A simple protocol for fault tolerant verification of quantum computation

Alexandru Gheorghiu,[1, 2] Matty J. Hoban,[3, 4] and Elham Kashefi[1, 5]

[1] *School of Informatics, University of Edinburgh,*
10 Crichton Street, Edinburgh EH8 9AB, UK
[2] *Department of Computing and Mathematical Sciences,*
California Institute of Technology, Pasadena, USA
[3] *Clarendon Laboratory, Department of Physics,*
University of Oxford, Parks Road, OX1 3PJ, UK
[4] *Department of Computing, Goldsmiths, University of London,*
New Cross, London SE14 6NW, United Kingdom
[5] *CNRS LIP6, Université Pierre et Marie Curie, Paris, France*

With experimental quantum computing technologies now in their infancy, the search for efficient means of testing the correctness of these quantum computations is becoming more pressing. An approach to the verification of quantum computation within the framework of interactive proofs has been fruitful for addressing this problem. Specifically, an untrusted agent (prover) alleging to perform quantum computations can have his claims verified by another agent (verifier) who only has access to classical computation and a small quantum device for preparing or measuring single qubits. However, when this quantum device is prone to errors, verification becomes challenging and often existing protocols address this by adding extra assumptions, such as requiring the noise in the device to be uncorrelated with the noise on the prover's devices. In this paper, we present a simple protocol for verifying quantum computations, in the presence of noisy devices, with no extra assumptions. This protocol is based on *post hoc* techniques for verification, which allow for the prover to know the desired quantum computation and its input. We also perform a simulation of the protocol, for a one-qubit computation, and find the error thresholds when using the qubit repetition code as well as the Steane code.

I. INTRODUCTION

There is now substantial evidence that quantum computers cannot be simulated efficiently by their classical counterparts. Shor's factoring algorithm is one example where an efficient (polynomial time) solution to a problem can be found with a quantum computer, but the best classical algorithm that we know runs in superpolynomial time (in the worst case) [1]. While the inability to be efficiently classically simulated can be of great use in computing, it does raise other problems. In particular, how can we check that the device is producing the correct answer if it is hard to simulate with a classical computer? For the case of factoring, we can just multiply the factors, but for other problems (such as simulating a quantum system) there is no *a priori* classically efficient means of verifying whether a quantum computation was carried out [2].

To be a bit more explicit, the scenario is that of a *verifier* that can do probabilistic classical computation, and an untrusted *prover* who is limited to universal quantum computation. The verifier wishes to use the prover's quantum computer, but might not be able to trust the prover's output, and so the verifier will perform some sort of verification. In full generality, the prover and the verifier can exchange multiple rounds of classical communication with each other (as the verifier has no quantum capabilities). Both parties' computations have a running time that is, at most, a polynomial in the size of the input to the computation (hence efficient by standard notions in computational complexity). The verifier would like to be able to perform any efficient quantum computation by delegating the task to the untrusted prover. Clearly, if the prover is honest and actually has a quantum computer, they can perform the task for the verifier. On the other hand, if the prover is dishonest then they could lie or deviate in any manner they deem fit. The goal of verification of quantum computation is to catch (with high probability) when a prover is being dishonest and reject his output.

There has been a lot of progress in the development of verification protocols (see [3] for a survey paper). It is still an open problem whether verification is possible in this setting, with no extra computational assumptions [4]. Two main modifications of the classical client, single server setting have been considered: give the verifier some quantum device and allow some quantum communication between the prover and verifier, or introduce multiple non-communicating quantum provers that share entanglement. In the former approach – the approach we take in this paper – this quantum device could prepare particular quantum states or make measurements in a particular basis, and this quantum process becomes an integral part of the prover's quantum computation. In this approach, existing protocols typically assume an ideal setting in which the only "errors" that can occur are a result of malicious behaviour of the prover. But realistically, quantum devices are highly susceptible to noise, and so a verifier could introduce noise into a computation implemented by an honest prover. Reducing this noise and controlling it is one of the great challenges in developing scalable quantum computers. The *threshold theorem* shows that as long the error rate per quantum gate is below a constant threshold, it is always possible to perform a *fault tolerant quantum computation* with only a polylogarithmic increase in overhead [5–7].

We would like to avoid the verifier's quantum device becoming too powerful, since even being polylogarithmic in the size of the input to the computation would be too powerful [8]. Therefore, we want to restrict it to having a quantum register that is constant-size. Even if the errors of the verifier's device can be suppressed, it still needs to be proven that this is not detrimental for the verification of a quantum computation. To wit, we want that a malicious prover does not exploit these errors in order to successfully trick the verifier into accepting incorrect results.

For protocols in which the verifier is fully classical, fault tolerance is not a concern since one can assume that the provers are performing their quantum operations on top of a quantum error correcting code. Since provers are assumed to have universal quantum computing power, we naturally have to assume that they are capable of fault tolerant quantum computation between themselves. We emphasise that discussions about fault tolerance only make sense in the setting in which the verifier possesses a quantum device.

Thus, we are faced with the following problem:

Problem statement: *Can a verifier with a constant-size and imperfect quantum device verifiably delegate a quantum computation to a single prover?*

We show that this is indeed possible. Furthermore, it is possible even if the verifier's device is an imperfect *single-qubit* measurement device. Our approach is based on that of *post hoc verification* [9–11], where a prover sends quantum systems to a verifier that should be the ground state of a Hamiltonian. This ground state encodes the desired quantum computation and can be used to "read off" the outcome of that computation. If the verifier can indeed certify that this is the ground state, then the computation is verified. In our protocol we encode the qubits of this ground state into a *logical ground state* where each qubit of the original state is encoded into a larger number of physical qubits via a quantum error correction code. This logical state is then the ground state of a logical Hamiltonian described by the quantum computation. In the protocol, the physical qubits in this logical state are then measured one at a time, and appropriate classical corrections are made on the outcomes of these measurements in post-processing if errors are detected. An honest prover's probability of successful computation will be boosted by this error correction, but importantly we can still verify if the logical ground state was indeed prepared by the prover.

Finally, we consider a simple example of this protocol in the honest prover scenario. That is, using the repetition code and the Steane code [12], we can simulate and characterise the protocol's behaviour under bit-flip errors and depolarizing noise.

Paper outline — In Section II we give some basic complexity theoretic notions to formalize what we mean by verifying efficient quantum computations. We also outline post hoc quantum verification, which is the basis for our approach. Next, in Section III we give our protocol for fault tolerant verification of quantum computation, and also prove its correctness; we also describe our simulation of the protocol, with various degrees of noise and outline the obtained results. We then conclude, in Section IV, with some discussions and open problems.

Let us first comment on approaches that have also addressed the aforementioned problem of fault tolerant verification.

Related work — For protocols in which the verifier has a small quantum device, the question of fault tolerance has been addressed in [13–17]. In [13–16] the authors proposed protocols in which a classical client possessing either a single qubit preparation or measurement device, susceptible to noise, could verifiably delegate quantum computations to a prover. All these protocols are *computationally blind*, meaning that the delegated computation is kept secret from the prover. We will return to this issue in detail in Section IV. Moreover, blindness is *required* for achieving verifiability. However, this requirement of blindness introduces new difficulties when considering fault tolerant computation. To circumvent these difficulties, extra (potentially unrealistic) assumptions were made about the noise, which rule out the possibility of the prover utilising the noise to deceive the verifier. A discussion of the general difficulty in realizing a verifiable, blind, fault tolerant protocol is provided in [17].

II. PRELIMINARIES

A. Complexity theory

Complexity theory classifies computational problems as sets of "yes/no" decision problems that are solvable by a particular model of computation, under certain constraints. Decision problems are modeled as sets of binary strings, known as *languages*. The input to the problem is a bit string and the output is yes or no, depending on whether the string belongs to the language or not. The primary class that is of interest to us, is that of decision problems decidable efficiently by a quantum computer, which is denoted BQP. By "efficiently" we always mean in a number of time steps that scales as some polynomial in the size of the input to the problem. For completeness we reproduce the definition of this class.

Definition 1. *A language $L \subseteq \{0,1\}^*$ belongs to* BQP *iff there exist a polynomial p, and a uniform quantum circuit family $\{C_n\}_n$, such that for any $x \in \{0,1\}^n$ the following is true:*

- *when $x \in L$, $C_n(x)$ accepts with probability at least a, and*
- *when $x \notin L$, $C_n(x)$ accepts with probability at most b,*

where $a - b \geq 1/p(n)$ and $|C_n| \leq p(n)$.

If we replace quantum circuits with classical boolean circuits, having access to random bits, we obtain the class BPP, of problems that can be decided efficiently on a classical computer. We will frequently refer to machines that can solve either BPP problems or BQP problems as BPP machines or BQP machines, respectively.

Another class of interest is MA which consists of decisions problems for which the "yes" instances can be checked by a BPP machine, when given access to a polynomial-sized bitstring known as a *proof*. More formally, MA is defined as follows:

Definition 2. *A language $L \subseteq \{0,1\}^*$ belongs to* MA *iff there exist a polynomial p, and a* BPP *machine $\mathcal{V}$ (known as verifier), such that for any $x \in \{0,1\}^n$ the following is true:*

- *when $x \in L$, there exists a string $w \in \{0,1\}^{\leq p(n)}$ such that $\mathcal{V}(x,w)$ accepts with probability at least a, and*
- *when $x \notin L$, for all strings $w \in \{0,1\}^{\leq p(n)}$, $\mathcal{V}(x,w)$ accepts with probability at most b,*

where $a - b \geq 1/p(n)$.

Essentially, one can view problems in MA as those for which a computationally powerful prover can convince a BPP verifier that the answer is "yes", with high probability, by providing a proof string that the verifier can check. There is a quantum analogue of this known as QMA, in which the proof string is a quantum state. Specifically:

Definition 3. *A language $L \subseteq \{0,1\}^*$ belongs to* QMA *iff there exist a polynomial p, and a* BQP *machine $\mathcal{V}$ (known as verifier), such that for any $x \in \{0,1\}^n$ the following is true:*

- *when $x \in L$, there exists a quantum state $|\psi\rangle$ having at most $p(n)$-many qubits, such that $\mathcal{V}(x, |\psi\rangle)$ accepts with probability at least a, and*
- *when $x \notin L$, for all quantum states $|\psi\rangle$ having at most $p(n)$-many qubits, $\mathcal{V}(x, |\psi\rangle)$ accepts with probability at most b,*

where $a - b \geq 1/p(n)$.

Clearly, BQP $\subseteq$ QMA, since the BQP verifier can simply ignore the proof state from the prover. It is believed that the containment is strict, since, in principle, the prover can produce proof states that cannot be generated by the poly-time quantum verifier. In fact, it was shown in [18] that the "quantum overhead" of the verifier can be reduced to simply performing single-qubit measurements, while maintaining the ability to correctly decide all problems in QMA. This is achieved by instructing the prover to prepare special states that satisfy two properties:

- Any BQP computation can be performed through suitable *single-qubit* measurements of these states.
- Any such state of a given size can be tested through *single-qubit* measurements.

If one adds a further condition, namely that these special states can be prepared by a BQP machine (essentially restricting the prover to BQP), then one obtains a scheme for verifying an arbitrary BQP computation using only single-qubit measurements. This is precisely what Fitzsimons, Hajdušek and Morimae have done in their post hoc verification protocols [9–11]. One could ask whether this merely allows the verifier to check the "yes" instances of BQP problems, however BQP is closed under complement[19] meaning that the "no" instances can also be verified.

B. Post hoc verification

As mentioned, the core idea of post hoc verification is to have a BQP prover (or provers) prepare a quantum proof state that the verifier can check using single-qubit measurements. To explain how the protocol works, we first define a problem known as the *k-local Hamiltonian problem*, which was introduced by Kitaev in [20]. A k-local Hamiltonian, acting on a system of n qubits, is a hermitian operator H that can be expressed as $H = \sum_i H_i$, where each H_i is a hermitian operator which acts non-trivially on at most k qubits. The k-local Hamiltonian problem, for which we have taken the definition from [21], is then the following:

Definition 4 (The k-local Hamiltonian (LH) problem)**.**

- ***Input:*** *$H_1, \ldots, H_m$, a set of m Hermitian matrices each acting on k qubits out of an n-qubit system and satisfying $\|H_i\| \leq 1$. Each matrix entry is specified by poly(n)-many bits. Apart from the H_i we are also given two real numbers, a and b (again, with polynomially many bits of precision) such that $a - b > 1/poly(n)$.*
- ***Output:*** *Is the smallest eigenvalue of $H = H_1 + H_2 + \ldots + H_m$ smaller than b or are all its eigenvalues larger than a?*

Kitaev showed that this problem is complete for the class QMA. In other words, the problem is in QMA and any problem in QMA can be reduced to it, in (classical) polynomial time [20]. The idea is essentially this: for some language $L \in$ QMA, and given a and b, one can construct a k-local Hamiltonian such that, whenever $x \in L$, its smallest eigenvalue is less than b and whenever $x \notin L$, all of its eigenvalues are greater than a. The proof state, $|\psi\rangle$, when $x \in L$, is the eigenstate of H corresponding to its lowest eigenvalue (or a state that is close, in trace distance, to this state), known as the ground state. The verifier receives the state from the prover and measures one of the local terms H_i (which is an observable) on that state. One can prove that this can be done with a polynomial-sized quantum circuit. This yields an estimate for measuring H itself. Therefore, when $x \in L$ and the prover sends $|\psi\rangle$, with high probability the verifier will obtain the corresponding eigenvalue of $|\psi\rangle$ which will be smaller than b.

If the prover is malicious then it would have to convince the verifier to accept when $x \notin L$. However, when this is the case, all the eigenvalues of H will be larger than a and so, no matter what state the prover sends, when the verifier measures the local term H_i it will, with high probability, obtain a value greater than a and will therefore reject. We will refer to $a - b$ as the *promise gap* of the local Hamiltonian.

The constant k in the definition of the k-local Hamiltonian problem is not arbitrary. In the initial construction of Kitaev, k had to be larger than 5 for the problem to be QMA-complete. Subsequent work showed that for $k = 3$ and $k = 4$ the problem remains QMA-complete [22]. Then, in 2006, Kempe et al. proved that even with $k = 2$ the problem is still QMA-complete, whereas it is known that for $k = 1$ the problem can be decided in classical polynomial time [23]. For our purposes, the value of k itself is not important (as long as the problem is QMA-complete). What is important, however, is the *form* of the Hamiltonian. As is mentioned in [9, 11, 24] it is possible to have a QMA-complete k-local Hamiltonian, in which the local terms consist exclusively of tensor products of identities, Pauli X's and Pauli Z's. We will refer to this as an XZ-Hamiltonian.

To be a bit more precise, let us introduce some helpful notation. Consider an n-qubit operator S, which we shall refer to as XZ-term, such that:

$$S = \bigotimes_{j=1}^{n} P_j \tag{1}$$

with $P_j \in \{I, X, Z\}$, where X and Z are the Pauli X and Z operators and I is the identity. Denote $w_X(S)$ as the X-weight of S, representing the total number of j's for which $P_j = X$. Similarly denote $w_Z(S)$ as the Z-weight for S. A k-local Hamiltonian is an XZ-Hamiltonian if it can be expressed as:

$$H = \sum_i a_i S_i \tag{2}$$

where the a_i's are real numbers and the S_i's are XZ-terms having $w_X(S_i) + w_Z(S_i) \leq k$. Essentially, as the name suggests, an XZ-Hamiltonian is one in which each local term has at most k operators, acting on the n qubits, which can be either Pauli X or Z operators.

We can now explain the post hoc protocol of [11]. The protocol relies on the observation that BQP $\subseteq$ QMA. This means that any problem in BQP can be viewed as an instance of the k-local Hamiltonian problem. Therefore, for any language $L \in$ BQP there exists an XZ-Hamiltonian, H, and a polynomial-time quantum verifier which can measure a local term of H on the quantum witness sent by the prover and decide the problem correctly, with high probability. But since the local terms of an XZ-Hamiltonian have at most k operators that are either X or Z, the verifier will essentially have to measure at most a constant number of k qubits. Each qubit is measured in the standard basis ($|0\rangle, |1\rangle$) or the Hadamard basis ($|+\rangle, |-\rangle$).

We now restrict attention to $L \in$ BQP. As mentioned, when $x \in L$, the proof state that the prover should send to the verifier should be close to the ground state of the XZ-Hamiltonian. When $L \in$ BQP, the Hamiltonian can be chosen so that the ground state is close to a particular type of state known as a *Feynman-Kitaev clock state* (also known as *history state*), which can be prepared by a BQP prover [20]. To describe this state, consider a quantum circuit $\mathcal{C} = U_T U_{T-1} ... U_1$, with classical input $|x\rangle$, where $T = poly(|x|)$, for testing whether $x \in L$. Denoting $U_0 = I$, the Feynman-Kitaev state associated to $\mathcal{C}$ and $|x\rangle$ is the following:

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t U_{t-1} ... U_0 |x\rangle |1^t 0^{T-t}\rangle \tag{3}$$

There exists an XZ-Hamiltonian, H, such that when $x \in L$, we have that $\langle\psi|H|\psi\rangle \leq b$, and when $x \notin L$ we have that for any $|\phi\rangle$, $\langle\phi|H|\phi\rangle \geq a$, for some a, b such that $a - b > 1/poly(|x|)$. The exact form of H is not important for understanding the protocol. What is important is that for any $L \in$ BQP, the verifier can efficiently compute the description of H.

The post hoc protocol then works as follows:

1. The verifier computes the terms a_i and S_i of the XZ-Hamiltonian, $H = \sum_i a_i S_i$, corresponding to L and input x. They then send the description of H and x to the prover.

2. The prover responds by preparing the ground state of H (the Feynman-Kitaev state, described above), denoted $|\psi\rangle$, and sends it to the verifier. This constitutes the quantum proof state for the statement that $x \in L$ (if $x \notin L$, the same procedure is performed for the complement of L, denoted L^c, which is also in BQP).

3. The verifier chooses one of the XZ-terms S_i, according to the normalized probability distribution $\{|a_i|\}_i$, and measures it on $|\psi\rangle$. They accept on outcome $-sgn(a_i)$ of the measurement.

The protocol is correct, in that when the prover aims to convince the verifier that $x \in L$ (or $x \in L^c$, respectively) and sends the correct state (for the Hamiltonian corresponding to L or to L^c, respectively), the verifier will accept with probability:

$$p_{acc} \geq \frac{1}{2}\left(1 - \frac{b}{\sum_i |a_i|}\right) \tag{4}$$

Additionally, the protocol is sound in that when the prover aims to convince the verifier that $x \notin L$ (or $x \notin L^c$, respectively), irrespective of the state that the prover sends, the verifier will accept with probability:

$$p_{acc} \leq \frac{1}{2}\left(1 - \frac{a}{\sum_i |a_i|}\right) \tag{5}$$

Since $\sum_i |a_i|$ is a constant and $a - b > 1/poly(|x|)$, the gap between the two probabilities is inverse polynomial in the size of the input.

III. FAULT TOLERANT VERIFICATION OF QUANTUM COMPUTATION

The protocol described above works assuming an ideal setting in which the quantum devices of both the prover and the verifier are perfect. Of course, this is an unrealistic assumption since any implementation of the protocol will be subject to noise stemming from an imperfect isolation of the quantum systems from the environment, and the use of faulty devices. It is straightforward to show that a constant rate of noise on these devices will lead to the failure of the protocol for sufficiently large computations. This is because the gap between acceptance and rejection, defined by $a - b$, is inverse polynomial in the size of the input. As a result of noisy devices, the acceptance threshold is shifted to $a - c$, and the rejection threshold is shifted to $b + c$, where c is some positive constant that depends on the noise rate of the devices. We can see that as long as $c < (a - b)/2$, the verifier can still distinguish reliably between acceptance and rejection. However, it is clear that for a sufficiently long input, we will have that $c \geq (a - b)/2$. At this point, the protocol no longer satisfies the correctness nor the soundness criteria. In fact, this is common to all other verification protocols in the single-prover setting [3]. To address this issue we now give a fault tolerant version of the post hoc protocol that works in the presence of quantum devices subject to local noise having a constant error-rate.

A. The fault tolerant protocol

Our construction is simple: we ask the prover to encode the history state in a CSS (Calderbank-Shor-Steane) error-correcting code [25] and send it to the verifier. The verifier will then perform a *transversal* measurement of the X and Z operators. Transversality results in the logical operators being expressed as tensor products of physical X and Z operators, i.e.:

$$\tilde{X} = \bigotimes_{i=1}^{m} X_i \qquad \tilde{Z} = \bigotimes_{i=1}^{m} Z_i \tag{6}$$

where $\tilde{X}$ and $\tilde{Z}$ are the logical (or encoded) X and Z operators. In effect, the original Hamiltonian is replaced with an encoded Hamiltonian by substituting each XZ-term with its corresponding logical form.

The idea of encoding the history state in a CSS code is briefly mentioned in the independent work of [26], and CSS codes are also considered in [16], though not for post hoc verification.

CSS codes are transversal and this ensures that the verifier needs to perform only single-qubit measurements. We also require an additional property, that is possessed by CSS codes, namely that the outcomes for the transversal measurements (of the X and Z operators) are encoded in a classical error-correcting code. This is because the verifier will not perform any quantum correction on the state sent by the prover. Instead, this state will be measured and the measurement outcomes are classically post-processed.

To clarify, consider the following simple example. Assume that the CSS code is a repetition code in which $|\tilde{0}\rangle = |0\rangle^{\otimes m}$ and $|\tilde{1}\rangle = |1\rangle^{\otimes m}$, for some odd $m > 1$. This code can correct $\lfloor \frac{m}{2} \rfloor$ bit-flip errors. If the verifier wishes to measure the $\tilde{Z}$ observable on an encoded state, they will instead measure the single-qubit Z_i observables, with i ranging from 1 to m. The m-bit outcome corresponds to the outcome of $\tilde{Z}$ encoded in a classical repetition code. Thus, the verifier will simply take the majority bit as the outcome of $\tilde{Z}$.

For our protocol, the verifier will measure a local term of the encoded Hamiltonian, in a transversal way, and perform the classical post-processing of the results in order to extract the corrected measurement outcome. With this corrected outcome, the acceptance condition is the same as in the "unencoded" case (i.e. if the outcome for the measurement of term $\tilde{S}_i$ is $-sgn(a_i)$).

To guarantee that this construction works, we show the following:

(1) The encoded Hamiltonian preserves the $a - b$ promise gap of the original Hamiltonian. This is equivalent to showing that the encoded ground state of the original Hamiltonian is a ground state of the encoded Hamiltonian having the same energy.

(2) A polylogarithmic number of concatenations of the CSS code is sufficient to maintain an inverse polynomial acceptance-rejection gap in the presence of noise.

Having these properties guarantees that the fault tolerant post hoc protocol is both correct and sound, even in the presence of noisy devices. Before stating this as a theorem we first need to describe the noise model we are considering. The verifier makes X and Z measurements, but with probability ϵ_m the measurement outcome is erroneous. The probability of error is assumed to be independent between uses of the measurement devices, i.e. there are no correlated errors. To be a bit more precise, for ideal measurement operator M_x for outcome x, we apply a unital map $\mathcal{E}$ to M_x, where with probability $1 - \epsilon_m$, M_x is unchanged, and with probability ϵ_m, M_x is changed to something else. Alternatively, if we measure an n-qubit state ρ one qubit at a time, the noisy measurement is equivalent to transforming ρ to $(\mathcal{E}^\dagger)^{\otimes n}(\rho)$, and then making an ideal measurement on each qubit individually, where $\mathcal{E}^\dagger$ is the channel that is dual to $\mathcal{E}$. This error model of the measurement device is exactly how errors are traditionally modelled in quantum computation, where they are identically and independently distributed on the qubits. We can now state the result:

Theorem 1. *The post hoc protocol of Morimae and Fitzsimons can be made fault tolerant by encoding the XZ-Hamiltonian of the protocol in a CSS code and having the verifier perform the X and Z measurements in a transversal fashion.*

Proof. Let $\tilde{X}$ and $\tilde{Z}$ be the logical X and Z operators in the chosen CSS code. We have that $\{\tilde{X}, \tilde{Z}\} = 0$ and we will assume that these operators act on $m > 0$ qubits. Since these are operators for an error correcting code, there exists an encoding unitary, denoted E, such that:

$$E(X \otimes I^{\otimes m-1})E^\dagger = \tilde{X} \tag{7}$$

$$E(Z \otimes I^{\otimes m-1})E^\dagger = \tilde{Z} \tag{8}$$

Now let $H = \sum_i a_i S_i$ be an XZ-Hamiltonian acting on $n > 0$ qubits, and let $H' = H \otimes I^{n(m-1)}$. Clearly, H and H' have the same eigenvalues. But note that using Equations 7 and 8 we have that:

$$E^{\otimes n} S_i \otimes I^{n(m-1)} E^{\otimes n} = \tilde{S}_i \tag{9}$$

where $\tilde{S}_i$ is obtained by replacing X, Z and I by $\tilde{X}$, $\tilde{Z}$ and $I^{\otimes m}$, respectively. This then implies that:

$$E^{\otimes n} H' E^{\otimes n} = \tilde{H} \tag{10}$$

where $\tilde{H} = \sum_i a_i \tilde{S}_i$ is the encoded XZ-Hamiltonian. Thus, since $\tilde{H}$ and H' are unitarily related, they will also have the same eigenvalues. Moreover, if $\left|\tilde{\psi}\right\rangle = E^{\otimes n}|\psi\rangle|anc\rangle$ is the encoded version of some n-qubit state $|\psi\rangle$, for a suitably chosen ancilla state $|anc\rangle$, it is clear that for any such $|\psi\rangle$ we have that:

$$\langle\psi|H|\psi\rangle = \left\langle\tilde{\psi}\middle|\tilde{H}\middle|\tilde{\psi}\right\rangle \tag{11}$$

Therefore, if $|\psi\rangle$ is a ground state of H, $\left|\tilde{\psi}\right\rangle$ will be a ground state of $\tilde{H}$.

This proves property **(1)**, since it shows that the encoded Hamiltonian will have the same promise gap as the original Hamiltonian.

We now move on to property **(2)**. As mentioned, when measuring an n-qubit state ρ one qubit at a time, the noisy measurement is equivalent to transforming ρ to $(\mathcal{E}^{\dagger})^{\otimes n}(\rho)$, followed by an ideal measurement on each qubit. Thus, if each qubit in the Hamiltonian is encoded in a block of qubits, then due to the error-correcting code, the probability of obtaining an incorrect outcome (after classical post-processing) has been suppressed from ϵ_m on the original qubit to at most $\alpha\epsilon_m^2$ on the whole block, for some constant α (determined by the code). Here we have implicitly used the fact that the measurement outcome for the logical qubit in one block is obtained through classical error correction (post-processing) of the outcomes of measuring the block qubits. Concatenating k times then results in probability $\alpha^{(2^k-1)}\epsilon_m^{2^k}$ of there being an error upon measuring an encoded qubit.

The verifier will make two logical qubit measurements, so to achieve a final error rate η, we must have the error for each logical qubit after k concatenations be $\alpha^{(2^k-1)}\epsilon_m^{2^k} \leq \frac{\eta}{2}$. Provided that ϵ_m is below the threshold probability $p_{th} = \alpha^{-1}$ of the code, then if each block consists of b qubits with k levels of concatenation, for each qubit we have

$$b^k = \left(\frac{\log(2/\alpha\eta)}{\log(1/\alpha\epsilon_m)}\right)^{\log b}, \qquad (12)$$

which is $O(\text{polylog}(\frac{2}{\eta}))$. So if the total number of qubits in the ground state of the original Hamiltonian is n, after k levels of encoding in blocks of size b, the total number of qubits in the encoded ground state is $O(n \text{ polylog}(\frac{2}{\eta}))$.

If the probability of acceptance (rejecting) in the original protocol (without noisy measurements) is p_{acc} (p_{rej}) and we have that $p_{acc} - p_{rej} \leq \frac{1}{\text{poly(n)}}$. Now with noisy measurements, we have that the new probability of acceptance (with error correction) is $\tilde{p}_{acc} \geq p_{acc} - \eta$ and $\tilde{p}_{rej} \leq p_{rej} + \eta$. Therefore, to maintain a polynomial gap between acceptance and rejection we must have that η is sufficiently smaller than an inverse polynomial, which only incurs a polylogarithmic overhead. Note that only a polynomial overhead is required if we wish for η to be exponentially small. □

The idea of encoding the proof state in an error-correcting code while maintaining a single-qubit measurement device for the verifier has also been considered, in the context of general QMA problems, in [27]. In that case, however, the proof state is a graph state that is used by the verifier to perform a fault tolerant measurement-based quantum computation. The verifier is also required to test that this state corresponds to the correct graph state and this is achieved through a stabilizer test.

In our case, by restricting to BQP computations, we simply require the verifier to measure the history state associated to the quantum computation. By showing that the encoded Hamiltonian has the same promise gap as the original Hamiltonian it is therefore sufficient to request that the prover encode the history state in a CSS code.

B. Example

Let us consider a toy example of our protocol in the case of an honest prover, for which we will give numerical results when using the repetition code and the Steane code, respectively. To start with, we should consider a quantum computation for which we want to construct a history state. Given that the Steane code will encode one logical qubit as 7 physical qubits, this computation needs to be small enough so that we are able to perform multiple runs of the protocol, in a reasonable amount of time. For this reason, we will choose the following one-qubit computation:

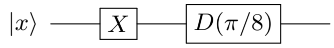

FIG. 1. Example computation.

where:

$$D(\phi) = cos(\phi)Z + sin(\phi)X \qquad (13)$$

Note that $D(\phi)$ is universal for single-qubit quantum computations[28]. The computation has two time steps, hence $T = 2$. Consider the case $x = 0$. The input state starts out as $|0\rangle$, it is then flipped to $|1\rangle$ and upon application of the $D(\pi/8)$ gate it becomes $sin(\pi/8)|0\rangle - cos(\pi/8)|1\rangle$. If we designate output $|1\rangle$ as acceptance, then this circuit will accept $x = 0$ with probability $cos(\pi/8)^2$. The history state, for $x = 0$, will be:

$$|\psi_{x=0}\rangle = \frac{1}{\sqrt{3}}(|0\rangle|00\rangle + |1\rangle|10\rangle + \\ (sin(\pi/8)|0\rangle - cos(\pi/8)|1\rangle)|11\rangle)$$

where we have separated the computation register from the clock register. For the $x = 1$ case, the history state will be:

$$|\psi_{x=1}\rangle = \frac{1}{\sqrt{3}}(|1\rangle|00\rangle + |0\rangle|10\rangle + (cos(\pi/8)|0\rangle + sin(\pi/8)|1\rangle)|11\rangle)$$

We now need to consider an XZ-Hamiltonian such that the ground state is close to $|\psi_{x=0}\rangle$. We will consider a 3-local Hamiltonian. Following the works of [22, 24], the Hamiltonian will have the following form:

$$H = H_{in} + H_{clock} + H_{prop} + H_{out}$$

where:

- H_{in} penalizes terms in which the input is not of the correct form, at the start of the computation ($T = 0$).
- H_{clock} penalizes terms in which the clock register is not of the correct form, throughout the computation.
- H_{prop} penalizes terms that do not correspond to the chosen computation.
- H_{out} penalizes terms for which the output of the computation register is not $|1\rangle$ (i.e. non-accepting computations).

In our case, we have:

$$H_{in} = (I - |x\rangle\langle x|) \otimes |0\rangle\langle 0| \otimes I$$

$$H_{clock} = I \otimes |01\rangle\langle 01|$$

$$H_{prop} = H_{prop_1} + H_{prop_2}$$

where:

$$H_{prop_1} = \frac{1}{2}(I \otimes |0\rangle\langle 0| \otimes I - X \otimes X \otimes I + I \otimes |10\rangle\langle 10|)$$

$$H_{prop_2} = \frac{1}{2}(I \otimes I \otimes |1\rangle\langle 1| - D(\pi/8) \otimes I \otimes X + I \otimes |10\rangle\langle 10|)$$

and finally:

$$H_{out} = |0\rangle\langle 0| \otimes I \otimes |1\rangle\langle 1|$$

It should be noted that $|\psi_x\rangle$ is the ground state of $H_{in} + H_{clock} + H_{prop}$, but not the ground state of H. It is the H_{out} term that singles out $|\psi_{x=0}\rangle$ and makes the ground state of H be close, in trace distance, to the history state for the $x = 0$ case. This is because in that case, the output of the computation will be $|1\rangle$, with high probability.

We now write H in XZ form:

$$H = \frac{7}{4}III + \frac{1}{4}(1 - (-1)^x)ZII - \frac{1}{4}(-1)^x ZZI - \frac{1}{4}IZZ - \frac{1}{2}XXI - \frac{1}{2}XXZ - \frac{1}{2}sin(\pi/8)XIX + \frac{1}{2}sin(\pi/8)XZX - \frac{1}{2}cos(\pi/8)ZIX + \frac{1}{2}cos(\pi/8)ZZX - \frac{1}{4}ZIZ \quad (14)$$

The protocol proceeds as follows. The verifier will inform the prover that they wish to perform the computation from Figure 1, for input $x = 0$. The prover reports that the computation accepts (with high probability) and prepares the history state $|\psi_{x=0}\rangle$, encoded in a CSS code. This state is sent qubit by qubit to the verifier. The verifier, will choose one of the terms from Equation 14, with its corresponding probability, and perform the transversal measurement of the state. For instance, the term XZX will be chosen with probability $\frac{1}{2K}sin(\pi/8)$, where $K = \sum_i |a_i| \approx 4.8$. The verifier measures the X and Z operators, performs classical post-processing on their results and combines them so as to recover the outcome of measuring XZX. She accepts on outcome -1 for this measurement, since $\frac{1}{2}sin(\pi/8)$ is positive.

For the $x = 1$ case, the situation is similar. In this case, the prover will inform the verifier that the computation rejects (with high probability) and so the verifier will change the H_{out} term of the Hamiltonian to:

$$H_{out} = |1\rangle\langle 1| \otimes I \otimes |1\rangle\langle 1| \quad (15)$$

and otherwise proceed as in the $x = 1$ case.

C. Numerical results

To simulate the above protocol, we considered two error-correcting codes: the repetition code and the Steane code. In both instances, we wanted to compare how the verifier's probability of acceptance changes as we increase the amount of noise applied to the history state. Before showing the results, we should first ask: what is the probability of acceptance, for $x = 0$, when there is no noise in the system? One can show that:

$$p_{acc} = \frac{1}{2}\left(1 - \frac{\langle\psi_{x=0}|H|\psi_{x=0}\rangle}{\sum_i |a_i|}\right) \quad (16)$$

and in our case $\langle\psi_{x=0}|H|\psi_{x=0}\rangle \approx 0.0488$. We therefore find that $p_{acc} \approx 0.4949$.

The first case we considered is the repetition code, with 3 physical qubits per logical state. This code can only correct for X errors. We therefore considered the noise channel:

$$\mathcal{F}(\rho) = (1-p)\rho + pX\rho X$$

acting independently on each individual qubit. The results are shown in Figure 2.

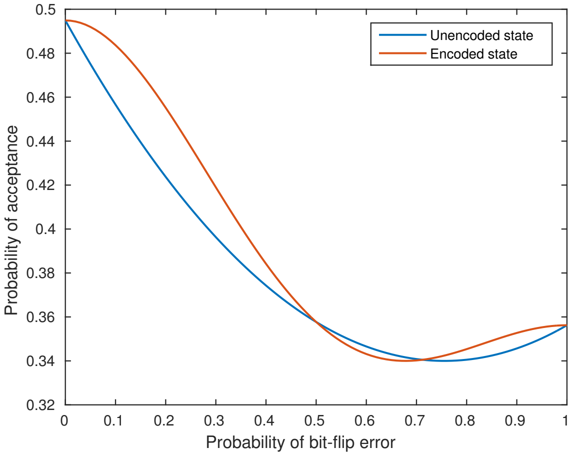

FIG. 2. Comparison between encoded and unencoded states for the 3-qubit repetition code.

As we can see, the point where the encoded state yields the same acceptance probability as the unencoded state is $p = 0.5$. The acceptance probabilities for the unencoded state were determined by applying the channel $\mathcal{F}$ to each qubit in $|\psi_{x=0}\rangle$, resulting in a state ρ, and then computing:

$$p_{acc} = \frac{1}{2} - \frac{Tr(H\rho)}{2\sum_i |a_i|} \tag{17}$$

The same is true for the encoded state, except that logical Z operators are replaced with:

$$Z_M = M_0 - M_1$$

where:

$$M_0 = |000\rangle\langle000| + |001\rangle\langle001| + |010\rangle\langle010| + |100\rangle\langle100|$$

$$M_1 = |111\rangle\langle111| + |110\rangle\langle110| + |101\rangle\langle101| + |011\rangle\langle011|$$

Essentially, the $+1$ eigenspace of Z_M is spanned by states containing a majority of $|0\rangle$ and the -1 eigenspace is spanned by states containing a majority of $|1\rangle$. Measuring Z_M is the same as performing a transversal Z measurement and taking the majority outcome.

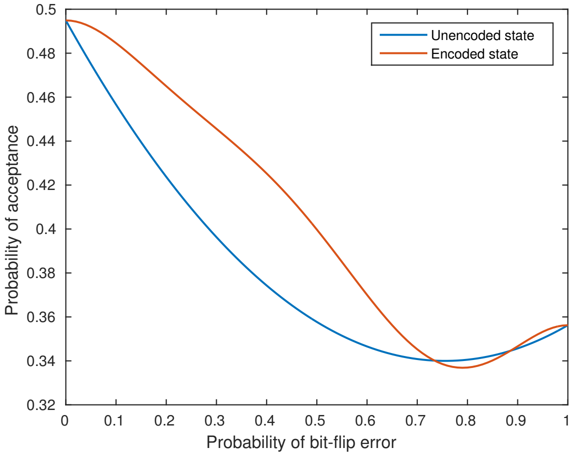

FIG. 3. Comparison between encoded and unencoded states for the 5-qubit repetition code.

If we increase the size of the encoded state to 5 qubits, we obtain the results from Figure 3. As expected, the noise threshold increases and is around $p \approx 0.72$.

We now consider the Steane code, which can detect and correct for arbitrary errors on a single qubit, while encoding one logical state in 7 physical qubits. This means that the encoded state will comprise of 21 qubits. For this case, we will assume that each qubit is subject to depolarizing noise, characterised by the channel:

$$\mathcal{D}(\rho) = (1-3p/4)\rho + p/4(X\rho X + Y\rho Y + Z\rho Z)$$

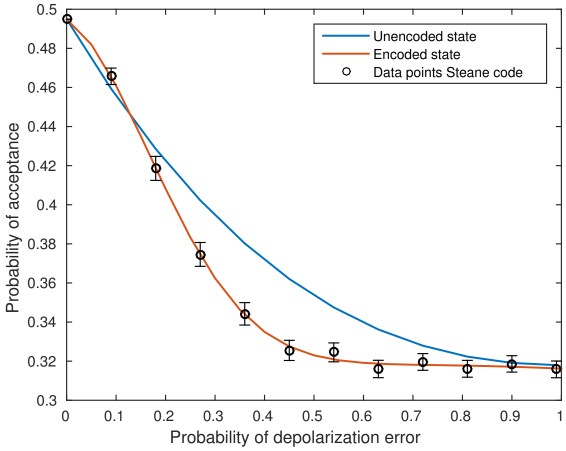

FIG. 4. Comparison between encoded and unencoded states for Steane's code.

Due to the large number of entries for the density matrix of the encoded state, we were unable to directly apply the channel $\mathcal{D}$. Instead, for each qubit in $\left|\tilde{\psi}_{x=0}\right\rangle$, we chose to either leave it unchanged, with probability $(1-3p/4)$ or, with probability $p/4$, apply either X, Y or Z. This process is repeated multiple times, and in each case the probability of acceptance is computed using Equation 17. The overall probability of acceptance is then estimated by taking the average over all of these runs. The results are shown in Figure 4.

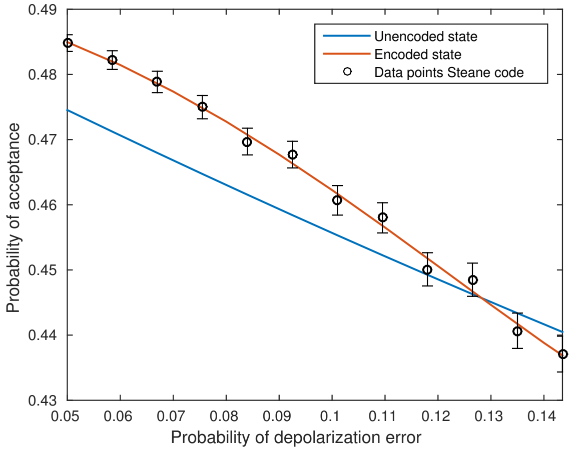

FIG. 5. Threshold for the Steane code.

We considered 12 data points, spread equally in the interval $[0, 1]$, and for each we performed 1000 repetitions of applying noise in order to estimate p_{acc}. The error bars represent confidence intervals for the computed values, assuming a confidence of 95%. Additionally, the orange curve represents the best fit interpolation of the given samples, when assuming a Gaussian model. As we can see, the threshold point appears to be between 0.1 and 0.2. By considering 12 samples in the range between 0.05 and 0.15, and 4000 repetitions per sample, in Figure 5, we find that the threshold point is between 0.12 and 0.13.

The simulations were performed in MATLAB, on the Eddie Mark 3 cluster of The University of Edinburgh. The code for our simulations is available on Github [29].

It should be emphasized that these numerical results are a demonstration of the proof-of-concept and not a rigorous numerical analysis of the typical improvement one can obtain with our method. Indeed, we only considered a specific single-qubit quantum circuit, and a more in-depth analysis would consider a general family of multi-qubit quantum circuits.

IV. CONCLUSIONS

We have given a simple construction for a fault tolerant quantum verification protocol. In a nutshell, the construction involves taking the original post hoc verification protocol of Morimae and Fitzsimons and encoding it in a CSS error-correcting code. Since the original protocol was not blind, neither is its fault tolerant counterpart. A protocol being blind means that the delegated computation is kept secret from the prover, and they only learn at most the size the computation. A major open problem that remains to be addressed is whether one can achieve fault tolerant verification of blind quantum computation without resorting to additional assumptions, as in [13–15]. Specifically, the protocols from [13–15] assumed (either implicitly or explicitly) that the noise on the verifier's device is independent of the secret parameters that are used to achieve blindness. Additionally, the noise, on that device, should be uncorrelated with the prover's private system.

Following the discussion in [17], the authors stress that, so far, there is no protocol that simultaneously achieves all of the following properties:

(1) The verifier has a preparation or measurement device whose size is at most polylogarithmic in the size of the delegated quantum computation.

(2) The noise rate for each quantum operation is below some constant threshold. Additionally, the noise on the verifier's device can depend on whatever operations the verifier performs and can be correlated with the prover's quantum system.

(3) The protocol is unconditionally blind. In other words, throughout the interaction with the verifier, the prover only learns the size of the delegated quantum computation.

As mentioned, previous approaches achieved conditions 1 and 3 but not 2. The protocol we proposed achieves conditions 1 (with a constant size device) and 2 but not 3.

Recently, a protocol has been proposed in which a classical client can delegate and verify the computations performed by a quantum server [30]. This protocol, however, relies on certain computational assumptions about whether a quantum computer can solve a particular problem. Therefore, the verifier would not need to worry about introducing errors into the prover's quantum computation, as was the concern in our work, but this comes at the cost of making these computational assumptions. Interestingly, the protocol in [30] also uses post hoc verification as a primitive, except now the prover measures the qubits in the history state and relays the outcomes to the verifier. The preparation of the history state is slightly more complex than in our case since it uses cryptographic one-way functions which introduce some overhead.

Returning to our results, the simulations are encouraging. Given that the obtained thresholds are higher than the error rates observed in current experimental implementations [31–33], a demonstration of the protocol in the near future is likely. The major obstacle to such a demonstration would be the production of these highly entangled history states. The use of CSS codes, however, means that one can encode these states in codes having even higher noise thresholds than the Steane code, such as surface codes [34].

Acknowledgments. We acknowledge useful correspondence with Thomas Vidick. This work was supported by EPSRC grants EP/N003829/1 and EP/M013243/1. MJH also acknowledges funding from the EPSRC grant Building Large Quantum States out of Light (EP/K034480/1).

[1] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[2] Dorit Aharonov and Umesh Vazirani. *Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics*. Computability: Turing, Gödel, Church, and Beyond. MIT Press, 2013.

[3] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *arXiv preprint arXiv:1709.06984*, 2017.

[4] Recently, a protocol has been proposed in which a classical client can delegate and verify the computations performed by a quantum server [30], however that protocol assumes that a certain problem (called *learning with errors* [35]) is intractable for quantum computers.

[5] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008.

[6] Alexei Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997.

[7] Emanuel Knill, Raymond Laflamme, and Wojciech H Zurek. Resilient quantum computation: error models and thresholds. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 365–384. The Royal Society, 1998.

[8] For polylogarithmic size (uniform) quantum circuits, there is no known general means of classically simulating them, since a brute force simulation would run in quasipolynomial time.

[9] Joseph F Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Physical review letters*, 120(4):040501, 2018.

[10] Joseph F Fitzsimons and Michal Hajdušek. Post hoc verification of quantum computation. *arXiv preprint arXiv:1512.04375*, 2015.

[11] Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

[12] Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5):793, 1996.

[13] Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *New Journal of Physics*, 17(8):083040, 2015.

[14] Theodoros Kapourniotis and Animesh Datta. Nonadaptive fault-tolerant verification of quantum supremacy with noise. *arXiv preprint arXiv:1703.09568*, 2017.

[15] Keisuke Fujii and Masahito Hayashi. Verifiable fault tolerance in measurement-based quantum computation. *Physical Review A*, 96(3):030301, 2017.

[16] Yuki Takeuchi, Keisuke Fujii, Tomoyuki Morimae, and Nobuyuki Imoto. Fault-tolerant verifiable blind quantum computing with logical state remote preparation. *arXiv preprint arXiv:1607.01568v3*, 2017.

[17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint arXiv:1704.04487*, 2017.

[18] Tomoyuki Morimae. Two-message quantum-arthur-merlin game with single-qubit measurements. *Physical Review A*, 93(6):062333, 2016.

[19] A complexity class is closed under complement, if for all languages L contained in that class, the complement of L, denoted L^c and consisting of all strings not contained in L, is also contained in the class.

[20] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*, volume 47. American Mathematical Society Providence, 2002.

[21] Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: the quantum pcp conjecture. *Acm sigact news*, 44(2):47–79, 2013.

[22] Julia Kempe and Oded Regev. 3-local hamitonian is qma-complete. *Quantum Information & Computation*, 3(3):258–264, 2003.

[23] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[24] Jacob D. Biamonte and Peter J. Love. Realizable

hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.

[25] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[26] Tomoyuki Morimae. Blind quantum computing can always be made verifiable. *arXiv preprint arXiv:1803.06624*, 2018.

[27] Tomoyuki Morimae, Keisuke Fujii, and Harumichi Nishimura. Quantum merlin-arthur with noisy channel. *arXiv preprint arXiv:1608.04829*, 2016.

[28] Additionally, $\{CNOT, D(\phi)\}$ is universal for general quantum computations.

[29] Github repository. `https://bit.ly/2JSJDBA`.

[30] Urmila Mahadev. Classical verification of quantum computations. *arXiv preprint arXiv:1804.01082*, 2018.

[31] Michel H Devoret and Robert J Schoelkopf. Superconducting circuits for quantum information: an outlook. *Science*, 339(6124):1169–1174, 2013.

[32] Maika Takita, Andrew W Cross, AD Córcoles, Jerry M Chow, and Jay M Gambetta. Experimental demonstration of fault-tolerant state preparation with superconducting qubits. *Physical review letters*, 119(18):180501, 2017.

[33] Google bristlecone. `https://bit.ly/2FZcZwD`.

[34] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012.

[35] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.