq \frac{1}{2^n \times 2} \right) > \frac{1}{8}. \tag{E.7}$$

We apply Lem. 17 and set $\beta = \gamma = 1/16$ and $\epsilon = \frac{1}{64}\left(1 - \frac{1}{16}\right)$ giving:

$$\Pr_{(a,x) \in \mathcal{A}^n \times \{0,1\}^n} \left(\mathcal{P}_a(x) \geq \frac{1}{2^n \times 4} \right) \leq \frac{1}{8} \tag{E.8}$$

where both Eq. (E.7) and (E.8) are with respect to the same measure τ_n thus implying a contradiction.

□

Thm. 18 establishes that poly-sparsity and anti-concentration are mutually exclusive properties. However, these properties can nonetheless jointly fail to be satisfied. For example, we can take an infinite family of circuits (growing faster than 2^n) which is poly-sparse then for each n append a single circuit with output distribution that is uniform. This change ensures that the family now breaks poly-sparsity but is insufficient for anti-concentration to be instated. In addition, if we assume that the family admits a poly-box then we notice that by Thm. 10 this family was ϵ-simulable before the change but after the change no longer satisfies the requirements for the application of Thm. 10 despite the fact that we have only added a sequence of uniform distributions which are ϵ-simulable. A clean mathematical characterization of when poly-sparsity and anti-concentration can jointly fail may help inform how to best resolve this undesirable situation.

Bibliography

[1] H. Pashayan, J. J. Wallman, and S. D. Bartlett, "Estimating outcome probabilities of quantum circuits using quasiprobabilities," *Physical Review Letters*, vol. 115, no. 7, p. 070501, 2015.

[2] S. Bravyi and D. Gosset, "Improved classical simulation of quantum circuits dominated by Clifford gates," *Physical Review Letters*, vol. 116, no. 25, p. 250501, 2016.

[3] H. Pashayan, S. D. Bartlett, and D. Gross, "From estimation of quantum probabilities to simulation of quantum circuits," *arXiv preprint arXiv:1712.02806*, 2017.

[4] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937.

[5] K. Gödel, *On Undecidable Propositions of Formal Mathematics Systems.* Institute for Advanced Study, 1934.

[6] A. Church, "An unsolvable problem of elementary number theory," *American Journal of Mathematics*, vol. 58, no. 2, pp. 345–363, 1936.

[7] R. P. Feynman, "Simulating Physics with Computers," *Int. J. Theor. Phys.*, vol. 21, no. 6/7, pp. 467–488, 1982.

[8] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 439, no. 1907, pp. 553–558, 1992.

[9] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.

[10] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), pp. 212–219, ACM, 1996.

[11] D. Aharonov and A. Ta-Shma, "Adiabatic quantum state generation and statistical zero knowledge," in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, (New York, NY, USA), pp. 20–29, ACM, 2003.

[12] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, "Quantum fingerprinting," *Physical Review Letters*, vol. 87, p. 167902, 2001.

[13] S. Aaronson and A. Arkhipov, "The computational complexity of linear optics," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 333–342, ACM, 2011.

[14] R. J. Lipton, "New directions in testing.," *Distributed computing and cryptography*, vol. 2, pp. 191–202, 1989.

[15] D. Shepherd and M. J. Bremner, "Temporally unstructured quantum computation," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 465, pp. 1413–1439, The Royal Society, 2009.

[16] M. J. Bremner, R. Jozsa, and D. J. Shepherd, "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 2010.

[17] D. Gottesman, "The Heisenberg representation of quantum computers," *arXiv quant-ph/9807006*, 1998.

[18] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, no. 5, p. 052328, 2004.

[19] L. G. Valiant, "Quantum circuits that can be simulated classically in polynomial time," *SIAM Journal on Computing*, vol. 31, no. 4, pp. 1229–1254, 2002.

[20] B. M. Terhal and D. P. DiVincenzo, "Classical simulation of noninteracting-fermion quantum circuits," *Physical Review A*, vol. 65, no. 3, p. 032325, 2002.

[21] R. Jozsa and A. Miyake, "Matchgates and classical simulation of quantum circuits," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 464, no. 2100, pp. 3089–3106, 2008.

[22] V. Veitch, C. Ferrie, D. Gross, and J. Emerson, "Negative quasi-probability as a resource for quantum computation," *New Journal of Physics*, vol. 14, no. 11, p. 113011, 2012.

[23] A. Mari and J. Eisert, "Positive Wigner functions render classical simulation of quantum computation efficient," *Physical Review Letters*, vol. 109, no. 23, p. 230503, 2012.

[24] V. Veitch, N. Wiebe, C. Ferrie, and J. Emerson, "Efficient simulation scheme for a class of quantum optics experiments with non-negative Wigner representation," *New Journal of Physics*, vol. 15, no. 1, p. 013037, 2013.

[25] R. Jozsa and N. Linden, "On the role of entanglement in quantum-computational speed-up," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 459, no. 2036, pp. 2011–2032, 2003.

[26] B. M. Terhal and D. P. DiVincenzo, “Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games,” *Quantum Information & Computation*, vol. 4, no. 2, pp. 134–145, 2004.

[27] T. Morimae, K. Fujii, and H. Nishimura, “Power of one nonclean qubit,” *Physical Review A*, vol. 95, no. 4, p. 042336, 2017.

[28] E. T. Campbell, H. Anwar, and D. E. Browne, “Magic-state distillation in all prime dimensions using quantum Reed-Muller codes,” *Physical Review X*, vol. 2, p. 041021, 2012.

[29] V. Veitch, S. A. H. Mousavian, D. Gottesman, and J. Emerson, “The resource theory of stabilizer quantum computation,” *New Journal of Physics*, vol. 16, no. 1, p. 013009, 2014.

[30] M. Howard, J. Wallman, V. Veitch, and J. Emerson, “Contextuality supplies the ‘magic’ for quantum computation,” *Nature*, vol. 510, no. 7505, p. 351, 2014.

[31] F. Siyouri, M. El Baz, and Y. Hassouni, “The negativity of Wigner function as a measure of quantum correlations,” *Quantum Information Processing*, vol. 15, no. 10, pp. 4237–4252, 2016.

[32] E. Tang, “A quantum-inspired classical algorithm for recommendation systems,” *arXiv preprint arXiv:1807.04271*, 2018.

[33] R. S. Bennink, E. M. Ferragut, T. S. Humble, J. A. Laska, J. J. Nutaro, M. G. Pleszkoch, and R. C. Pooser, “Unbiased simulation of near-Clifford quantum circuits,” *Physical Review A*, vol. 95, no. 6, p. 062337, 2017.

[34] M. Howard and E. Campbell, “Application of a resource theory for magic states to fault-tolerant quantum computing,” *Physical Review Letters*, vol. 118, no. 9, p. 090501, 2017.

[35] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” *Physical Review Letters*, vol. 119, no. 18, p. 180509, 2017.

[36] S. Bravyi, G. Smith, and J. A. Smolin, “Trading classical and quantum computational resources,” *Physical Review X*, vol. 6, no. 2, p. 021043, 2016.

[37] P. Rall, D. Liang, J. Cook, and W. Kretschmer, “Simulation of qubit quantum circuits via Pauli propagation,” *Physical Review A*, vol. 99, p. 062337, 2019.

[38] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, “Simulation of quantum circuits by low-rank stabilizer decompositions,” *arXiv preprint arXiv:1808.00128*, 2018.

[39] M. Yoganathan, R. Jozsa, and S. Strelchuk, “Quantum advantage of unitary clifford circuits with magic state inputs,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 475, no. 2225, p. 20180427, 2019.

[40] C. M. Dawson, A. P. Hines, D. Mortimer, H. L. Haselgrove, M. A. Nielsen, and T. J. Osborne, "Quantum computing and polynomial equations over the finite field Z2," *Quantum Info. Comput.*, vol. 5, no. 2, pp. 102–112, 2005.

[41] A. Montanaro, "Quantum circuits and low-degree polynomials over," *Journal of Physics A: Mathematical and Theoretical*, vol. 50, no. 8, p. 084002, 2017.

[42] I. Markov and Y. Shi, "Simulating quantum computation by contracting tensor networks," *SIAM Journal on Computing*, vol. 38, no. 3, pp. 963–981, 2008.

[43] D. Stahlke, "Quantum interference as a resource for quantum speedup," *Physical Review A*, vol. 90, p. 022302, 2014.

[44] V. c. v. Havlí*c*ek and S. Strelchuk, "Quantum schur sampling circuits can be strongly simulated," *Physical Review Letters*, vol. 121, p. 060505, 2018.

[45] K. Fujii and T. Morimae, "Commuting quantum circuits and complexity of ising partition functions," *New Journal of Physics*, vol. 19, no. 3, p. 033003, 2017.

[46] L. A. Goldberg and H. Guo, "The complexity of approximating complex-valued Ising and Tutte partition functions," *computational complexity*, vol. 26, no. 4, pp. 765–833, 2017.

[47] C. Ferrie and J. Emerson, "Frame representations of quantum mechanics and the necessity of negativity in quasi-probability representations," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 35, p. 352001, 2008.

[48] C. Ferrie and J. Emerson, "Framed Hilbert space: hanging the quasi-probability pictures of quantum theory," *New Journal of Physics*, vol. 11, no. 6, p. 063040, 2009.

[49] D. Gross, "Hudson's theorem for finite-dimensional quantum systems," *Journal of Mathematical Physics*, vol. 47, no. 12, p. 122107, 2006.

[50] K. Gibbons, M. Hoffman, and W. Wootters, "Discrete phase space based on finite fields," *Physical Review A*, vol. 70, 2004.

[51] D. M. Appleby, I. Bengtsson, and S. Chaturvedi, "Spectra of phase point operators in odd prime dimensions and the extended Clifford group," *Journal of Mathematical Physics*, vol. 49, no. 1, p. 012102, 2008.

[52] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[53] T. P. Hayes, "A large-deviation inequality for vector-valued martingales," *Combinatorics, Probability and Computing*, 2005.

[54] E. T. Campbell, Y. Ouyang, H. Pashayan, B. Regula, and J. Seddon, "Classical simulation and state conversion bounds in the dyadic frame," Unpublished notes.

[55] H. Qassim, J. J. Wallman, and J. Emerson, "Clifford recompilation for faster classical simulation of quantum circuits," *Quantum*, vol. 3, p. 170, 2019.

[56] M. Heinrich and D. Gross, "Robustness of Magic and Symmetries of the Stabiliser Polytope," *Quantum*, vol. 3, p. 132, 2019.

[57] R. Jozsa and M. Van Den Nest, "Classical simulation complexity of extended Clifford circuits," *Quantum Information & Computation*, vol. 14, no. 7&8, pp. 633–648, 2014.

[58] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[59] A. W. Harrow and A. Montanaro, "Quantum computational supremacy," *Nature*, vol. 549, no. 7671, p. 203, 2017.

[60] S. T. Flammia and Y.-K. Liu, "Direct fidelity estimation from few Pauli measurements," *Physical Review Letters*, vol. 106, p. 230501, 2011.

[61] M. Kliesch, "Lecture notes: Characterization, verification, and validation of quantum systems," 2019.

[62] X. Zhou, D. W. Leung, and I. L. Chuang, "Methodology for quantum logic gate construction," *Physical Review A*, vol. 62, no. 5, p. 052316, 2000.

[63] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* New York, NY, USA: Cambridge University Press, 10th ed., 2011.

[64] H. J. García, I. L. Markov, and A. W. Cross, "On the geometry of stabilizer states," *Quant. Inf. and Comp.*, vol. 14, no. 7-8, pp. 683–720, 2014.

[65] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, p. 150502, 2009.

[66] A. Gilyén, S. Lloyd, and E. Tang, "Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension," *arXiv preprint arXiv:1811.04909*, 2018.

[67] M. Hebenstreit, R. Jozsa, B. Kraus, S. Strelchuk, and M. Yoganathan, "All pure fermionic non-gaussian states are magic states for matchgate computations," *Physical Review Letters*, vol. 123, p. 080503, 2019.

[68] S. D. Bartlett, B. C. Sanders, S. L. Braunstein, and K. Nemoto, "Efficient classical simulation of continuous variable quantum information processes," in *Quantum Information with Continuous Variables*, pp. 47–55, Springer, 2002.

[69] L. Gurvits, "On the complexity of mixed discriminants and related problems," in *International Symposium on Mathematical Foundations of Computer Science*, pp. 447–458, Springer, 2005.

[70] M. J. Bremner, A. Montanaro, and D. J. Shepherd, "Average-case complexity versus approximate simulation of commuting quantum computations," *Physical Review Letters*, vol. 117, no. 8, p. 080501, 2016.

[71] X. Gao, S.-T. Wang, and L.-M. Duan, "Quantum supremacy for simulating a translation-invariant Ising spin model," *Physical Review Letters*, vol. 118, no. 4, p. 040502, 2017.

[72] J. Bermejo-Vega, D. Hangleiter, M. Schwarz, R. Raussendorf, and J. Eisert, "Architectures for quantum simulation showing a quantum speedup," *Physical Review X*, vol. 8, p. 021010, 2018.

[73] B. Fefferman and C. Umans, "The power of quantum fourier sampling," *arXiv preprint arXiv:1507.05592*, 2015.

[74] T. Morimae, K. Fujii, and J. F. Fitzsimons, "Hardness of classically simulating the one-clean-qubit model," *Physical Review Letters*, vol. 112, no. 13, p. 130502, 2014.

[75] 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," *Nature Physics*, p. 1, 2018.

[76] A. Bouland, J. F. Fitzsimons, and D. E. Koh, "Complexity Classification of Conjugated Clifford Circuits," in *33rd Computational Complexity Conference (CCC 2018)* (R. A. Servedio, ed.), vol. 102 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (Dagstuhl, Germany), pp. 21:1–21:25, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[77] D. J. Brod, "Efficient classical simulation of matchgate circuits with generalized inputs and measurements," *Physical Review A*, vol. 93, no. 6, p. 062332, 2016.

[78] M. J. Bremner, A. Montanaro, and D. J. Shepherd, "Achieving quantum supremacy with sparse and noisy commuting quantum computations," *Quantum*, vol. 1, p. 8, 2017.

[79] M. Oszmaniec and D. J. Brod, "Classical simulation of photonic linear optics with lost particles," *New Journal of Physics*, vol. 20, no. 9, p. 092002, 2018.

[80] M. Nest, "Efficient classical simulations of quantum Fourier transforms and normalizer circuits over Abelian groups," *arXiv preprint arXiv:1201.4867*, 2012.

[81] J. Bermejo-Vega and M. Van Den Nest, "Classical simulations of Abelian-group normalizer circuits with intermediate measurements," *Quantum Information & Computation*, vol. 14, no. 3-4, pp. 181–216, 2014.

[82] M. Schwarz and M. V. d. Nest, "Simulating quantum circuits with sparse output distributions," *arXiv preprint arXiv:1310.6749*, 2013.

[83] M. Van Den Nest, "Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond," *Quantum Info. Comput.*, vol. 10, no. 3, pp. 258–271, 2010.

[84] T. Morimae, "Hardness of classically sampling the one-clean-qubit model with constant total variation distance error," *Physical Review A*, vol. 96, no. 4, p. 040302, 2017.

[85] S. Aaronson, "The equivalence of sampling and searching," *Theory of Computing Systems*, vol. 55, no. 2, pp. 281–298, 2014.

[86] M. Van Den Nest, "Simulating quantum computers with probabilistic methods," *Quantum Info. Comput.*, vol. 11, no. 9-10, pp. 784–812, 2011.

[87] D. Shepherd, "Binary matroids and quantum probability distributions," *arXiv preprint arXiv:1005.1744*, 2010.

[88] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal clifford gates and noisy ancillas," *Physical Review A*, vol. 71, p. 022316, 2005.

[89] G. Kuperberg, "How hard is it to approximate the Jones polynomial?," *Theory of Computing*, vol. 11, no. 6, pp. 183–219, 2015.

[90] D. J. Shepherd, "Quantum complexity: restrictions on algorithms and architectures," *arXiv preprint arXiv:1005.1425*, 2010.

[91] We thank an anonymous QIP referee for bringing this connection to our attention.

[92] D. Vertigan, "Bicycle dimension and special points of the Tutte polynomial," *Journal of Combinatorial Theory, Series B*, vol. 74, no. 2, pp. 378–396, 1998.

[93] We thank an anonymous QIP referee for pointing out a simple example that uses encoding to smooth out Born rule probabilities and marginals.

[94] R. Renner and S. Wolf, "Smooth Rényi entropy and applications," in *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, pp. 233–, 2004.

[95] L. Stockmeyer, "The complexity of approximate counting," in *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pp. 118–126, ACM, 1983.

[96] S. Toda, "PP is as hard as the polynomial-time hierarchy," *SIAM Journal on Computing*, vol. 20, no. 5, pp. 865–877, 1991.

[97] S. Aaronson and T. Hance, "Generalizing and derandomizing Gurvits's approximation algorithm for the permanent," *Quantum Information & Computation*, vol. 14, no. 7&8, pp. 541–559, 2014.