Theory Comput Syst (2019) 63:715–808
https://doi.org/10.1007/s00224-018-9872-3

Verification of Quantum Computation: An Overview of Existing Approaches

Alexandru Gheorghiu[1] · **Theodoros Kapourniotis**[2] · **Elham Kashefi**[1,3]

Published online: 6 July 2018

Abstract Quantum computers promise to efficiently solve not only problems believed to be intractable for classical computers, but also problems for which verifying the solution is also considered intractable. This raises the question of how one can check whether quantum computers are indeed producing correct results. This task, known as *quantum verification*, has been highlighted as a significant challenge on the road to scalable quantum computing technology. We review the most significant approaches to quantum verification and compare them in terms of structure, complexity and required resources. We also comment on the use of cryptographic techniques which, for many of the presented protocols, has proven extremely useful in performing verification. Finally, we discuss issues related to fault tolerance, experimental implementations and the outlook for future protocols.

Keywords Verification of quantum computation · Delegated quantum computation · Quantum cryptography · Blind quantum computing

This article is part of the Topical Collection on *Computer Science Symposium in Russia*

✉ Alexandru Gheorghiu
a.gheorghiu@sms.ed.ac.uk

[1] School of Informatics, University of Edinburgh, Edinburgh, UK

[2] Department of Physics, University of Warwick, Coventry, UK

[3] CNRS LIP6, Université Pierre et Marie Curie, Paris, France

1 Introduction

Quantum computation is the subject of intense research due to the potential of quantum computers to efficiently solve problems which are believed to be intractable for classical computers. The current focus of experiments, aiming to realize scalable quantum computation, is to demonstrate a *quantum computational advantage*. In other words, this means performing a quantum computation in order to solve a problem which is proven to be classically intractable, based on plausible complexity-theoretic assumptions. Examples of such problems, suitable for near-term experiments, include *boson sampling* [1], *instantaneous quantum polynomial time* (IQP) computations [2] and others [3–5]. The prospect of achieving these tasks has ignited a flurry of experimental efforts [6–9]. However, while demonstrating a quantum computational advantage is an important milestone towards scalable quantum computing, it also raises a significant challenge:

> If a quantum experiment solves a problem which is proven to be intractable for classical computers, how can one verify the outcome of the experiment?

The first researcher who formalised the above "paradox" as a *complexity theoretic* question was Gottesman, in a 2004 conference [10]. It was then promoted, in 2007, as a complexity challenge by Aaronson who asked: "*If a quantum computer can efficiently solve a problem, can it also efficiently convince an observer that the solution is correct? More formally, does every language in the class of quantumly tractable problems (*BQP*) admit an interactive proof where the prover is in* BQP *and the verifier is in the class of classically tractable problems (*BPP*)?*" [10]. Vazirani, then emphasized the importance of this question, not only from the perspective of complexity theory, but from a philosophical point of view [11]. In 2007, he raised the question of whether quantum mechanics is a *falsifiable theory*, and suggested that a computational approach could answer this question. This perspective was explored in depth by Aharonov and Vazirani in [12]. They argued that although many of the predictions of quantum mechanics have been experimentally verified to a remarkable precision, all of them involved systems of low complexity. In other words, they involved few particles or few degrees of freedom for the quantum mechanical system. But the same technique of "*predict and verify*" would quickly become infeasible for systems of even a few hundred interacting particles due to the exponential overhead in classically simulating quantum systems. And so what if, they ask, the predictions of quantum mechanics start to differ significantly from the real world in the high complexity regime? How would we be able to check this? Thus, the fundamental question is whether there exists a verification procedure for quantum mechanical predictions which is efficient for arbitrarily large systems.

In trying to answer this question we return to complexity theory. The primary complexity class that we are interested in is BQP, which, as mentioned above, is the class of problems that can be solved efficiently by a quantum computer. The analogous class for classical computers, with randomness, is denoted BPP. Finally, concerning verification, we have the class MA, which stands for Merlin-Arthur. This consists of problems whose solutions can be verified by a BPP machine when given

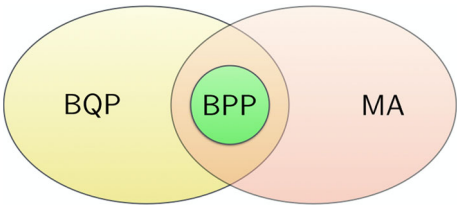


Fig. 1 Suspected relationship between BQP and MA

a proof string, called a *witness*.[1] BPP is contained in BQP, since any problem which can be solved efficiently on a classical computer can also be solved efficiently on a quantum computer. Additionally BPP is contained in MA since any BPP problem admits a trivial empty witness. Both of these containments are believed to be strict, though this is still unproven.

What about the relationship between BQP and MA? Problems are known that are contained in both classes and are believed to be outside of BPP. One such example is *factoring*. Shor's polynomial-time quantum algorithm for factoring demonstrates that the problem is in BQP [14]. Additionally, for any number to be factored, the witness simply consists of a list of its prime factors, thus showing that the problem is also in MA. In general, however, it is believed that BQP is *not* contained in MA [15, 16]. The conjectured relationship between these complexity classes is illustrated in Fig. 1.

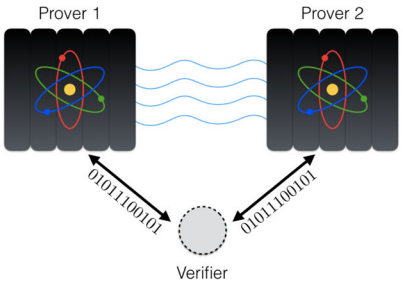
What this tells us is that, very likely, there do not exist witnesses certifying the outcomes of general quantum experiments.[2] We therefore turn to a generalization of MA known as an *interactive-proof system*. This consists of two entities: a *verifier* and a *prover*. The verifier is a BPP machine, whereas the prover has unbounded computational power. Given a problem for which the verifier wants to check a reported solution, the verifier and the prover interact for a number of rounds which is polynomial in the size of the input to the problem. At the end of this interaction, the verifier should accept a valid solution with high probability and reject, with high probability, otherwise. The class of problems which admit such a protocol is denoted IP.[3] In contrast to MA, instead of having a single proof string for each problem, one has a transcript of back-and-forth communication between the verifier and the prover.

If we are willing to allow our notion of verification to include such interactive protocols, then one would like to know whether BQP is contained in IP. Unlike the relation between BQP and MA, it is, in fact, the case that BQP $\subseteq$ IP, which means that every problem which can be efficiently solved by a quantum computer admits an interactive-proof system. One would be tempted to think that this solves

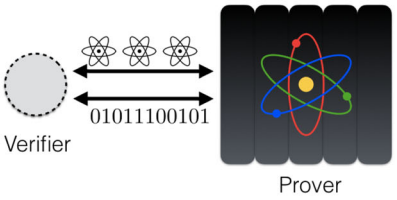
[1] BPP and MA are simply the probabilistic versions of the more familiar classes P and MA. Under plausible derandomization assumptions, BPP = P and MA = MA [13].

[2] Even if this were the case, i.e. BQP $\subseteq$ MA, for this to be useful in practice one would require that computing the witness can also be done in BQP. In fact, there are candidate problems known to be in both BQP and MA, for which computing the witness is believed to not be in BQP (a conjectured example is [17]).

[3] MA can be viewed as an interactive-proof system where only one message is sent from the prover (Merlin) to the verifier (Arthur).



(a) Classical verifier interacting with two entangled but non-communicating quantum provers



(b) Verifier with the ability to prepare or measure constant-size quantum states interacting with a single quantum prover

Fig. 2 Models for verifiable quantum computation

the question of verification, however, the situation is more subtle. Recall that in IP, the prover is computationally unbounded, whereas for our purposes we would require the prover to be restricted to BQP computations. Hence, the question that we would like answered and, arguably, the main open problem concerning quantum verification is the following:

Problem 1 (**Verifiability of BQP computations**) *Does every problem in* BQP *admit an interactive-proof system in which the prover is restricted to* BQP *computations?*

As mentioned, this complexity theoretic formulation of the problem was considered by Gottesman et al. [10, 11] and, in fact, Aaronson has offered a 25$ prize for its resolution [10]. While, as of yet, the question remains open, one does arrive at a positive answer through slight alterations of the interactive-proof system. Specifically, if the verifier interacts with two or more BQP-restricted provers, instead of one, and the provers are not allowed to communicate with each other during the protocol, then it is possible to efficiently verify arbitrary BQP computations [18–24]. Alternatively, in the single-prover setting, if we allow the verifier to have a constant-size quantum computer and the ability to send/receive quantum states to/from the prover then it is again possible to verify all polynomial-time quantum computations [25–33]. Note that in this case, while the verifier is no longer fully "classical", its computational capability is still restricted to BPP since simulating a constant-size quantum computer can be done in constant time. These scenarios are depicted in Fig. 2.

The primary technique that has been employed in most, thought not all, of these settings, to achieve verification, is known as *blindness*. This entails delegating a computation to the provers in such a way that they cannot distinguish this computation from any other of the same size, unconditionally.[4] Intuitively, verification then follows by having most of these computations be *tests* or *traps* which the verifier can check. If the provers attempt to deviate they will have a high chance of triggering these traps and prompt the verifier to reject.

[4]In other words, the provers would not be able to differentiate among the different computations even if they had unbounded computational power.

In this paper, we review all of these approaches to verification. We broadly classify the protocols as follows:

1. **Single-prover prepare-and-send**. These are protocols in which the verifier has the ability to prepare quantum states and send them to the prover. They are covered in Section 2.
2. **Single-prover receive-and-measure**. In this case, the verifier receives quantum states from the prover and has the ability to measure them. These protocols are presented in Section 3.
3. **Multi-prover entanglement-based**. In this case, the verifier is fully classical, however it interacts with more than one prover. The provers are not allowed to communicate during the protocol. Section 4 is devoted to these protocols.

From the complexity-theoretic perspective, the protocols from the first two sections are classified as QPIP (quantum prover interactive proofs) protocols, or protocols in which the verifier has a minimal quantum device and can send or receive quantum states. Conversely, the entanglement-based protocols are classified as MIP* (multi prover interactive proofs with entanglement) protocols, in which the verifier is classical and interacting with provers that share entanglement.[5]

After reviewing the major approaches to verification, in Section 5, we address a number of related topics. In particular, while all of the protocols from Sections 2–4 are concerned with the verification of general BQP computations, in Section 5.1 we mention *sub-universal* protocols, designed to verify only a particular subclass of quantum computations. Next, in Section 5.2 we discuss an important practical aspect concerning verification, which is *fault tolerance*. We comment on the possibility of making protocols resistant to noise which could affect any of the involved quantum devices. This is an important consideration for any realistic implementation of a verification protocol. Finally, in Section 5.3 we outline some of the existing experimental implementations of these protocols.

Throughout the review, we are assuming familiarity with the basics of quantum information theory and some elements of complexity theory. However, we provide a brief overview of these topics as well as other notions that are used in this review (such as *measurement-based quantum computing*) in the appendix, Section 1. Note also, that we will be referencing complexity classes such as BQP, QMA, QPIP and MIP*. Definitions for all of these are provided in Section 1 of the appendix. We begin with a short overview of *blind quantum computing*.

1.1 Blind Quantum Computing

The concept of blind computing is highly relevant to quantum verification. Here, we simply give a succinct outline of the subject. For more details, see this review of blind quantum computing protocols by Fitzsimons [34] as well as [35–39]. Note that, while the review of Fitzsimons covers all of the material presented in this section (and more), we restate the main ideas, so that our review is self-consistent

[5]The definitions of these classes can be found in Section 1.

and also in order to establish some of the notation that is used throughout the rest of the paper.

Blindness is related to the idea of *computing on encrypted data* [40]. Suppose a client has some input x and would like to compute a function f of that input, however, evaluating the function directly is computationally infeasible for the client. Luckily, the client has access to a server with the ability to evaluate $f(x)$. The problem is that the client does not trust the server with the input x, since it might involve private or secret information (e.g. medical records, military secrets, proprietary information etc). The client does, however, have the ability to encrypt x, using some encryption procedure $\mathcal{E}$, to a ciphertext $y \leftarrow \mathcal{E}(x)$. As long as this encryption procedure hides x sufficiently well, the client can send y to the server and receive in return (potentially after some interaction with the server) a string z which decrypts to $f(x)$. In other words, $f(x) \leftarrow \mathcal{D}(z)$, where $\mathcal{D}$ is a decryption procedure that can be performed efficiently by the client.[6] The encryption procedure can, roughly, provide two types of security: *computational* or *information-theoretic*. Computational security means that the protocol is secure as long as certain computational assumptions are true (for instance that the server is unable to invert one-way functions). Information-theoretic security (sometimes referred to as unconditional security), on the other hand, guarantees that the protocol is secure even against a server of unbounded computational power. See [45] for more details on these topics.

In the quantum setting, the situation is similar to that of QPIP protocols: the client is restricted to BPP computations, but has some limited quantum capabilities, whereas the server is a BQP machine. Thus, the client would like to delegate BQP functions to the server, while keeping the input and the output hidden. The first solution to this problem was provided by Childs [35]. His protocol achieves information-theoretic security but also requires the client and the server to exchange quantum messages for a number of rounds that is proportional to the size of the computation. This was later improved in a protocol by Broadbent et al. [36], known as *universal blind quantum computing* (UBQC), which maintained information-theoretic security but reduced the quantum communication to a single message from the client to the server. UBQC still requires the client and the server to have a total communication which is proportional to the size of the computation, however, apart from the first quantum message, the interaction is purely classical. Let us now state the definition of perfect, or information-theoretic, blindness from [36]:

Definition 1 (**Blindness**) Let P be a delegated quantum computation protocol involving a client and a server. The client draws the input from the random variable X. Let $L(X)$ be any function of this random variable. We say that the protocol is *blind while leaking at most L(X)* if, on the client's input X, for any $l \in Range(L)$, the following two hold when given $l \leftarrow L(X)$:

[6] In the classical setting, computing on encrypted data culminated with the development of *fully homomorphic encryption* (FHE), which is considered the "*holly grail*" of the field [41–44]. Using FHE, a client can delegate the evaluation of *any* polynomial-size classical circuit to a server, such that the input and output of the circuit are kept hidden from the server, based on reasonable computational assumptions. Moreover, the protocol involves only one round of back-and-forth interaction between client and server.

1. The distribution of the classical information obtained by the server in P is independent of X.
2. Given the distribution of classical information described in 1, the state of the quantum system obtained by the server in P is fixed and independent of X.

The definition is essentially saying that the server's "view" of the protocol should be independent of the input, when given the length of the input. This view consists, on the one hand, of the classical information he receives, which is independent of X, given $L(X)$. On the other hand, for any fixed choice of this classical information, his quantum state should also be independent of X, given $L(X)$. Note that the definition can be extended to the case of multiple servers as well. To provide intuition for how a protocol can achieve blindness, we will briefly recap the main ideas from [35, 36]. We start by considering the *quantum one-time pad*.

Quantum One-Time Pad Suppose we have two parties, Alice and Bob, and Alice wishes to send one qubit, ρ, to Bob such that all information about ρ is kept hidden from a potential eavesdropper, Eve. For this to work, we will assume that Alice and Bob share two classical random bits, denoted b_1 and b_2, that are known only to them. Alice will then apply the operation $\mathsf{X}^{b_1}\mathsf{Z}^{b_2}$ (the quantum one-time pad) to ρ, resulting in the state $\mathsf{X}^{b_1}\mathsf{Z}^{b_2}\rho\mathsf{Z}^{b_2}\mathsf{X}^{b_1}$, and send this state to Bob. If Bob then also applies $\mathsf{X}^{b_1}\mathsf{Z}^{b_2}$ to the state he received, he will recover ρ. What happens if Eve intercepts the state that Alice sends to Bob? Because Eve does not know the random bits b_1 and b_2, the state that she will intercept will be:

$$\frac{1}{4}\sum_{b_1,b_2\in\{0,1\}} \mathsf{X}^{b_1}\mathsf{Z}^{b_2}\rho\mathsf{Z}^{b_2}\mathsf{X}^{b_1} \tag{1}$$

However, it can be shown that for any single-qubit state ρ:

$$\frac{1}{4}\sum_{b_1,b_2\in\{0,1\}} \mathsf{X}^{b_1}\mathsf{Z}^{b_2}\rho\mathsf{Z}^{b_2}\mathsf{X}^{b_1} = I/2 \tag{2}$$

In other words, the state that Eve intercepts is the totally mixed state, *irrespective of the original state* ρ. But the totally mixed state is, by definition, the state of maximal uncertainty. Hence, Eve cannot recover any information about ρ, regardless of her computational power. Note, that for this argument to work, and in particular for (2) to be true, Alice and Bob's shared bits must be *uniformly random*. If Alice wishes to send n qubits to Bob, then as long as Alice and Bob share $2n$ random bits, they can simply perform the same procedure for each of the n qubits. Equation (2) generalizes for the multi-qubit case so that for an n-qubit state ρ we have:

$$\frac{1}{4^n}\sum_{\mathbf{b_1},\mathbf{b_2}\in\{0,1\}^n} \mathsf{X}(\mathbf{b_1})\mathsf{Z}(\mathbf{b_2})\rho\mathsf{Z}(\mathbf{b_2})\mathsf{X}(\mathbf{b_1}) = I/2^n \tag{3}$$

Here, $\mathbf{b_1}$ and $\mathbf{b_2}$ are n-bit vectors, $\mathsf{X}(\mathbf{b}) = \bigotimes_{i=1}^{n}\mathsf{X}^{\mathbf{b(i)}}$, $\mathsf{Z}(\mathbf{b}) = \bigotimes_{i=1}^{n}\mathsf{Z}^{\mathbf{b(i)}}$ and I is the 2^n-dimensional identity matrix.

Childs' Protocol for Blind Computation Now suppose Alice has some n-qubit state ρ and wants a quantum circuit $\mathcal{C}$ to be applied to this state and the output to be measured in the computational basis. However, she only has the ability to store n qubits, prepare qubits in the $|0\rangle$ state, swap any two qubits, or apply a Pauli X or Z to any of the n qubits. So in general, she will not be able to apply a general quantum circuit $\mathcal{C}$, or perform measurements. Bob, on the other hand, does not have these limitations as he is a BQP machine and thus able to perform universal quantum computations. How can Alice delegate the application of $\mathcal{C}$ to her state without revealing any information about it, apart from its size, to Bob? The answer is provided by Childs' protocol [35]. Before presenting the protocol, recall that any quantum circuit, $\mathcal{C}$, can be expressed as a combination of Clifford operations and T gates. Additionally, Clifford operations normalise Pauli gates. All of these notions are defined in the appendix, Section 1.

First, Alice will one-time pad her state and send the padded state to Bob. As mentioned, this will reveal no information to Bob about ρ. Next, Alice instructs Bob to start applying the gates in $\mathcal{C}$ to the padded state. Apart from the T gates, all other operations in $\mathcal{C}$ will be Clifford operations, which normalise the Pauli gates.[7] Thus, if Alice's padded state is $\mathsf{X}(\mathbf{b_1})\mathsf{Z}(\mathbf{b_2})\rho\mathsf{Z}(\mathbf{b_2})\mathsf{X}(\mathbf{b_1})$ and Bob applies the Clifford unitary U_C, the resulting state will be:

$$U_C\mathsf{X}(\mathbf{b_1})\mathsf{Z}(\mathbf{b_2})\rho\mathsf{Z}(\mathbf{b_2})\mathsf{X}(\mathbf{b_1})U_C^\dagger = \mathsf{X}(\mathbf{b_1'})\mathsf{Z}(\mathbf{b_2'})U_C\rho U_C^\dagger\mathsf{Z}(\mathbf{b_2'})\mathsf{X}(\mathbf{b_1'}) \tag{4}$$

Here, $\mathbf{b_1'}$ and $\mathbf{b_2'}$ are linearly related to $\mathbf{b_1}$ and $\mathbf{b_2}$, meaning that Alice can compute them using only xor operations. This gives her an updated pad for her state. If $\mathcal{C}$ consisted exclusively of Clifford operations then Alice would only need to keep track of the updated pad (also referred to as the *Pauli frame*) after each gate. Once Bob returns the state, she simply undoes the one-time pad using the updated key, that she computed, and recovers $\mathcal{C}\rho\mathcal{C}^\dagger$. Of course, this will not work if $\mathcal{C}$ contains T gates, since, up to an overall phase, we have that:

$$\mathsf{T}\mathsf{X}^a = \mathsf{X}^a\mathsf{S}^a\mathsf{T} \tag{5}$$

where $\mathsf{S} = \mathsf{T}^2$ and is not a Pauli gate. In other words, if we try to commute the T operation with the one-time pad we will get an unwanted S gate applied to the state. Worse, the S will have a dependency on one of the secret pad bits for that particular qubit. This means that if Alice asks Bob to apply an S^a operation she will reveal one of her pad bits. Fortunately, as explained in [35], there is a simple way to remedy this problem. After each T gate, Alice asks Bob to return the quantum state to her. Suppose that Bob had to apply a T on qubit j. Alice then applies a new one-time pad on that qubit. If the previous pad had no X gate applied to j, she will swap this qubit with a dummy state that does not take part in the computation,[8] otherwise she leaves the state unchanged. She then returns the state to Bob and asks him to apply an S gate to qubit j. Since this operation will always be applied, after a T gate, it does not

[7] In other words, for all Pauli operators P and all Clifford operators C, there exists a Pauli operator Q such that $CP = QC$.

[8] For instance, her initial state ρ could contain a number of $|0\rangle$ qubits that is equal to the number of T gates in the circuit.

reveal any information about Alice's pad. Bob's operation will therefore cancel the unwanted S gate when this appears and otherwise it will act on a qubit which does not take part in the computation. The state should then be sent back to Alice so that she can undo the swap operation if it was performed. Once all the gates in $\mathcal{C}$ have been applied, Bob is instructed to measure the resulting state in the computational basis and return the classical outcomes to Alice. Since the quantum output was one-time padded, the classical outcomes will also be one-time padded. Alice will then undo the pad an recover her desired output.

While Childs' protocol provides an elegant solution to the problem of quantum computing on encrypted data, it has significant requirements in terms of Alice's quantum capabilities. If Alice's input is fully classical, i.e. some state $|x\rangle$, where $x \in \{0,1\}^n$, then Alice would only require a constant-size quantum memory. Even so, the protocol requires Alice and Bob to exchange multiple quantum messages. This, however, is not the case with UBQC which limits the quantum communication to one quantum message sent from Alice to Bob at the beginning of the protocol. Let us now briefly state the main ideas of that protocol.

Universal Blind Quantum Computation (UBQC) In UBQC the objective is to not only hide the input (and output) from Bob, but also the circuit which will act on that input[9] [36]. As in the previous case, Alice would like to delegate to Bob the application of some circuit $\mathcal{C}$ on her input (which, for simplicity, we will assume is classical). This time, however, we view $\mathcal{C}$ as an MBQC computation.[10] By considering some universal graph state, $|G\rangle$, such as the brickwork state (see Fig. 17), Alice can convert $\mathcal{C}$ into a description of $|G\rangle$ (the graph G) along with the appropriate measurement angles for the qubits in the graph state. By the property of the universal graph state, the graph G would be the same for all circuits $\mathcal{C}'$ having the same number of gates as $\mathcal{C}$. Hence, if she were to send this description to Bob, it would not reveal to him the circuit $\mathcal{C}$, merely an upper bound on its size. It is, in fact, the measurement angles and the ordering of the measurements (known as *flow*) that uniquely characterise $\mathcal{C}$ [46]. But the measurement angles are chosen assuming all qubits in the graph state were initially prepared in the $|+\rangle$ state. Since these are XY-plane measurements, as explained in Section 1, the probabilities, for the two possible outcomes, depend only on the difference between the measurement angle and the preparation angle of the state, which is 0, in this case.[11] Suppose instead that each qubit, indexed i, in the cluster state, were instead prepared in the state $|+_{\theta_i}\rangle$. Then, if the original measurement angle for qubit i was ϕ_i, to preserve the relative angles, the new value would be $\phi_i + \theta_i$. If the values for θ_i are chosen at random, then they effectively act as a one-time pad for the original measurement angles ϕ_i. This means that if Bob does not know the preparation angles of the qubits and were instructed to measure them at the updated angles $\phi_i + \theta_i$, to him, these angles would be indistinguishable from

[9] This is also possible in Childs' protocol by simply encoding the description of the circuit $\mathcal{C}$ in the input and asking Bob to run a universal quantum circuit. The one-time padded input that is sent to Bob would then comprise of both the description of $\mathcal{C}$ as well as x, the input for $\mathcal{C}$.

[10] For a brief overview of MBQC see Section 1.

[11] This remains true even if the qubits have been entangled with the CZ operation.

random, irrespective of the values of ϕ_i. He would, however, learn the measurement outcomes of the MBQC computation. But there is a simple way to hide this information as well. One can flip the probabilities of the measurement outcomes for a particular state by performing a π rotation around Z axis. In other words, the updated measurement angles will be $\delta_i = \phi_i + \theta_i + r_i\pi$, where r_i is sampled randomly from $\{0, 1\}$.

To recap, UBQC works as follows:

(1) Alice chooses an input x and a quantum computation $\mathcal{C}$ that she would like Bob to perform on $|x\rangle$.

(2) She converts x and $\mathcal{C}$ into a pair $(G, \{\phi_i\}_i)$, where $|G\rangle$ is an N-qubit universal graph state (with an established ordering for measuring the qubits), $N = O(|\mathcal{C}|)$ and $\{\phi_i\}_i$ is the set of computation angles allowing for the MBQC computation of $\mathcal{C}\,|x\rangle$.

(3) She picks, uniformly at random, values θ_i, with i going from 1 to N, from the set $\{0, \pi/4, 2\pi/4, ...7\pi/4\}$ as well as values r_i from the set $\{0, 1\}$.

(4) She then prepares the states $\left|+_{\theta_i}\right\rangle$ and sends them to Bob, who is instructed to entangle them, using CZ operations, according to the graph structure G.

(5) Alice then asks Bob to measure the qubits at the angles $\delta_i = \phi_i' + \theta_i + r_i\pi$ and return the measurement outcomes to her. Here, ϕ_i' is an updated version of ϕ_i that incorporates corrections resulting from previous measurements, as in the description of MBQC given in Section 1.

(6) After all the measurements have been performed, Alice undoes the r_i one-time padding of the measurement outcomes, thus recovering the true outcome of the computation.

The protocol is illustrated schematically in Fig. 3, reproduced from [47] (the variables b_1, b_2, b_3 indicate measurement outcomes).

We can see that as long as Bob does not know the values of the θ_i and r_i variables, the measurements he is asked to perform, as well as their outcomes, will appear totally random to him. The reason why Bob cannot learn the values of θ_i and r_i from the qubits prepared by Alice is due to the limitation, in quantum mechanics, that one cannot distinguish between non-orthogonal states. In fact, a subsequent paper

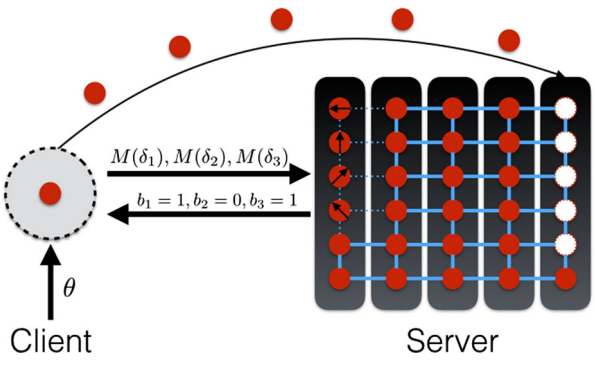


Fig. 3 Universal blind quantum computation

by Dunjko and Kashefi shows that Alice can utilize any two non-overlapping, non-orthogonal states in order to perform UBQC [48].

2 Prepare-and-Send Protocols

We start by reviewing QPIP protocols in which the only quantum capability of the verifier is to prepare and send constant-size quantum states to the prover (no measurement). The verifier must use this capability in order to delegate the application of some BQP circuit, $\mathcal{C}$, on an input $|\psi\rangle$.[12] Through interaction with the prover, the verifier will attempt to certify that the correct circuit was indeed applied on her input, with high probability, aborting the protocol otherwise.

There are three major approaches that fit this description and we devote a subsection to each of them:

1. **Section 2.1**: two protocols based on quantum authentication, developed by Aharonov et al. [25, 26].
2. **Section 2.2**: a trap-based protocol, developed by Fitzsimons and Kashefi [27].
3. **Section 2.3**: a scheme based on repeating indistinguishable runs of tests and computations, developed by Broadbent [28].

In the context of prepare-and-send protocols, it is useful to provide more refined notions of completeness and soundness than the ones in the definition of a QPIP protocol. This is because, apart from knowing that the verifier wishes to delegate a BQP computation to the prover, we also know that it prepares a particular quantum state and sends it to the prover to act with some unitary operation on it (corresponding to the quantum circuit associated with the BQP computation). This extra information allows us to define δ-*correctness* and ϵ-*verifiability*. We start with the latter:

Definition 2 (ϵ**-verifiability**) Consider a delegated quantum computation protocol between a verifier and a prover and let the verifier's quantum state be $|\psi\rangle\,|flag\rangle$, where $|\psi\rangle$ is the input state to the protocol and $|flag\rangle$ is a flag state denoting whether the verifier accepts ($|flag\rangle = |acc\rangle$) or rejects ($|flag\rangle = |rej\rangle$) at the end of the protocol. Consider also the quantum channel Enc_s (encoding), acting on the verifier's state, where s denotes a private random string, sampled by the verifier from some distribution $p(s)$. Let $\mathcal{P}_{honest}$ denote the CPTP map corresponding to the honest action of the prover in the protocol (i.e. following the instructions of the verifier) acting on the verifier's state. Additionally, define:

$$P^s_{incorrect} = (I - \left|\Psi^s_{out}\right\rangle\!\left\langle\Psi^s_{out}\right|) \otimes \left|acc^s\right\rangle\!\left\langle acc^s\right| \tag{6}$$

as a projection onto the orthogonal complement of the correct output:

$$\left|\Psi^s_{out}\right\rangle\!\left\langle\Psi^s_{out}\right| = Tr_{flag}(\mathcal{P}_{honest}(Enc_s(|\psi\rangle\,\langle\psi| \otimes |acc\rangle\,\langle acc|))) \tag{7}$$

[12] This input can be a classical bit string $|x\rangle$, though it can also be more general.

and on acceptance for the flag state:

$$\left|acc^s\right\rangle\left\langle acc^s\right| = Tr_{input}(\mathcal{P}_{honest}(Enc_s(\langle\psi\rangle\langle\psi|\otimes\langle acc\rangle\langle acc\rangle))) \tag{8}$$

We say that such a protocol is ϵ-verifiable (with $0 \leq \epsilon \leq 1$), if for any action $\mathcal{P}$, of the prover, we have that:[13]

$$Tr\left(\sum_s p(s)P^s_{incorrect}\mathcal{P}(Enc_s(|\psi\rangle\langle\psi|\otimes|acc\rangle\langle acc|))\right) \leq \epsilon \tag{9}$$

Essentially, this definition says that the probability for the output of the protocol to be incorrect *and* the verifier accepting, should be bounded by ϵ. As a simple mathematical statement we would write this as the joint distribution:

$$Pr(incorrect, accept) \leq \epsilon \tag{10}$$

One could also ask whether $Pr(incorrect|accept)$ should also be upper bounded. Indeed, it would seem like this conditional distribution is a better match for our intuition regarding the "*probability of accepting an incorrect outcome*". However, giving a sensible upper bound for the conditional distribution can be problematic. To understand why, note that we can express the conditional distribution as:

$$Pr(incorrect|accept) = \frac{Pr(incorrect, accept)}{Pr(accept)} \tag{11}$$

Now, it is true that if $Pr(accept)$ is close to 1 *and* the joint distribution is upper bounded, then the conditional distribution will also be upper bounded. Suppose however that $Pr(accept) = 2^{-O(|\mathcal{C}|)}$. In other words, the probability of acceptance is exponentially small in the size of the delegated computation.[14] In this case, to upper bound the conditional distribution, it must be that the joint probability is also inverse exponential in the size of the computation. But this is a highly unusual condition, for it would mean that the prover is more likely to deceive the verifier for smaller computations, rather than for larger ones. Moreover, as we will see with the presented protocols, it is typical for the joint probability to be upper bounded by a quantity that is independent of the size of the computation. For this reason, approaches to verification will either bound $Pr(incorrect, accept)$, or provide a bound for $Pr(incorrect|accept)$ conditioned on the fact that $Pr(accept)$ is close to 1 (see [18] for an example of this).

We now define δ-correctness:

[13] An alternative to (9) is: $TD(\rho_{out}, p\left|\Psi^s_{out}\right\rangle\left\langle\Psi^s_{out}\right| \otimes |acc^s\rangle\langle acc^s| + (1-p)\rho \otimes |rej^s\rangle\langle rej^s|) \leq \epsilon$, for some $0 \leq p \leq 1$ and some density matrix ρ, where TD denotes trace distance. In other words, the output state of the protocol, ρ_{out}, is close to a state which is a mixture of the correct output state with acceptance and an arbitrary state and rejection. This definition can be more useful when one is interested in a *quantum output* for the protocol (i.e. the prover returns a quantum state to the verifier). Such a situation is particularly useful when composing verification protocols [19, 49–51].

[14] One could imagine this happening if, for instance, the prover provides random responses to the verifier instead of performing the desired computation $\mathcal{C}$.

Definition 3 (δ**-correctness**) Consider a delegated quantum computation protocol between a verifier and a prover. Using the notation from Definition 2, and letting:

$$P^s_{correct} = \left|\Psi^s_{out}\right\rangle\!\left\langle\Psi^s_{out}\right| \otimes \left|acc^s\right\rangle\!\left\langle acc^s\right| \tag{12}$$

be the projection onto the correct output and on acceptance for the flag state, we say that such a protocol is δ-correct (with $0 \leq \delta \leq 1$), if for all strings s we have that:

$$Tr\left(P^s_{correct}\mathcal{P}_{honest}(Enc_s(|\psi\rangle\langle\psi| \otimes |acc\rangle\langle acc|))\right) \geq \delta \tag{13}$$

This definition says that when the prover behaves honestly, the verifier obtains the correct outcome, with high probability, for any possible choice of its secret parameters.

If a prepare-and-send protocol has both δ-correctness and ϵ-verifiability, for some $\delta > 0, \epsilon < 1$, it will also have completeness $\delta(1/2 + 1/poly(n))$ and soundness ϵ as a QPIP protocol, where n is the size of the input. The reason for the asymmetry in completeness and soundness is that in the definition of δ-correctness we require that the output quantum state of the protocol is δ-close to the output quantum state of the desired computation. But the computation outcome is dictated by a measurement of this state, which succeeds with probability at least $1/2 + 1/poly(n)$, from the definition of BQP. Combining these facts leads to $\delta(1/2+1/poly(n))$ completeness. It follows that for this to be a valid QPIP protocol it must be that $\delta(1/2+1/poly(n))-\epsilon \geq 1/poly(n)$, for all inputs. For simplicity, we will instead require $\delta/2 - \epsilon \geq 1/poly(n)$, which implies the previous inequality. As we will see, for all prepare-and-send protocols $\delta = 1$. This condition is easy to achieve by simply designing the protocol so that the honest behaviour of the prover leads to the correct unitary being applied to the verifier's quantum state. Therefore, the main challenge with these protocols will be to show that $\epsilon \leq 1/2 - 1/poly(n)$.

2.1 Quantum Authentication-Based Verification

This subsection is dedicated to the two protocols presented in [25, 26] by Aharonov et al. These protocols are extensions of *Quantum Authentication Schemes* (QAS), a security primitive introduced in [52] by Barnum et al. A QAS is a scheme for transmitting a quantum state over an insecure quantum channel and being able to indicate whether the state was corrupted or not. More precisely, a QAS involves a *sender* and a *receiver*. The sender has some quantum state $|\psi\rangle|flag\rangle$ that it would like to send to the receiver over an insecure channel. The state $|\psi\rangle$ is the one to be authenticated, while $|flag\rangle$ is an indicator state used to check whether the authentication was performed successfully. We will assume that $|flag\rangle$ starts in the state $|acc\rangle$. It is also assumed that the sender and the receiver share some classical key k, drawn from a probability distribution $p(k)$. To be able to detect the effects of the insecure channel on the state, the sender will first apply some encoding procedure Enc_k thus obtaining $\rho = \sum_k p(k)Enc_k(|\psi\rangle|acc\rangle)$. This state is then sent over the quantum channel where it can be tampered with by an eavesdropper resulting in a new state ρ'. The receiver, will then apply a decoding procedure to this state, resulting in $Dec_k(\rho')$

and decide whether to accept or reject by measuring the flag subsystem.[15] Similar to verification, this protocol must satisfy two properties:

1. δ**-correctness**. Intuitively this says that if the state sent through the channel was not tampered with, then the receiver should accept with high probability (at least δ), irrespective of the used keys. More formally, for $0 \leq \delta \leq 1$, let:

$$P_{correct} = |\psi\rangle\langle\psi| \otimes |acc\rangle\langle acc|$$

be the projector onto the correct state $|\psi\rangle$ and on acceptance for the flag state. Then, it must be the case that for all keys k:

$$Tr\left(P_{correct} Dec_k(Enc_k(|\psi\rangle\langle\psi| \otimes |acc\rangle\langle acc|))\right) \geq \delta$$

2. ϵ**-security**. This property states that for any deviation that the eavesdropper applies on the sent state, the probability that the resulting state is far from ideal *and* the receiver accepts is small. Formally, for $0 \leq \epsilon \leq 1$, let:

$$P_{incorrect} = (I - |\psi\rangle\langle\psi|) \otimes |acc\rangle\langle acc|$$

be the projector onto the orthogonal complement of the correct state $|\psi\rangle$, and on acceptance, for the flag state. Then, it must be the case that for any CPTP action, $\mathcal{E}$, of the eavesdropper, we have:

$$Tr\left(P_{incorrect} \sum_k p(k) Dec_k(\mathcal{E}(Enc_k(|\psi\rangle\langle\psi| \otimes |acc\rangle\langle acc|)))\right) \leq \epsilon$$

To make the similarities between QAS and prepare-and-send protocols more explicit, suppose that, in the above scheme, the receiver were trying to authenticate the state $U|\psi\rangle$ instead of $|\psi\rangle$, for some unitary U. In that case, we could view the sender as the verifier at the beginning of the protocol, the eavesdropper as the prover and the receiver as the verifier at the end of the protocol. This is illustrated in Fig. 4, reproduced from [47]. If one could therefore augment a QAS scheme with the ability of applying a quantum circuit on the state, while keeping it authenticated, then one would essentially have a prepare-and-send verification protocol. This is what is achieved by the two protocols of Aharonov et al. (Fig. 4).

Clifford-QAS VQC The first protocol, named *Clifford QAS-based Verifiable Quantum Computing* (Clifford-QAS VQC) is based on a QAS which uses Clifford operations in order to perform the encoding procedure. Strictly speaking, this protocol is not a prepare-and-send protocol, since, as we will see, it involves the verifier performing measurements as well. However, it is a precursor to the second protocol

[15] The projectors for the measurement are assumed to be $P_{acc} = |acc\rangle\langle acc|$, for acceptance and $P_{rej} = I - |acc\rangle\langle acc|$ for rejection.

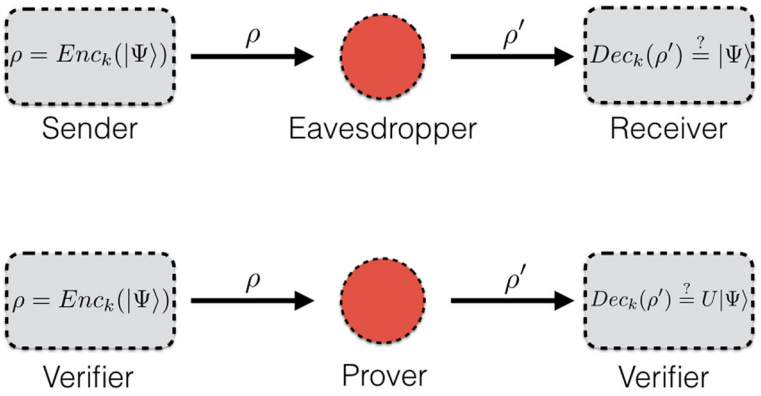


Fig. 4 QAS-based verification

from [25, 26], which is a prepare-and-send protocol. Hence, why we review the Clifford-QAS VQC protocol here.

Let us start by explaining the authentication scheme first. As before, let $|\psi\rangle\,|flag\rangle$ be the state that the sender wishes to send to the receiver and k be their shared random key. We will assume that $|\psi\rangle$ is an n-qubit state, while $|flag\rangle$ is an m-qubit state. Let $t = n + m$ and $\mathfrak{C}_t$ be the set of t-qubit Clifford operations[16] We also assume that each possible key, k, can specify a unique t-qubit Clifford operation, denoted C_k.[17] The QAS works as follows:

(1) The sender performs the encoding procedure Enc_k. This consists of applying the Clifford operation C_k to the state $|\psi\rangle|acc\rangle$.
(2) The state is sent through the quantum channel.
(3) The receiver applies the decoding procedure Dec_k which consists of applying $C_k^\dagger$ to the received state.
(4) The receiver measures the $flag$ subsystem and accepts if it is in the $|acc\rangle$ state.

We can see that this protocol has correctness $\delta = 1$, since, the sender and receiver's operations are exact inverses of each other and, when there is no intervention from the eavesdropper, they will perfectly cancel out. It is also not too difficult to show that the protocol achieves security $\epsilon = 2^{-m}$. We will include a sketch proof of this result as all other proofs of security, for prepare-and-send protocols, rely on similar ideas. Aharonov et al start by using the following lemma:

[16]Note that:

$$\mathfrak{C}_t = \{U \in U(2^t) | \sigma \in \mathbb{P}_t \implies U\sigma U^\dagger \in \mathbb{P}_t\} \tag{14}$$

where:

$$\mathbb{P}_t = \{\alpha\ \sigma_1 \otimes ... \otimes \sigma_t | \alpha \in \{+1, -1, +i, -i\}, \sigma_i \in \{I, \mathsf{X}, \mathsf{Y}, \mathsf{Z}\}\} \tag{15}$$

is the t-qubit Pauli group. See Section 1 for more details.

[17]Hence $|k| = O(log(|\mathfrak{C}_t|))$.

Lemma 1 (**Clifford twirl**) *Let P_1, P_2 be two operators from the n-qubit Pauli group, such that $P_1 \neq P_2$.*[18] *For any n-qubit density matrix ρ it is the case that:*

$$\sum_{C \in \mathfrak{C}_n} C^\dagger P_1 C \rho C^\dagger P_2 C = 0 \tag{16}$$

To see how this lemma is applied, recall that any CPTP map admits a Kraus decomposition, so we can express the eavesdropper's action as:

$$\mathcal{E}(\rho) = \sum_i K_i \rho K_i^\dagger \tag{17}$$

where, $\{K_i\}_i$ is the set of Kraus operators, satisfying:

$$\sum_i K_i^\dagger K_i = I \tag{18}$$

Additionally, recall that the n-qubit Pauli group is a basis for all $2^n \times 2^n$ matrices, which means that we can express each Kraus operator as:

$$K_i = \sum_j \alpha_{ij} P_j \tag{19}$$

where j ranges over all indices for n-qubit Pauli operators and $\{\alpha_{ij}\}_{i,j}$ is a set of complex numbers such that:

$$\sum_{ij} \alpha_{ij} \alpha_{ij}^* = 1 \tag{20}$$

For simplicity, assume that the phase information of each Pauli operator, i.e. whether it is $+1$, -1, $+i$ or $-i$, is absorbed in the α_{ij} terms. One can then re-express the eavesdropper's deviation as:

$$\mathcal{E}(\rho) = \sum_{ijk} \alpha_{ij} \alpha_{ik}^* P_j \rho P_k \tag{21}$$

We would now like to use Lemma 1 to see how this deviation affects the encoded state. Given that the encoding procedure involves applying a random Clifford operation to the initial state, which we will denote $|\Psi_{in}\rangle = |\psi\rangle\,|acc\rangle$, the state received by the eavesdropper will be:

$$\rho = \frac{1}{|\mathfrak{C}_t|} \sum_l C_l |\Psi_{in}\rangle\langle\Psi_{in}| C_l^\dagger \tag{22}$$

Acting with $\mathcal{E}$ on this state and using (21) yields:

$$\mathcal{E}(\rho) = \frac{1}{|\mathfrak{C}_t|} \sum_{ijkl} \alpha_{ij} \alpha_{ik}^* P_j C_l |\Psi_{in}\rangle\langle\Psi_{in}| C_l^\dagger P_k \tag{23}$$

[18] Technically, what is required here is that $|P_1| \neq |P_2|$, since global phases are ignored.

The receiver takes this state and applies the decoding operation, which involves inverting the Clifford that was applied by the sender. This will produce the state:

$$\frac{1}{|\mathfrak{C}_t|}\sum_{ijkl}\alpha_{ij}\alpha_{ik}^{*}C_l^{\dagger}P_jC_l|\Psi_{in}\rangle\langle\Psi_{in}|C_l^{\dagger}P_kC_l \tag{24}$$

Finally, using Lemma 1 we can see that all terms which act with different Pauli operations on both sides (i.e. $j \neq k$) will vanish, resulting in:

$$\sigma = \frac{1}{|\mathfrak{C}_t|}\sum_{ijl}\alpha_{ij}\alpha_{ij}^{*}C_l^{\dagger}P_jC_l|\Psi_{in}\rangle\langle\Psi_{in}|C_l^{\dagger}P_jC_l \tag{25}$$

Let us take a step back and understand what happened. We saw that any general map can be expressed as a combination of Pauli operators acting on both sides of the target state, ρ. Importantly, the Pauli operators on both sides needed not be equal. However, if the target state is an equal mixture of Clifford terms acting on some other state (in our case $|\Psi_{in}\rangle\,\langle\Psi_{in}\rangle$), which are then "undone" by the decoding procedure, the Clifford twirl lemma makes all non-equal Pauli terms vanish. In the resulting state, σ, we notice that each Pauli term is conjugated by Clifford operators from the set $\mathfrak{C}_t$. We know that conjugating a Pauli matrix by a Clifford operator results in a new Pauli matrix. Moreover, we know that for all j it is the case that:

$$\sum_{l}C_l^{\dagger}P_jC_l = \sum_{P\in\mathbb{P}_t}P \tag{26}$$

In other words, averaging over the Clifford group results in an equal mixture of all Pauli operations. From this and since $\alpha_{ij}\alpha_{ij}^{*} = |\alpha_{ij}|^2$ is a positive real number and $\sum_{ij}\alpha_{ij}\alpha_{ij}^{*} = 1$, the resulting state is a *uniform* convex combination of Pauli operators acting on the initial state. Mathematically, this means:

$$\sigma = \beta\,|\Psi_{in}\rangle\,\langle\Psi_{in}\rangle + \frac{1-\beta}{4^t-1}\sum_{i,P_i\neq I}P_i|\Psi_{in}\rangle\langle\Psi_{in}|P_i \tag{27}$$

where $0 \leq \beta \leq 1$.

The last element in the proof is to compute $Tr(P_{incorrect}\sigma)$. Since the first term in the mixture is the ideal state, we will be left with:

$$Tr(P_{incorrect}\sigma) = \frac{1-\beta}{4^t-1}\sum_{i,P_i\neq I}Tr(P_{incorrect}P_i|\Psi_{in}\rangle\langle\Psi_{in}|P_i) \tag{28}$$

The terms in the summation will be non-zero whenever P_i acts as identity on the flag subsystem. The number of such terms can be computed to be exactly $4^n2^m - 1$ and using the fact that $t = m + n$ and $1 - \beta \leq 1$, we have:

$$Tr(P_{incorrect}\sigma) \leq (1-\beta)\frac{4^n2^m-1}{4^{m+n}} \leq \frac{1}{2^m} \tag{29}$$

concluding the proof.

As mentioned, in all prepare-and-send protocols we assume that the verifier will prepare some state $|\psi\rangle$ on which it wants to apply a quantum circuit denoted $\mathcal{C}$. Since we are assuming that the verifier has a constant-size quantum device, the state $|\psi\rangle$

will be a product state, i.e. $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle$. For simplicity, assume each $|\psi_i\rangle$ is one qubit, though any constant number of qubits is allowed. In Clifford-QAS VQC the verifier will use the prover as an untrusted quantum storage device. Specifically, each $|\psi_i\rangle$, from $|\psi\rangle$, will be paired with a constant-size flag system in the accept state, $|acc\rangle$, resulting in a *block* of the form $|block_i\rangle = |\psi_i\rangle|acc\rangle$. Each block will be encoded, by having a random Clifford operation applied on top of it. The verifier prepares these blocks, one at a time, for all $i \in \{1, ...n\}$, and sends them to the prover. The prover is then asked to return pairs of blocks to the verifier so that she may apply gates from $\mathcal{C}$ on them (after undoing the Clifford operations). The verifier then applies new random Clifford operations on the blocks and sends them back to the prover. The process continues until all gates in $\mathcal{C}$ have been applied.

But what if the prover corrupts the state or deviates in some way? This is where the QAS enters the picture. Since each block has a random Clifford operation applied, the idea is to have the verifier use the Clifford QAS scheme to ensure that the quantum state remains authenticated after each gate in the quantum circuit is applied. In other words, if the prover attempts to deviate at any point resulting in a corrupted state, this should be detected by the authentication scheme. Putting everything together, the protocol works as follows:

(1) Suppose the input state that the verifier intends to prepare is $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle$, where each $|\psi_i\rangle$ is a one qubit state.[19] Also let $\mathcal{C}$ be quantum circuit that the verifier wishes to apply on $|\psi\rangle$. The verifier prepares (one block at a time) the state $|\psi\rangle\,|flag\rangle = |block_1\rangle \otimes |block_2\rangle \otimes ... \otimes |block_n\rangle$, where $|block_i\rangle = |\psi_i\rangle\,|acc\rangle$ and each $|acc\rangle$ state consists of a constant number m of qubits. Additionally let the size of each block be $t = m + 1$.

(2) The verifier applies a random Clifford operation, from the set $\mathfrak{C}_t$ on each block and sends it to the prover.

(3) The verifier requests a pair of blocks, $(|block_i\rangle, |block_j\rangle)$, from the prover, in order to apply a gate from $\mathcal{C}$ on the corresponding qubits, $(|\psi_i\rangle, |\psi_j\rangle)$. Once the blocks have been received, the verifier undoes the random Clifford operations and measures the flag registers, aborting if these are not in the $|acc\rangle$ state. Otherwise, the verifier performs the gate from $\mathcal{C}$, applies new random Clifford operations on each block and sends them back to the prover. This step repeats until all gates in $\mathcal{C}$ have been performed.

(4) Once all gates have been performed, the verifier requests all the blocks (one by one) in order to measure the output. As in the previous step, the verifier will undo the Clifford operations first and measure the flag registers, aborting if any of them are not in the $|acc\rangle$ state.

We can see that the security of this protocol reduces to the security of the Clifford QAS. Moreover, it is also clear that if the prover behaves honestly, then the verifier will obtain the correct output state exactly. Hence:

[19] This can simply be the state $|x\rangle$, if the verifier wishes to apply $\mathcal{C}$ on the classical input x. However, the state can be more general which is why we are not restricting it to be $|x\rangle$.

Theorem 1 *For a fixed constant $m > 0$, Clifford-QAS VQC is a prepare-and-send* QPIP *protocol having correctness $\delta = 1$ and verifiability $\epsilon = 2^{-m}$.*

Poly-QAS VQC The second protocol in [25, 26], is referred to as *Polynomial QAS-based Verifiable Quantum Computing* (Poly-QAS VQC). It improves upon the previous protocol by removing the interactive quantum communication between the verifier and the prover, reducing it to a single round of quantum messages sent at the beginning of the protocol. To encode the input, this protocol uses a specific type of quantum error correcting code known as a *polynomial CSS code* [53]. We will not elaborate on the technical details of these codes as that is beyond the scope of this review. We only mention a few basic characteristics which are necessary in order to understand the Poly-QAS VQC protocol. The polynomial CSS codes operate on *qudits* instead of qubits. A q-qudit is simply a quantum state in a q-dimensional Hilbert space. The generalized computational basis for this space is given by $\{|i\rangle\}_{i\leq q}$. The code takes a q-qudit, $|i\rangle$, as well as $|0\rangle$ states, and encodes them into a state of $t = 2d + 1$ qudits as follows:

$$E|i\rangle|0\rangle^{\otimes t-1} = \sum_{p,deg(p)\leq d,p(i)=0} |p(\alpha_1)\rangle|p(\alpha_2)\rangle...|p(\alpha_t)\rangle \tag{30}$$

where E is the encoding unitary, p ranges over polynomials of degree less than d over the field F_q of integers $mod\ q$, and $\{\alpha_j\}_{j\leq t}$ is a fixed set of m *non-zero* values from F_q (it is assumed that $q > t$). The code can detect errors on at most d qudits and can correct errors on up to $\lfloor \frac{d}{2} \rfloor$ qudits (hence $\lfloor \frac{d}{2} \rfloor$ is the weight of the code). Importantly, the code is transversal for Clifford operations. Aharonov et al consider a slight variation of this scheme called a *signed polynomial code*, which allows one to randomize over different polynomial codes. The idea is to have the encoding (and decoding) procedure also depend on a key $k \in \{-1, +1\}^t$ as follows:

$$E_k|i\rangle|0\rangle^{\otimes t-1} = \sum_{p,deg(p)\leq d,p(i)=0} |k_1 p(\alpha_1)\rangle|k_2 p(\alpha_2)\rangle...|k_t p(\alpha_t)\rangle \tag{31}$$

The signed polynomial CSS code can be used to create a simple authentication scheme having security $\epsilon = 2^{-d}$. This works by having the sender encode the state $|\Psi_{in}\rangle = |\psi\rangle|0\rangle^{\otimes t-1}$, where $|\psi\rangle$ is a qudit to be authenticated, in the signed code and then one-time padding the encoded state. Note that the $|0\rangle^{\otimes t-1}$ part of the state is acting as a flag system. We are assuming that the sender and the receiver share both the sign key of the code and the key for the one-time padding. The one-time padded state is then sent over the insecure channel. The receiver undoes the pad and applies the inverse of the encoding operation. It then measures the last $t-1$ qudits, accepting if and only if they are all in the $|0\rangle$ state. Proving security is similar to the Clifford QAS and relies on two results:

Lemma 2 (**Pauli twirl**) *Let P_1, P_2 be two operators from the n-qudit Pauli group, denoted $\mathbb{P}_n$, such that $P_1 \neq P_2$. For any n-qudit density matrix ρ it is the case that:*

$$\sum_{Q \in \mathbb{P}_n} Q^\dagger P_1 Q \rho Q^\dagger P_2 Q = 0 \tag{32}$$

This result is identical to the Clifford twirl lemma, except the Clifford operations are replaced with Pauli operators.[20] The result is also valid for qubits.

Lemma 3 (**Signed polynomial code security**) *Let $\rho = |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes t-1}$, be a state which will be encoded in the signed polynomial code, $P = (I - |\psi\rangle\langle\psi|) \otimes |0\rangle\langle 0|^{\otimes t-1}$, be a projector onto the orthogonal complement of $|\psi\rangle$ and on $|0\rangle^{t-1}$, and $Q \in \mathbb{P}_t \backslash \{I\}$ be a non-identity Pauli operation on t qudits. Then it is the case that:*

$$\frac{1}{2^t} \sum_{k \in \{-1,+1\}^t} Tr\left(P \;\; E_k^\dagger Q E_k \; \rho \; E_k^\dagger Q E_k\right) \leq \frac{1}{2^{t-1}} \tag{33}$$

Using these two results, and the ideas from the Clifford QAS scheme, it is not difficult to prove the security of the above described authentication scheme. As before, the eavesdropper's map is decomposed into Kraus operators which are then expanded into Pauli operations. Since the sender's state is one-time padded (and the receiver will undo the one-time pad), the Pauli twirl lemma will turn the eavesdropper's deviation into a convex combination of Pauli deviations:

$$\frac{1}{2^t} \sum_{k \in \{-1,+1\}^t} \sum_{Q \in \mathbb{P}_t} \beta_Q \;\; Q E_k |\Psi_{in}\rangle\langle\Psi_{in}| E_k^\dagger Q^\dagger \tag{34}$$

which can be split into the identity and non-identity Pauli terms:

$$\frac{1}{2^t} \sum_{k \in \{-1,+1\}^t} \left(\beta_I \, E_k |\Psi_{in}\rangle\langle\Psi_{in}| E_k^\dagger + \sum_{Q \in \mathbb{P}_t \backslash \{I\}} \beta_Q \;\; Q E_k \, |\Psi_{in}\rangle \, \langle\Psi_{in}| \; E_k^\dagger Q^\dagger \right) \tag{35}$$

where β_Q are positive real coefficients satisfying:

$$\sum_{Q \in \mathbb{P}_n} \beta_Q = 1 \tag{36}$$

The receiver takes this state and applies the inverse encoding operation, resulting in:

$$\rho = \frac{1}{2^t} \sum_{k \in \{-1,+1\}^t} \left(\beta_I |\Psi_{in}\rangle\langle\Psi_{in}| + \sum_{Q \in \mathbb{P}_t \backslash \{I\}} \beta_Q \;\; Q E_k |\Psi_{in}\rangle\langle\Psi_{in}| E_k^\dagger Q^\dagger \right) \tag{37}$$

[20] Note that by abuse of notation we assume $\mathbb{P}_n$ refers to the group of generalized Pauli operations over qudits, whereas, typically, one uses this notation to refer to the Pauli group of qubits.

But now we know that $\epsilon = Tr(P_{incorrect}\rho)$, and using Lemma 3 together with the facts that $Tr(P_{incorrect}|\Psi_{in}\rangle\langle\Psi_{in}|) = 0$ and that the β_Q coefficients sum to 1 we end up with:

$$\epsilon \leq \frac{1}{2^{t-1}} \leq \frac{1}{2^d} \tag{38}$$

There are two more aspects to be mentioned before giving the steps of the Poly-QAS VQC protocol. The first is that the encoding procedure for the signed polynomial code is implemented using the following interpolation operation:

$$D_k|i\rangle|k_2 p(\alpha_2)\rangle...|k_{d+1}p(\alpha_{d+1})\rangle|0\rangle^{\otimes d} = |k_1 p(\alpha_1)\rangle...|k_t p(\alpha_t)\rangle \tag{39}$$

The inverse operation $D_k^\dagger$ can be though of as a decoding of one term from the superposition in (31). Akin to Lemma 3, the signed polynomial code has the property that, when averaging over all sign keys, k, if such a term had a non-identity Pauli applied to it, when decoding it with $D_k^\dagger$, the probability that its last d qudits are not $|0\rangle$ states is upper bounded by 2^{-d}.

The second aspect is that, as mentioned, the signed polynomial code is transversal for Clifford operations. However, in order to apply non-Clifford operations it is necessary to measure encoded states together with so-called *magic states* (which will also be encoded). This manner of performing gates is known as *gate teleportation* [54]. The target state, on which we want to apply a non-Clifford operation, and the magic state are first entangled using a Clifford operation and then the magic state is measured in the computational basis. The effect of the measurement is to have a non-Clifford operation applied on the target state, along with Pauli errors which depend on the measurement outcome. For the non-Clifford operations, Aharonov et al use Toffoli gates.[21]

Given all of these, the Poly-QAS VQC protocol works as follows:

(1) Suppose the input state that the verifier intends to prepare is $|\psi\rangle = |\psi_1\rangle\otimes|\psi_2\rangle\otimes ... \otimes |\psi_n\rangle$, where each $|\psi_i\rangle$ is a q-qudit. Also suppose that the verifier wishes to apply the quantum circuit $\mathcal{C}$ on $|\psi\rangle$, which contains L Toffoli gates. The verifier prepares the state $|\Psi_{in}\rangle = |\psi_1\rangle|0\rangle^{t-1}\otimes|\psi_2\rangle|0\rangle^{t-1}\otimes...\otimes|\psi_n\rangle|0\rangle^{t-1}\otimes |M_1\rangle|0\rangle^{3t-3}\otimes ...\otimes|M_L\rangle|0\rangle^{3t-3}$, where $t = 2d+1$ and each $|M_i\rangle$ is a 3-qudit magic state, used for performing Toffoli gates. Groups of t qubits will comprise a block as follows. The first n blocks are simply $|block_i\rangle = |\psi_i\rangle|0\rangle^{t-1}$, with $i \in \{1, ..., n\}$. Next, we have the states of the form $|M_i\rangle|0\rangle^{3t-3}$ which consist of 3 blocks, each. Each block, from such a state, will comprise of one qudit from $|M_i\rangle$ and a $|0\rangle^{t-1}$ state. Note that we can no longer represent these blocks as pure states, since the 3 qudits of a $|M_i\rangle$ state are entangled. So, to summarize, each block contains one qudit from either the state $|\psi\rangle$ or a magic state $|M_i\rangle$, together with a flag system, $|0\rangle^{t-1}$.

(2) The verifier encodes each block in a signed polynomial code with a randomly chosen key $k \in \{-1, +1\}^t$ (the same key for each block) and then quantum one-time pads each block (using different keys for the padding of each block).

[21] See Section 1 for the definition of the Toffoli gate.

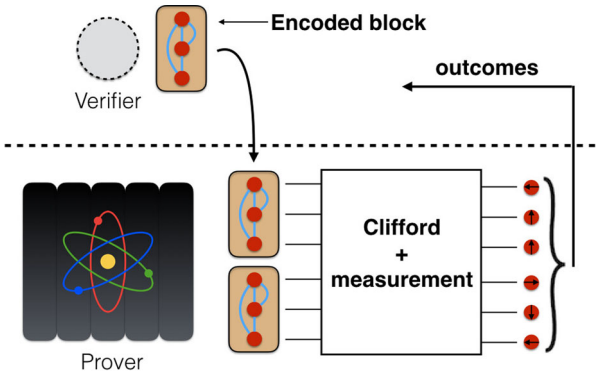


Fig. 5 Poly-QAS VQC

The blocks are prepared and encoded in sequence (the verifier has the ability to process 3 blocks, or $3t$ qudits, at a time) and then sent to the prover.

(3) When applying Clifford operations, the verifier simply asks the prover to apply the gates in a transversal fashion. Since Clifford operations normalise Pauli operators, the verifier then updates the one-time pad keys similar to Childs' protocol (see Section 1.1).

(4) When applying a Toffoli gate, the verifier asks the prover to measure 3 blocks, comprising a magic state, in the computational basis and report the measurement outcomes. It is assumed that the magic state was entangled, using a Clifford operation, with 3 target blocks on which the Toffoli gate is to be applied. The verifier undoes the (classical) one-time padding of the measurement outcomes and expects each of the 3 groups of measurement outcomes (associated with each of the 3 blocks) to be of the form $[k_1 p(\alpha_1), ..., k_t p(\alpha_t)]$. The verifier then takes these classical strings and turns them into states of the form $|\phi\rangle = |k_1 p(\alpha_1)\rangle ... |k_t p(\alpha_t)\rangle$ (using her constant-sized quantum computer).[22] She then applies $D_k^\dagger$ on each of these $|\phi\rangle$ states and checks that the last d qudits, of each state, are $|0\rangle$, aborting otherwise. Assuming not-abort, the verifier instructs the prover to perform the appropriate Pauli corrections resulting from the gate teleportation.

(5) Once all gates have been performed, the verifier instructs the prover to measure all blocks in the computational basis. As in step 4, the verifier will then de-one-time pad the outcomes, apply $D_k^\dagger$ to each state of the form $|\phi\rangle$ (prepared from these outcomes), and check that the last d qudits are $|0\rangle$, aborting otherwise.

The protocol is schematically illustrated in Fig. 5.

[22] Note that no actual quantum state was returned to the verifier by the prover. Instead, she locally prepared a quantum state from the classical outcomes reported by the prover.

As with the previous protocol, the security is based on the security of the authentication scheme. However, there is a significant difference. In the Clifford-QAS VQC protocol, one could always assume that the state received by the verifier was the correctly encoded state with a deviation on top that was independent of this encoding. However, in the Poly-QAS VQC protocol, the quantum state is never returned to the verifier and, moreover, the prover's instructed actions on this state are adaptive based on the responses of the verifier. Since the prover is free to deviate at any point throughout the protocol, if we try to commute all of his deviations to the end (i.e. view the output state as the correct state resulting from an honest run of the protocol, with a deviation on top that is independent of the secret parameters), we find that the output state will have a deviation on top which depends on the verifier's responses. Since the verifier's responses depend on the secret keys, we cannot directly use the security of the authentication scheme to prove that the protocol is 2^{-d}-verifiable.

The solution, as explained in [26], is to consider the state of the entire protocol comprising of the prover's system, the verifier's system *and* the transcript of all classical messages exchanged during the protocol. For a fixed interaction transcript, the prover's attacks can be commuted to the end of the protocol. This is because, if the transcript is fixed, there is no dependency of the prover's operations on the verifier's messages. We simply view all of his operations as unitaries acting on the joint system of his private memory, the input quantum state and the transcript. One can then use Lemma 2 and Lemma 3 to bound the projection of this state onto the incorrect subspace with acceptance. The whole state, however, will be a mixture of all possible interaction transcripts, but since each term is bounded and the probabilities of the terms in the mixture must add up to one, it follows that the protocol is 2^{-d}-verifiable:

Theorem 2 *For a fixed constant $d > 0$, Poly-QAS VQC is a prepare-and-send* QPIP *protocol having correctness $\delta = 1$ and verifiability $\epsilon = 2^{-d}$.*

Let us briefly summarize the two protocols in terms of the verifier's resources. In both protocols, if one fixes the security parameter, ϵ, the verifier must have a $O(log(1/\epsilon))$-size quantum computer. Additionally, both protocols are interactive with the total amount of communication (number of messages times the size of each message) being upper bounded by $O(|\mathcal{C}| \cdot log(1/\epsilon))$, where $\mathcal{C}$ is the quantum circuit to be performed.[23] However, in Clifford-QAS VQC, this communication is quantum whereas in Poly-QAS VQC only one quantum message is sent at the beginning of the protocol and the rest of the interaction is classical.

Before ending this subsection, we also mention the result of Broadbent et al. from [55]. This result generalises the use of quantum authentication codes for achieving verification of delegated quantum computation (not limited to decision problems). Moreover, the authors prove the security of these schemes in the *universal composability framework*, which allows for secure composition of cryptographic protocols and primitives [56].

[23] To be precise, the communication in the Poly-QAS VQC scheme is $O((n + L) \cdot log(1/\epsilon))$, where n is the size of the input and L is the number of Toffoli gates in $\mathcal{C}$.

2.2 Trap-Based Verification

In this subsection we discuss *Verifiable Universal Blind Quantum Computing* (VUBQC), which was developed by Fitzsimons and Kashefi in [27]. The protocol is written in the language of MBQC and relies on two essential ideas. The first is that an MBQC computation can be performed blindly, using UBQC, as described in Section 1.1. The second is the idea of embedding checks or *traps* in a computation in order to verify that it was performed correctly. Blindness will ensure that these checks remain hidden and so any deviation by the prover will have a high chance of triggering a trap. Notice that this is similar to the QAS-based approaches where the input state has a flag subsystem appended to it in order to detect deviations and the whole state has been encoded in some way so as to hide the input and the flag subsystem. This will lead to a similar proof of security. However, as we will see, the differences arising from using MBQC and UBQC lead to a reduction in the quantum resources of the verifier. In particular, in VUBQC the verifier requires only the ability to prepare single qubit states, which will be sent to the prover, in contrast to the QAS-based protocols which required the verifier to have a constant-size quantum computer.

Recall the main steps for performing UBQC. The client, Alice, sends qubits of the form $\left|+_{\theta_i}\right\rangle$ to Bob, the server, and instructs him to entangle them according to a graph structure, G, corresponding to some universal graph state. She then asks him to measure qubits in this graph state at angles $\delta_i = \phi_i' + \theta_i + r_i\pi$, where ϕ_i' is the corrected computation angle and $r_i\pi$ acts a random Z operation which flips the measurement outcome. Alice will use the measurement outcomes, denoted b_i, provided by Bob to update the computation angles for future measurements. Throughout the protocol, Bob's perspective is that the states, measurements and measurement outcomes are indistinguishable from random. Once all measurements have been performed, Alice will undo the r_i padding of the final outcomes and recover her output. Of course, UBQC does not provide any guarantee that the output she gets is the correct one, since Bob could have deviated from her instructions.

Transitioning to VUBQC, we will identify Alice as the verifier and Bob as the prover. To augment UBQC with the ability to detect malicious behaviour on the prover's part, the verifier will introduce traps in the computation. How will she do this? Recall that the qubits which will comprise $|G\rangle$ need to be entangled with the CZ operation. Of course, for XY-plane states CZ does indeed entangle the states. However, if either qubit, on which CZ acts, is $|0\rangle$ or $|1\rangle$, then no entanglement is created. So suppose that we have a $|+_\theta\rangle$ qubit whose neighbours, according to G, are computational basis states. Then, this qubit will remain disentangled from the rest of the qubits in $|G\rangle$. This means that if the qubit is measured at its preparation angle, the outcome will be deterministic. The verifier can exploit this fact to certify that the prover is performing the correct measurements. Such states are referred to as *trap qubits*, whereas the $|0\rangle$, $|1\rangle$ neighbours are referred to as *dummy qubits*. Importantly,

as long as G's structure remains that of a universal graph state[24] and as long as the dummy qubits and the traps are chosen at random, adding these extra states as part of the UBQC computation will not affect the blindness of the protocol. The implication of this is that the prover will be completely unaware of the positions of the traps and dummies. The traps effectively play a role that is similar to that of the flag subsystem in the authentication-based protocols. The dummies, on the other hand, are there to ensure that the traps do not get entangled with the rest of qubits in the graph state. They also serve another purpose. When a dummy is in a $|1\rangle$ state, and a CZ acts on it and a trap qubit, in the state $|+_\theta\rangle$, the effect is to "flip" the trap to $|-_\theta\rangle$ (alternatively $|-_\theta\rangle$ would have been flipped to $|+_\theta\rangle$). This means that if the trap is measured at its preparation angle, θ, the measurement outcome will also be flipped, with respect to the initial preparation. Conversely, if the dummy was initially in the state $|0\rangle$, then no flip occurs. Traps and dummies, therefore, serve to also certify that the prover is performing the CZ operations correctly. Thus, by using the traps (and the dummies), the verifier can check both the prover's measurements and his entangling operations and hence verify his MBQC computation.

We are now ready to present the steps of VUBQC:

(1) The verifier chooses an input x and a quantum computation $\mathcal{C}$ that she would like the prover to perform on $|x\rangle$.[25]

(2) She converts x and $\mathcal{C}$ into a pair $(G, \{\phi_i\}_i)$, where $|G\rangle$ is an N-qubit universal graph state (with an established ordering for measuring the qubits), which admits an embedding of T traps and D dummies. We therefore have that $N = T + D + Q$, where $Q = O(|\mathcal{C}|)$ is the number of computation qubits used for performing $\mathcal{C}$ and $\{\phi_i\}_{i\leq Q}$ is the associated set of computation angles.[26]

(3) Alice picks, uniformly at random, values θ_i, with i going from 1 to $T + Q$, from the set $\{0, \pi/4, 2\pi/4, ...7\pi/4\}$ as well as values r_i from the set $\{0, 1\}$ for the trap and computation qubits.

(4) She then prepares the $T + Q$ states $\left|+_{\theta_i}\right\rangle$, as well as D dummy qubits which are states chosen at random from $\{|0\rangle, |1\rangle\}$. All these states are sent to Bob, who is instructed to entangle them, using CZ operations, according to the graph structure G.

(5) Alice then asks Bob to measure the qubits as follows: computation qubits will be measured at $\delta_i = \phi_i' + \theta_i + r_i\pi$, where ϕ_i' is an updated version of ϕ_i that incorporates corrections resulting from previous measurements; trap qubits will

[24] Note that adding dummy qubits into the graph will have the effect of disconnecting qubits that would otherwise have been connected. It is therefore important that the chosen graph state allows for the embedding of traps and dummies so that the desired computation can still be performed. For instance, the brickwork state from Section 1 allows for only one trap qubit to be embedded, whereas other graph states allows for multiple traps. See [27, 57] for more details.

[25] As in the previous protocols, this need not be a classical input and the verifier could prepare an input of the form $|\psi\rangle = |\psi_1\rangle \otimes ... \otimes |\psi_n\rangle$.

[26] Note that the number of traps, T, and the number of dummies, D, are related, since each trap should have only dummy neighbours in $|G\rangle$.

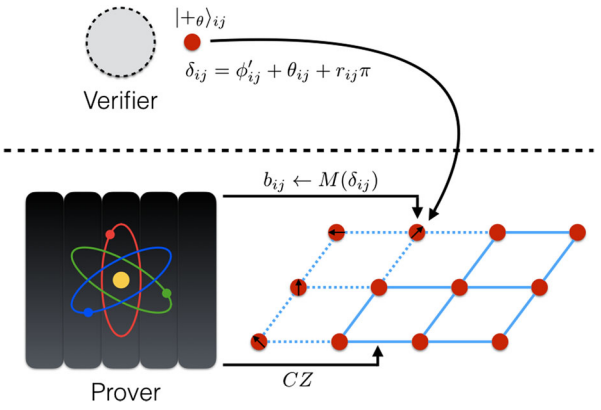


Fig. 6 Verifiable universal blind quantum computing

be measured at $\delta_i = \theta_i + r_i\pi$; dummy qubits are measured at randomly chosen angles from $\{0, \pi/4, 2\pi/4, ...7\pi/4\}$. This step is interactive as Alice needs to update the angles of future measurements based on past outcomes. The number of rounds of interaction is proportional to the depth of $\mathcal{C}$. If any of the trap measurements produce incorrect outcomes, Alice will abort upon completion of the protocol.

(6) Assuming all trap measurements succeeded, after all the measurements have been performed, Alice undoes the r_i one-time padding of the measurement outcomes, thus recovering the outcome of the computation.

The protocol is illustrated schematically in Fig. 6, where all the parameters have been labelled by their position, (i, j), in a rectangular cluster state.

One can see that VUBQC has correctness $\delta = 1$, since if the prover behaves honestly then all trap measurements will produce the correct result and the computation will have been performed correctly. What about verifiability? We will first answer this question for the case where there is a single trap qubit ($T = 1$) at a uniformly random position in $|G\rangle$, denoted $|+_{\theta_t}\rangle$. Adopting a similar notation to that from [27], we let:

$$\mathcal{B}_j(\nu) = \sum_{\mathbf{s}} p_{\nu,j}(\mathbf{s})|\mathbf{s}\rangle\langle\mathbf{s}| \otimes \rho^{\mathbf{s}}_{\nu,j} \tag{40}$$

denote the outcome density operator of *all classical and quantum messages* exchanged between the verifier and the prover throughout the protocol, excluding the last round of measurements (which corresponds to measuring the output of the computation). Additionally, ν denotes the set of secret parameters of Alice (i.e. the positions of the traps and dummies as well as the sets $\{\phi_i\}_i$, $\{\theta_i\}_i$ and $\{r_i\}_i$); j ranges over the possible strategies of the prover[27] with $j = 0$ corresponding to the

[27] Since the prover is unbounded and is free to choose any of the *uncountably* many CPTP strategies, j should be thought more of as a symbolic parameter indicating that there is a dependence on the prover's strategy and whether or not this strategy is the ideal one.

honest strategy; $\mathbf{s}$ is a binary vector which ranges over all possible *corrected* values of the measurement outcomes sent by the prover; lastly, $\rho^{\mathbf{s}}_{\nu,j}$ is the state of the unmeasured qubits, representing the output state of the computation (prior to the final measurement). To match Definition 2, one also considers:

$$P^{\nu}_{incorrect} = (I - \mathcal{C}|x\rangle\langle x|\mathcal{C}^{\dagger}) \otimes \left|+^{\nu}_{\theta_t}\right\rangle\left\langle+^{\nu}_{\theta_t}\right| \tag{41}$$

to be the projection onto the orthogonal complement of the correct output together with the trap state being projected onto acceptance. The dependence on ν, for the trap qubit, arises because the acceptance outcome depends on the states of the dummy neighbors for that qubit. This is because if one of the dummies is $|1\rangle$, the CZ operation has the effect of flipping $\left|+_{\theta_t}\right\rangle$ to $\left|-_{\theta_t}\right\rangle$. Additionally, ν also encodes the position of this trap, in the graph state, as well as the Z flip specified by the r_i parameter, for $i = t$. One then needs to find an ϵ such that:

$$Tr\left(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} \mathcal{B}_j(\nu)\right) \leq \epsilon \tag{42}$$

This is done in a manner similar to the proof of security for the Poly-QAS VQC scheme of the previous section.[28] Specifically, one fixes the interaction transcript for the protocol. This just means fixing the measurement angles δ_i, and then considering all possible transcripts compatible with the fixed angles. One can do this because UBQC guarantees that the prover learns nothing from the interaction except for, at most, an upper bound on $|\mathcal{C}|$. This means that there will be multiple transcripts compatible with the same values for the δ_i angles. It also means that any deviation that the prover performs is independent of the secret parameters of the verifier (though it can depend on the δ_i angles) and can therefore be commuted to the end of the protocol. The outcome density operator $\mathcal{B}_j(\nu)$ can then be expressed as the ideal outcome with a CPTP deviation, $\mathcal{E}_j$, on top, that is independent of ν:

$$\mathcal{B}_j(\nu) = \mathcal{E}_j(\mathcal{B}_0(\nu)) \tag{43}$$

The deviation $\mathcal{E}_j$ is then decomposed into Kraus operators which, in turn, are decomposed into Pauli operators leading to:

$$\mathcal{B}_j(\nu) = \sum_{k,l,m} \alpha_{kl}(j)\alpha^{*}_{km}(j)\ P_l \mathcal{B}_0(\nu) P_m \tag{44}$$

where $\alpha_{kl}(j)$ (and their conjugates) are the complex coefficients for the Pauli operators. This summation can be split into the terms that act as identity on $\mathcal{B}_0(\nu)$ and

[28] Note that the security proof for Poly-QAS VQC was in fact inspired from that of the VUBQC protocol, as mentioned in [26].

those that do not. Suppose the terms that act trivially have weight $0 \leq \beta \leq 1$, we then have:

$$\mathcal{B}_j(\nu) = \beta\mathcal{B}_0(\nu) + (1-\beta)\sum_{k,l,m}\alpha_{kl}(j)\alpha^*_{km}(j)\ P_l\mathcal{B}_0(\nu)P_m \tag{45}$$

where the second term is summing over Pauli operators that act non-trivially. We use this to compute the probability of accepting an incorrect outcome, noting that $P^{\nu}_{incorrect}\mathcal{B}_0(\nu) = 0$:

$$\begin{aligned} &Tr\left(\sum_{\nu} p(\nu)P^{\nu}_{incorrect}\mathcal{B}_j(\nu)\right) \\ &= (1-\beta)Tr\left(\sum_{\nu}\sum_{k,l,m} p(\nu)P^{\nu}_{incorrect}(\alpha_{kl}(j)\alpha^*_{km}(j)\ P_l\mathcal{B}_0(\nu)P_m)\right) \end{aligned} \tag{46}$$

We now use the fact that $P^{\nu}_{incorrect} = (I - \mathcal{C}|x\rangle\langle x|\mathcal{C}^\dagger)\otimes\left|+^{\nu}_{\theta_t}\right\rangle\left\langle+^{\nu}_{\theta_t}\right|$ and keep only the projection onto the trap qubit. The projection onto the space orthogonal to the correct state is a trace decreasing operation and also $(1-\beta) \leq 1$ hence:

$$\begin{aligned} &Tr\left(\sum_{\nu} p(\nu)P^{\nu}_{incorrect}\mathcal{B}_j(\nu)\right) \\ &\leq Tr\left(\sum_{\nu} p(\nu)\left|+^{\nu}_{\theta_t}\right\rangle\left\langle+^{\nu}_{\theta_t}\right| \sum_{k,l,m}\alpha_{kl}(j)\alpha^*_{km}(j)\ P_l\mathcal{B}_0(\nu)P_m\right) \end{aligned} \tag{47}$$

The summation over ν can be broken into two summations: one over the position of the trap (and the dummies) and one over the remaining parameters. This latter sum makes the reduced state appear totally mixed to the prover (a fact which is ensured by UBQC). The above expression then becomes:

$$Tr\left(\sum_{\nu^t} p(\nu^t)\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right| \sum_{k,l,m}\alpha_{kl}(j)\alpha^*_{km}(j)\ P_l(\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right| \otimes (I/Tr(I)))P_m\right) \tag{48}$$

where ν^t denotes the secret parameters for the trap qubit and consists of θ_t, r_t and the position of the trap in the graph. But notice that, on the identity system, the terms in which $l \neq m$ will have no contribution to the summation. This is because at least one of the Pauli terms (either P_l or P_m) will act on the identity system. Since Pauli operators are traceless, when taking the trace these terms will be zero. For the trap system we will have:

$$Tr\left(\sum_{\nu^t} p(\nu^t)\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right|\ P_l\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right|P_m\right) = \sum_{\nu^t} p(\nu^t)\left\langle+^{\nu^t}_{\theta_t}\right|P_l\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right|P_m\left|+^{\nu^t}_{\theta_t}\right\rangle \tag{49}$$

Note that we are taking $p(\nu^t)$ to be the uniform distribution over these parameters. By summing over θ_t and r_t, the above expression becomes zero, whenever $l \neq m$.

This is a result of the Pauli twirl Lemma 2. Thus, only terms in which $l = m$ will remain. Substituting this back into expression (48) leads to:

$$Tr\left(\sum_{v^t} p(v^t)\left|+^{v^t}_{\theta_t}\right\rangle\left\langle+^{v^t}_{\theta_t}\right| \sum_{k,l} |\alpha_{kl}(j)|^2 \; P_l(\left|+^{v^t}_{\theta_t}\right\rangle\left\langle+^{v^t}_{\theta_t}\right| \otimes (I/Tr(I)))P_l\right) \tag{50}$$

In other words, the resulting state is a convex combination of Pauli deviations. The position of the trap is completely randomised so that it is equally likely that any of the N qubits is the trap. Therefore, in the above summation, there will be N terms (corresponding to the N possible positions of the trap), one of which will be zero (the one in which the non-trivial Pauli deviations act on the trap qubit). Hence:

$$Tr\left(\sum_{v} p(v) P^{v}_{incorrect} \mathcal{B}_j(v)\right) \leq \frac{N-1}{N} = 1 - \frac{1}{N} \tag{51}$$

We have found that for the case of a single trap qubit, out of the total N qubits, one has $\epsilon = 1 - \frac{1}{N}$.

If however, there are multiple trap states, the bound improves. Specifically, for a type of resource state called *dotted-triple graph*, the number of traps can be a constant fraction of the total number of qubits, yielding $\epsilon = 8/9$. If the protocol is then repeated a constant number of times, d, with the verifier aborting if any of these runs gives incorrect trap outcomes, it can be shown that $\epsilon = (8/9)^d$ [57]. Alternatively, if the input state and computation are encoded in an error correcting code of distance d, then one again obtains $\epsilon = (8/9)^d$. This is useful if one is interested in a quantum output, or a classical bit string output. If, instead, one would only like a single bit output (i.e. the outcome of the decision problem) then sequential repetition and taking the majority outcome is sufficient. The fault tolerant encoding need not be done by the verifier. Instead, the prover will simply be instructed to prepare a larger resource state which also offers topological error-correction. See [27, 58, 59] for more details. An important observation, however, is that the fault tolerant encoding, just like in the Poly-QAS VQC protocol, is used *only to boost security* and not for correcting deviations arising from faulty devices. This latter case is discussed in Section 5.2. To sum up:

Theorem 3 *For a fixed constant $d > 0$, VUBQC is a prepare-and-send* QPIP *protocol having correctness $\delta = 1$ and verifiability $\epsilon = (8/9)^d$.*

It should be noted that in the original construction of the protocol, the fault tolerant encoding, used for boosting security, required the use of a resource state having $O(|\mathcal{C}|^2)$ qubits. The importance of the dotted-triple graph construction is that it achieves the same level of security while keeping the number of qubits linear in $|\mathcal{C}|$. The same effect is achieved by a composite protocol which combines the Poly-QAS VQC scheme, from the previous section, with VUBQC [51]. This works by having the verifier run small instances of VUBQC in order to prepare the encoded blocks used in the Poly-QAS VQC protocol. Because of the blindness property, the prover does not learn the secret keys used in the encoded blocks. The verifier can then run the Poly-QAS VQC protocol with the prover, using those blocks. This hybrid approach

illustrates how composition can lead to more efficient protocols. In this case, the composite protocol maintains a single qubit preparation device for the verifier (as opposed to a $O(log(1/\epsilon))$-size quantum computer) while also achieving linear communication complexity. We will encounter other composite protocols when reviewing entanglement-based protocols in Section 4.

Lastly, let us explicitly state the resources and overhead of the verifier throughout the VUBQC protocol. As mentioned, the verifier requires only a single-qubit preparation device, capable of preparing states of the form $|+_\theta\rangle$, with $\theta \in \{0, \pi/4, 2\pi/4, ...7\pi/4\}$, and $|0\rangle, |1\rangle$. The number of qubits needed is on the order of $O(|\mathcal{C}|)$. After the qubits have been sent to the prover, the two interact classically and the size of the communication is also on the order of $O(|\mathcal{C}|)$.

2.3 Verification Based on Repeated Runs

The final prepare-and-send protocol we describe is the one defined by Broadbent in [28]. While the previous approaches relied on hiding a flag subsystem or traps in either the input or the computation, this protocol has the verifier alternate between different runs designed to either test the behaviour of the prover or perform the desired quantum computation. We will refer to this as the *Test-or-Compute* protocol. From the prover's perspective, the possible runs are indistinguishable from each other, thus making him unaware if he is being tested or performing the verifier's chosen computation. Specifically, suppose the verifier would like to delegate the quantum circuit $\mathcal{C}$ to be applied on the $|0\rangle^{\otimes n}$ state,[29] where n is the size of the input. The verifier then chooses randomly between three possible runs:

- **Computation run**. The verifier delegates $\mathcal{C}|0\rangle^{\otimes n}$ to the prover.
- **X-test run**. The verifier delegates the identity computation on the $|0\rangle^{\otimes n}$ state to the prover.
- **Z-test run**. The verifier delegates the identity computation on the $|+\rangle^{\otimes n}$ state to the prover.

It turns out that this suffices in order to test against any possible malicious behavior of the prover, with high probability.

In more detail, the protocol uses a technique for quantum computing on encrypted data, described in [60], which is similar to Childs' protocol from Section 1.1, except it does not involve two-way quantum communication. The verifier will one-time pad either the $|0\rangle^{\otimes n}$ state or the $|+\rangle^{\otimes n}$ state and send the qubits to the prover. The prover is then instructed to apply the circuit $\mathcal{C}$, which consists of the gates X, Z, H, T, CNOT. As we know, the Clifford operations commute (or normalise) with the one-time pad, so the verifier would only need to appropriately update the one-time pad to account for this. However, T gates do not commute with the pad. In particular, commuting them past the X gates introduces unwanted S operations. To resolve this issue, the verifier will use a particular gadget which will allow the prover to apply T and correct for S at the same time. This gadget is shown in Fig. 7, reproduced from [28].

[29] The preparation of a specific input $|x\rangle$ can be done as part of the circuit $\mathcal{C}$.

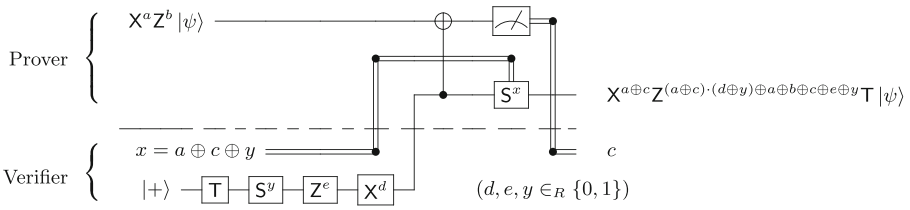


Fig. 7 Gadget for performing T on one-time padded state, reproduced from [28]

The gadget works in a gate teleportation fashion. For each qubit, labelled j, on which the prover should apply a T gate, the verifier sends a qubit of the form $\mathsf{X}^d\mathsf{Z}^c\mathsf{S}^y\mathsf{T}|+\rangle$, as well as the classical bit $x = a \oplus c \oplus y$, where a is the X padding of qubit j and c, d and y are chosen at random. The verifier then instructs the prover to apply a CNOT between the sent qubit and qubit j, effectively entangling them, and then measure qubit j in the computational basis. Lastly, the verifier instructs the prover to apply an S^x gate to the sent qubit. The end result is that this qubit will be the same as the de-one-time padded qubit j but with a T and a new one-time pad acting on it. Importantly, the new pad is kept secret from the prover.

The T gate gadget allows the verifier to control the application of either a non-Clifford operation or a Clifford operation through gate teleportation. For note that if the verifier does not apply a T gate on the qubit sent to the prover, the resulting computation is Clifford. This is what allows the verifier to switch between the computation run and the two test runs. The prover cannot distinguish between the two cases, since his side of the gadget is identical in both instances. Thus, in a test run, the computation the prover performs will be Clifford and the verifier can simply update the one-time pad, of the input, accordingly. There is, however, one complication. In an X-test run, the input is $|0\rangle^{\otimes n}$ and should remain this way until the end of the circuit, up to qubit flips resulting from the one-time pad. But any Hadamard gate in the circuit will map $|0\rangle$ to $|+\rangle$. The same is true for Z-test runs, where $|+\rangle$ states can be mapped to $|0\rangle$. To resolve this issue, Broadbent uses the following identities:

$$\mathsf{HT}^2\mathsf{HT}^2\mathsf{HT}^2\mathsf{H} = \mathsf{H} \tag{52}$$

$$\mathsf{HHHH} = I \tag{53}$$

The idea is to have the prover implement each Hadamard operation in $\mathcal{C}$ by applying four H gates alternating with $\mathsf{S} = \mathsf{T}^2$ gates. Each T^2 operation is performed by using the T gate gadget twice. When the verifier chooses a computation run, she will apply the T gates in the gadget and therefore, via (52), this leads to a Hadamard operation. Conversely, in a rest run, no T gates are applied, hence, from (53), no Hadamard operation will act on the target qubit. Since the output is always measured, by the prover, in the computational basis, in an X-test run the verifier simply checks that the de-one-time padded output is $|0\rangle^{\otimes n}$.

There is, in fact, an additional testing step being performed during an X-test run. Consider the T gadget for such a run in Fig. 8, reproduced from [28].

Note that the measurement bit, c, provided by the prover to the verifier should be an *xor* of the original X padding of the input and the updated X padding of the input.

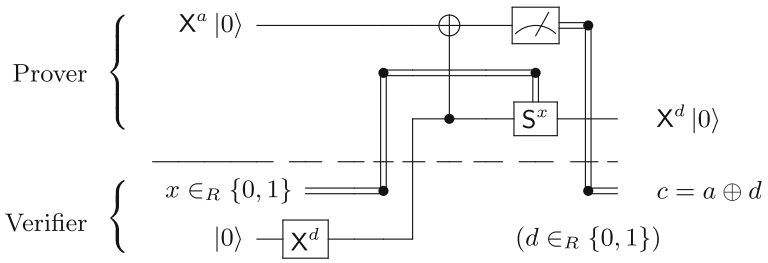


Fig. 8 T gate gadget in X-test run, reproduced from [28]

Checking the value of this bit allows the verifier to test that the gadget was applied correctly.

What about the Z-test run? In that case, the output should be the $|+\rangle^{\otimes n}$ which, upon measurement, should collapse with equal probability into any of the 2^n possible binary strings. The verifier does not test for this. Instead, each H gate that should be applied by the prover has the effect of switching between the Z-test run and the X-test run. Thus, whenever a Hadamard is applied to a qubit, during a Z-test run, the verifier switches to an X-test run until the next Hadamard operation. In the X-test runs, the verifier will use the T gate gadget from Fig. 8. These are the only checks that are performed in the Z-test run.

Putting everything together, the protocol starts by having the verifier choose an input x and a quantum computation $\mathcal{C}$ to act on this input (that includes preparing the input from the $|0\rangle^{\otimes n}$ state). Then, the verifier will randomly choose to perform one of the following:

- **Computation run**. The verifier one-time pads the $|0\rangle^{\otimes n}$ state and sends it to the prover. The prover is then instructed to apply $\mathcal{C}$ on this state, such that for each T gate in the circuit the prover and the verifier interact in order to perform the T gadget. Additionally, any H in $\mathcal{C}$ is performed as in (52). For Clifford operations, the verifier updates the one-time pad of the state accordingly. The prover is instructed to measure the output state of the circuit in the computational basis and return the outcome to the verifier. The verifier undoes the padding of this outcome and accepts if the output of the circuit indicates acceptance.
- **X-test run**. The verifier one-time pads the $|0\rangle^{\otimes n}$ state and sends it to the prover. As in the computation run, for each T, the verifier and the prover will interact to run the T gate gadget. In this case, however, the verifier will use the T gate gadget from Fig. 8, making the circuit effectively act as identity and checking that the prover is performing these gadgets correctly (rejecting otherwise). Additionally, the H gates in $\mathcal{C}$ will also act as identity, from (53), as described previously. The verifier updates the one-time padding of the state accordingly for all gates in the circuit. Once the circuit is finished, the prover is instructed to measure the output in the computational basis and report the outcome to the verifier. The verifier accepts if the de-one-time padded output is $|0\rangle^{\otimes n}$.

- **Z-test run**. The verifier one-time pads the $|+\rangle^{\otimes n}$ state and sends it to the prover. As in the X-test run, the T gate gadgets will act as identity. The H operations that the prover performs will temporarily switch the Z-test run into an X-test run, in which the verifier uses the gadget from Fig. 8 to check that prover implemented it correctly. Any subsequent H will switch back to a Z-test run. Additionally, the verifier updates the one-time padding of the state accordingly for all gates in the circuit. The prover is instructed to measure the output in the computational basis and report the outcome to the verifier, however in this case the verifier discards the output.

The asymmetry between the X-test run and the Z-test run stems from the fact that the output is always measured in the computational basis. This means that an incorrect output is one which has been bit-flipped. In turn, this implies that only X and Y operations on the output will act as deviations, since Z effectively acts as identity on computational basis states. If the circuit $\mathcal{C}$ does not contain any Hadamard gates and hence, the computation takes place entirely in the computational basis, then the X-test is sufficient for detecting such deviations. However, when Hadamard gates are present, this is no longer the case since deviations can occur in the conjugate basis, $(|+\rangle, |-\rangle)$, as well. This is why the Z-test is necessary. Its purpose is to check that the prover's operations are performed correctly when switching to the conjugate basis. For this reason, a Hadamard gate will switch a Z-test run into an X-test run which provides verification using the T gate gadget.

In terms of the correctness of the protocol, we can see that if the prover behaves honestly then the correct outcome is obtained in the computation run and the verifier will accept the test runs, hence $\delta = 1$.[30] For verifiability, the analysis is similar to the previous protocols. Suppose that $|\psi\rangle$ is either the $|0\rangle^{\oplus n}$ or the $|+\rangle^{\oplus n}$ state representing the input. Additionally, assuming there are t T gates in $\mathcal{C}$ (including the ones used for performing the Hadamards), let $|\phi\rangle$ be the state of the t qubits that the verifier sends for the T gate gadgets. Then, the one-time padded state that the prover receives is:

$$\frac{1}{4^{n+t}} \sum_{Q \in \mathbb{P}_{n+t}} Q|\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi| Q^{\dagger} \tag{54}$$

The prover is then instructed to follow the steps of the protocol in order to run the circuit $\mathcal{C}$. Note that all of the operations that he is instructed to perform are Clifford operations. This is because any non-Clifford operation from $\mathcal{C}$ is performed with the T gate gadgets (which require only Clifford operations) and the states from $|\phi\rangle$, prepared by the verifier. In following the notation from [28], we denote the honest action of the protocol as C. As in the previous protocols, the prover's deviation can

[30] However, note that if the verifier chooses a test run, in the case where the prover is honest, this will lead to acceptance irrespective of the outcome of the decision problem. This is in contrast to the previous protocols in which the testing is performed at the same time as the computation and, when the test succeeds, the verifier outputs the result of the computation.

be commuted to the end of the protocol, so that it acts on top of the correct state. After expressing the deviation map in terms of Pauli operators one gets:

$$\frac{1}{4^{n+t}} \sum_{P_i, P_j, Q \in \mathbb{P}_{n+t}} \alpha_i \alpha_j^* \; P_i Q C |\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi| C^\dagger Q^\dagger P_j^\dagger \tag{55}$$

Note that we have also commuted C past the one-time pad so that it acts on the state $|\psi\rangle|\phi\rangle$, rather than on the one-time padded versions of these states. This is possible precisely because C is a Clifford operation and therefore normalises Pauli operations. One can then assume that the verifier performs the decryption of the padding before the final measurement, yielding:

$$\frac{1}{4^{n+t}} \sum_{P_i, P_j, Q \in \mathbb{P}_{n+t}} \alpha_i \alpha_j^* \; Q^\dagger P_i Q C |\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi| C^\dagger Q^\dagger P_j^\dagger Q \tag{56}$$

We now use the Pauli twirl from Lemma 2 to get:

$$\frac{1}{4^{n+t}} \sum_{P_i \in \mathbb{P}_{n+t}} |\alpha_i|^2 \; P_i C |\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi| C^\dagger P_i^\dagger \tag{57}$$

which is a convex combination of Pauli attacks acting on the correct output state. If we now denote M to be the set of non-benign Pauli attacks (i.e. attacks which do not act as identity on the output of the computation), then one of the test runs will reject with probability:

$$\sum_{P_i \in M} |\alpha_i|^2 \tag{58}$$

This is because non-benign Pauli X or Y operations are detected by the X-test run, whereas non-benign Pauli Z operations are detected by the Z-test run. Since either test occurs with probability $1/3$, it follows that, the probability of the verifier accepting an incorrect outcome is at most $2/3$, hence $\epsilon = 2/3$.

Note that when discussing the correctness and verifiability of the Test-or-Compute protocol, we have slightly abused the terminology, since this protocol does not rigorously match the established definitions for correctness and verifiability that we have used for the previous protocols. The reason for this is the fact that in the Test-or-Compute protocol there is no additional flag or trap subsystem to indicate failure. Rather, the verifier detects malicious behaviour by alternating between different runs. It is therefore more appropriate to view the Test-or-Compute protocol simply as a QPIP protocol having a constant gap between completeness and soundness:

Theorem 4 *Test-or-Compute is a prepare-and-send* QPIP *protocol having completeness* $8/9$ *and soundness* $7/9$.

In terms of the verifier's quantum resources, we notice that, as with the VUBQC protocol, the only requirement is the preparation of single qubit states. All of these states are sent in the first round of the protocol, the rest of the interaction being completely classical.

2.4 Summary of Prepare-and-Send Protocols

The protocols, while different, have the common feature that they all use blindness or have the potential to be blind protocols. Out of the five presented protocols, only the Poly-QAS VQC and the Test-or-Compute protocols are not explicitly blind since, in both cases, the computation is revealed to the server. However, it is relatively easy to make the protocols blind by encoding the circuit into the input (which is one-time padded). Hence, one can say that all protocols achieve blindness.

This feature is essential in the proof of security for these protocols. Blindness combined with either the Pauli twirl Lemma 2 or the Clifford twirl Lemma 1 have the effect of reducing any deviation of the prover to a convex combination of Pauli attacks. Each protocol then has a specific way of detecting such an attack. In the Clifford-QAS VQC protocol, the convex combination is turned into a uniform combination and the attack is detected by a flag subsystem associated with a quantum authentication scheme. A similar approach is employed in the Poly-QAS VQC protocol, using a quantum authentication scheme based on a special type of quantum error correcting code. The VUBQC protocol utilizes trap qubits and either sequential repetition or encoding in an error correcting code to detect Pauli attacks. Finally, the Test-or-Compute protocol uses a hidden identity computation acting on either the $|0\rangle^{\otimes n}$ or $|+\rangle^{\otimes n}$ states, in order to detect the malicious behavior of the prover.

Because of these differences, each protocol will have different "quantum requirements" for the verifier. For instance, in the authentication-based protocols, the verifier is assumed to be a quantum computer operating on a quantum memory of size $O(log(1/\epsilon))$, where ϵ is the desired verifiability of the protocol. In VUBQC and Test-or-Compute, however, the verifier only requires a device capable of preparing single-qubit states. Additionally, out of all of these protocols, only Clifford-QAS VQC requires 2-way quantum communication, whereas the other three require the verifier to send only one quantum message at the beginning of the protocol, while the rest of the communication is classical. These facts, together with the communication complexities of the protocols are shown in Table 1.

As mentioned, if we want to make the Poly-QAS VQC and Test-or-Compute protocols blind, the verifier will hide her circuit by incorporating it into the input. The

Table 1 Comparison of prepare-and-send protocols

Protocol	Verifier resources	Communication	2-way quantum comm.
Clifford-QAS VQC	$O(log(1/\epsilon))$	$O(N \cdot log(1/\epsilon))$	Y
Poly-QAS VQC	$O(log(1/\epsilon))$	$O((n+L) \cdot log(1/\epsilon))$	N
VUBQC	$O(1)$	$O(N \cdot log(1/\epsilon))$	N
Test-or-Compute	$O(1)$	$O((n+T) \cdot log(1/\epsilon))$	N

If denote as $\mathcal{C}$ the circuit that the verifier wishes to delegate to the prover, and as x the input to this circuit, then $n = |x|$, $N = |\mathcal{C}|$. Additionally, T denotes the number of T gates in $\mathcal{C}$, L denotes the number of Toffoli gates in $\mathcal{C}$ and ϵ denotes the verifiability of the protocols. The second column refers to the verifier's *quantum* resources. The third column quantifies the *total* communication complexity, both classical and quantum, of the protocols (i.e. number of messages times the size of a message)

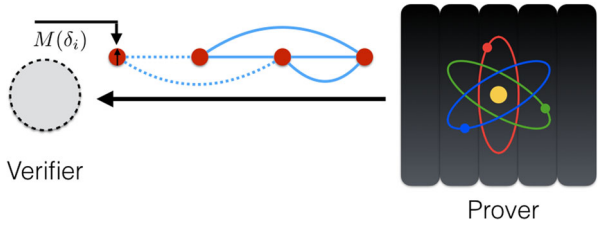


Fig. 9 Receive-and-measure protocols

input would then consist of an encoding of $\mathcal{C}$ and an encoding of x. The prover would be asked to perform controlled operations from the part of the input containing the description of $\mathcal{C}$, to the part containing x, effectively acting with $\mathcal{C}$ on x. We stress that in this case, the protocols would have a communication complexity of $O(|\mathcal{C}| \cdot log(1/\epsilon))$, just like VUBQC and Clifford-QAS VQC.[31]

3 Receive-and-Measure Protocols

The protocols presented so far have utilized a verifier with a trusted preparation device (and potentially a trusted quantum memory) interacting with a prover having the capability of storing and performing operations on arbitrarily large quantum systems. In this section, we explore protocols in which the verifier possesses a trusted measurement device. The point of these protocols is to have the prover prepare a specific quantum state and send it to the verifier. The verifier's measurements have the effect of either performing the quantum computation or extracting the outcome of the computation. An illustration of receive-and-measure protocols is shown in Fig. 9.

For prepare-and-send protocols we saw that blindness was an essential feature for achieving verifiability. While most of the receive-and-measure protocols are blind as well, we will see that it is possible to perform verification without hiding any information about the input or computation, from the prover. Additionally, while in prepare-and-send protocols the verifier was sending an encoded or encrypted quantum state to the prover, in receive-and-measure protocols, the quantum state received by the verifier is not necessarily encoded or encrypted. Moreover, this state need not contain a flag or a trap subsystem. For this reason, we can no longer consistently define ϵ-verifiability and δ-correctness, as we did for prepare-and-send protocols. Instead, we will simply view receive-and-measure protocols as QPIP protocols.

The protocols presented in this section are:

1. **Section 3.1**: a *measurement-only* protocol developed by Morimae and Hayashi that employs ideas from MBQC in order to perform verification [31].
2. **Section 3.2**: a *post hoc* verification protocol, developed by Morimae and Fitzsimons [29, 61] (and independently by Hangleiter et al. [30]).

[31] Technically, the complexity should be $O((|x| + |\mathcal{C}|) \cdot log(1/\epsilon))$, however we are assuming that $\mathcal{C}$ acts non-trivially on x (i.e. there are at least $|x|$ gates in $\mathcal{C}$).

There is an additional receive-and-measure protocol by Gheorghiu et al. [33] which we refer to as *Steering-based VUBQC*. That protocol, however, is similar to the entanglement-based GKW protocol from Section 4.1. We will therefore review Steering-based VUBQC in that subsection by comparing it to the entanglement-based protocol.

3.1 Measurement-Only Verification

In this section we discuss the measurement-only protocol from [31], which we shall simply refer to as the *measurement-only protocol*. This protocol uses MBQC to perform the quantum computation, like the VUBQC protocol from Section 2.2, however the manner in which verification is performed is more akin to Broabdent's Test-or-Compute protocol, from Section 2.3. This is because, just like in the Test-or-Compute protocol, the measurement-only approach has the verifier alternate between performing the computation or testing the prover's operations.

The key idea for this protocol, is the fact that graph states can be completely specified by a set of stabilizer operators. This fact is explained in Section 1. To reiterate the main points, recall that for a graph G, with associated graph state $|G\rangle$, if we denote as $V(G)$ the set of vertices in G and as $N_G(v)$ the set of neighbours for a given vertex v, then the generators for the stabilizer group of $|G\rangle$ are:

$$K_v = \mathsf{X}_v \prod_{w \in N_G(v)} \mathsf{Z}_w \tag{59}$$

for all $v \in V(G)$. In other words, the K_v operators generate the entire group of operators, O, such that $O|G\rangle = |G\rangle$.

When viewed as observables, stabilizers allow one to test that an unknown quantum state is in fact a particular graph state $|G\rangle$, with high probability. This is done by measuring random stabilizers of $|G\rangle$ on multiple copies of the unknown quantum state. If all measurements return the $+1$ outcome, then, the unknown state is close in trace distance to $|G\rangle$. This is related to a concept known as *self-testing*, which is the idea of determining whether an unknown quantum state and an unknown set of observables are close to a target state and observables, based on observed statistics. We postpone a further discussion of this topic to the next section, since self-testing is ubiquitous in entanglement-based protocols.

As mentioned, the measurement-only protocol involves a testing phase and a computation phase. The prover will be instructed to prepare multiple copies of a 2D cluster state, $|G\rangle$, and send them, qubit by qubit, to the verifier. The verifier will then randomly use one of these copies to perform the MBQC computation, whereas the other copies are used for testing that the correct cluster state was prepared.[32] This testing phase will involve checking all possible stabilizers of $|G\rangle$. In particular, the verifier will divide the copies to be tested into two groups, which we shall refer to as the XZ group and the ZX group. In the XZ group of states, the verifier will measure the qubits according to the 2D cluster structure, starting with an X operator in

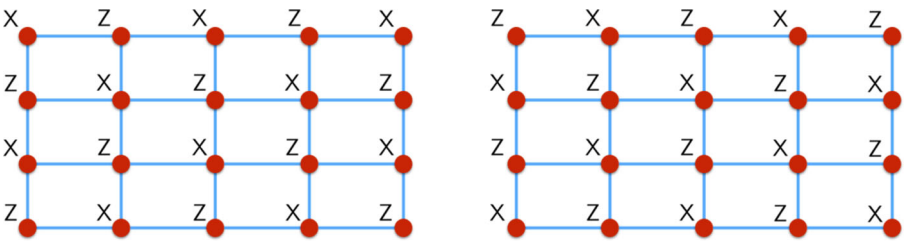
[32] This is very much in the spirit of a cryptographic technique known as *cut-and-choose* [62], which has also been used in the context of testing quantum states [63].

the upper left corner of the lattice and then alternating between X and Z. In the ZX group, she will measure the dual operators by swapping X with Z. The two cases are illustrated in Fig. 10.

Together, the measurement outcomes of the two groups can be used to infer outcomes of all stabilizer measurements defined by the K_v operators. For instance, given that the measurement outcomes for the qubits take values ± 1, to compute the outcome of a K_v measurement, for some node v that is measured with X, the verifier simply takes product of the measurement outcomes for all nodes in $\{v\} \cup N_v$. These tests allow the verifier to certify that the prover is indeed preparing copies of the state $|G\rangle$. She can then use one of these copies to run the computation. Since the prover does not know which state the verifier will use for the computation, any deviation he implements has a high chance of being detected by one of the verifier's tests. Hence, the protocol works as follows:

(1) The verifier chooses an input x and a quantum computation $\mathcal{C}$.
(2) She instructs the prover to prepare $2k + 1$ copies of a 2D cluster state, $|G\rangle$, for some constant k, and send all of the qubits, one at a time, to the verifier.
(3) The verifier randomly picks one copy to run the computation of $\mathcal{C}$ on x in an MBQC fashion. The remaining $2k$ copies are randomly divided into the XZ groups and the ZX group and measured, as described above, so as to check the stabilizers of $|G\rangle$.
(4) If all stabilizer measurement outcomes are successful (i.e. produced the outcome $+1$), then the verifier accepts the outcome of the computation, otherwise she rejects.

As with all protocols, completeness follows immediately, since if the prover behaves honestly, the verifier will accept the outcome of the computation. In the case of soundness, Hayashi and Morimae treat the problem as a *hypothesis test*. In other words, in the testing phase of the protocol the verifier is checking the hypothesis that the prover prepared $2k + 1$ copies of the state $|G\rangle$. Hayashi and Morimae then prove the following theorem:



(a) XZ group measurement operators (b) ZX group measurement operators

Fig. 10 Stabilizer measurements

Theorem 5 *Let* $1/(2k+1) \leq \alpha \leq 1$ *be the verifier's confidence level in the testing phase of the measurement-only protocol. Then, the state used by the verifier for the computation, denoted* ρ*, satisfies:*

$$\langle G|\rho|G\rangle \geq 1 - \frac{1}{\alpha(2k+1)} \tag{60}$$

This theorem is essentially showing that as the number of copies of $|G\rangle$, requested by the verifier, increases, and the verifier accepts in the testing phase, one gets that the state ρ, used by the verifier for the computation, is close in trace distance to the ideal state, $|G\rangle$. The confidence level, α, represents the maximum acceptance probability for the verifier, such that the computation state, ρ, *does not* satisfy (60). Essentially this represents the probability for the verifier to accept a computation state that is far from ideal. Hayashi and Morimae argue that the lower bound, $\alpha \geq 1/(2k+1)$, is tight, because if the prover corrupts one of the $2k+1$ states sent to the verifier, there is a $1/(2k+1)$ chance that that state will not be tested and the verifier accepts.

If one now denotes with C the POVM that the verifier applies on the computation state in order to perform the computation of $\mathcal{C}$, then it is the case that:

$$|Tr(C\rho) - Tr(C|G\rangle\langle G|)| \leq \frac{1}{\sqrt{\alpha(2k+1)}} \tag{61}$$

What this means is that the distribution of measurement outcomes for the state ρ, sent by the prover in the computation run, is almost indistinguishable from the distribution of measurement outcomes for the ideal state $|G\rangle$. The soundness of the protocol is therefore upper bounded by $\frac{1}{\sqrt{\alpha(2k+1)}}$. This implies that to achieve soundness below ϵ, for some $\epsilon > 0$, the number of copies that the prover would have to prepare scales as $O\left(\frac{1}{\alpha} \cdot \frac{1}{\epsilon^2}\right)$.

In terms of the quantum capabilities of the verifier, she only requires a single qubit measurement device capable of measuring the observables: X, Y, Z, $(\mathsf{X}+\mathsf{Y})/\sqrt{2}$, $(\mathsf{X}-\mathsf{Y})/\sqrt{2}$. Recently, however, Morimae, Takeuchi and Hayashi have proposed a similar protocol which uses *hypergraph states* [32]. These states have the property that one can perform universal quantum computations by measuring only the Pauli observables (X, Y and Z). Hypergraph states are generalizations of graph states in which the vertices of the graph are linked by hyperedges, which can connect more than two vertices. Hence, the entangling of qubits is done with a generalized CZ operation involving multiple qubits. The protocol itself is similar to the one from [31], as the prover is required to prepare many copies of a hypergraph state and send them to the verifier. The verifier will then test all but one of these states using stabilizer measurements and use the remaining one to perform the MBQC computation. For a computation, $\mathcal{C}$, the protocol has completeness lower bounded by $1 - |\mathcal{C}|e^{-|\mathcal{C}|}$ and soundness upper bounded by $1/\sqrt{|\mathcal{C}|}$. The communication complexity is higher than the previous measurement-only protocol, as the prover needs to send $O(|\mathcal{C}|^{21})$ copies

of the $O(|\mathcal{C}|)$-qubit graph state, leading to a total communication cost of $O(|\mathcal{C}|^{22})$. We end with the following result:

Theorem 6 *The measurement-only protocols are receive-and-measure* QPIP *protocols having an inverse polynomial gap between completeness and soundness.*

3.2 Post Hoc Verification

The protocols we have reviewed so far have all been based on cryptographic primitives. There were reasons to believe, in fact, that any quantum verification protocol would have to use some form of encryption or hiding. This is due to the parallels between verification and authentication, which were outlined in Section 2. However, it was shown that this is not the case when Morimae and Fitzsimons, and independently Hangleiter et al, proposed a protocol for *post hoc* quantum verification [29, 30]. The name "post hoc" refers to the fact that the protocol is not interactive, requiring a single round of back and forth communication between the prover and the verifier. Moreover, verification is performed after the computation has been carried out. It should be mentioned that the first post hoc protocol was proposed in [22], by Fitzsimons and Hajdušek, however, that protocol utilizes multiple quantum provers, and we review it in Section 4.3.

In this section, we will present the post hoc verification approach, refered to as *1S-Post-hoc*, from the perspective of the Morimae and Fitzsimons paper [29]. The reason for choosing their approach, over the Hangleiter et al one, is that the entanglement-based post hoc protocols, from Section 4.3, are also described using similar terminology to the Morimae and Fitzsimons paper. The protocol of Hangleiter et al is essentially identical to the Morimae and Fitzsimons one, except it is presented from the perspective of certifying the ground state of a gapped, local Hamiltonian. Their certification procedure is then used to devise a verification protocol for a class of quantum simulation experiments, with the purpose of demonstrating a quantum computational advantage [30].

The starting point is the complexity class QMA, for which we have stated the definition in Section 1. Recall, that one can think of QMA as the class of problems for which the solution can be checked by a BQP verifier receiving a quantum state $|\psi\rangle$, known as a witness, from a prover. We also stated the definition of the k-local Hamiltonian problem, a complete problem for the class QMA, in Definition 9. We mentioned that for $k = 2$ the problem is QMA-complete [64]. For the post hoc protocol, Morimae and Fitzsimons consider a particular type of 2-local Hamiltonian known as an XZ-Hamiltonian.

To define an XZ-Hamiltonian we introduce some helpful notation. Consider an n-qubit operator S, which we shall refer to as XZ-term, such that $S = \bigotimes_{j=1}^{n} P_j$, with $P_j \in \{I, \mathsf{X}, \mathsf{Z}\}$. Denote $w_X(S)$ as the X-weight of S, representing the total number of j's for which $P_j = \mathsf{X}$. Similarly denote $w_Z(S)$ as the Z-weight for S. An XZ-Hamiltonian is then a 2-local Hamiltonian of the form $H = \sum_i a_i S_i$, where the a_i's are real numbers and the S_i's are XZ-terms having $w_X(S_i) + w_Z(S_i) \leq 2$.

The 1S-Post-hoc protocol starts with the observation that BQP $\subseteq$ QMA. This means that any problem in BQP can be viewed as an instance of the 2-local Hamiltonian problem. Therefore, for any language $L \in$ BQP and input x, there exists an XZ-Hamiltonian, H, such that the smallest eigenvalue of H is less than a when $x \in L$ or larger than b, when $x \notin L$, where a and b are a pair of numbers satisfying $b - a \geq 1/poly(|x|)$. Hence, the lowest energy eigenstate of H (also referred to as *ground state*), denoted $|\psi\rangle$, is a quantum witness for $x \in L$. In a QMA protocol, the prover would be instructed to send this state to the verifier. The verifier then performs a measurement on $|\psi\rangle$ to estimate its energy, accepting if the estimate is below a and rejecting otherwise. However, we are interested in a verification protocol for BQP problems where the verifier has minimal quantum capabilities. This means that there will be two requirements: the verifier can only perform single-qubit measurements; the prover is restricted to BQP computations. The 1S-Post-hoc protocol satisfies both of these constraints.

The first requirement is satisfied because estimating the energy of a quantum state, $|\psi\rangle$, with respect to an XZ-Hamiltonian H, can be done by measuring one of the observables S_i on the state $|\psi\rangle$. Specifically, it is shown in [65] that if one chooses the local term S_i according to a probability distribution given by the normalized terms $|a_i|$, and measures $|\psi\rangle$ with the S_i observables, this provides an estimate for the energy of $|\psi\rangle$. Since H is an XZ-Hamiltonian, this entails performing at most two measurements, each of which can be either an X measurement or a Z measurement. This implies that the verifier need only perform single-qubit measurements.

For the second requirement, one needs to show that for any BQP computation, there exists an XZ-Hamiltonian such that the ground state can be prepared by a polynomial-size quantum circuit. Suppose the computation that the verifier would like to delegate is denoted as $\mathcal{C}$ and the input for this computation is x. Given what we have mentioned above, regarding the local Hamiltonian problem, it follows that there exists an XZ-Hamiltonian H and numbers a and b, with $b - a \geq 1/poly(|x|)$, such that if $\mathcal{C}$ accepts x with high probability then the ground state of H has energy below a, otherwise it has energy above b. It was shown in [64, 66, 67], that starting from $\mathcal{C}$ and x one can construct an XZ-Hamiltonian satisfying this property and which also has a ground state that can be prepared by a BQP machine. The ground state is known as the *Feynman-Kitaev clock state*. To describe this state, suppose the circuit $\mathcal{C}$ has T gates (i.e. $T = |\mathcal{C}|$) and that these gates, labelled in the order in which they are applied, are denoted $\{U_i\}_{i=0}^{T}$. For $i = 0$ we assume $U_0 = I$. The Feynman-Kitaev state is the following:

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t U_{t-1} ... U_0 |x\rangle \left| 1^t 0^{T-t} \right\rangle \tag{62}$$

This is essentially a superposition over all time steps, t, of the time evolved state in the circuit $\mathcal{C}$. Hence, the state can be prepared by a BQP machine. The XZ-Hamiltonian is then a series of 2-local constraints that are all simultaneously satisfied by this state.

We can now present the steps of the 1S-Post-hoc protocol:

(1) The verifier chooses a quantum circuit, $\mathcal{C}$, and an input x to delegate to the prover.

(1) The verifier computes the terms a_i of the XZ-Hamiltonian, $H = \sum_i a_i S_i$, having as a ground state the Feynman-Kitaev state asswith $\mathcal{C}$ and x, denoted $|\psi\rangle$.
(2) The verifier instructs the prover to send her $|\psi\rangle$, qubit by qubit.
(4) The verifier chooses one of the XZ-terms S_i, according to the normalized distribution $\{|a_i|\}_i$, and measures it on $|\psi\rangle$. She accepts if the measurement indicates the energy of $|\psi\rangle$ is below a.

Note that the protocol is not blind, since the verifier informs the prover about both the computation $\mathcal{C}$ and the input x.

As mentioned, the essential properties that any QPIP protocol should satisfy are completeness and soundness. For the post hoc protocol, these follow immediately from the local Hamiltonian problem. Specifically, we know that there exist a and b such that $b - a \geq 1/poly(|x|)$. When $\mathcal{C}$ accepts x with high probability, the state $|\psi\rangle$ will be an eigenstate of H having eigenvalue smaller than a. Otherwise, any state, when measured under the H observable, will have an energy greater than b. Of course, the verifier is not computing the exact energy $|\psi\rangle$ under H, merely an estimate. This is because she is measuring only one local term from H. However, it is shown in [29] that the precision of her estimate is also inverse polynomial in $|x|$. Therefore:

Theorem 7 *1S-Post-hoc is a receive-and-measure* QPIP *protocol having an inverse polynomial gap between completeness and soundness.*

The only quantum capability of the verifier is the ability to measure single qubits in the computational and Hadamard bases (i.e. measuring the Z and X observables). The protocol, as described, suggests that it is sufficient for the verifier to measure only two qubits. However, since the completeness-soundness gap decreases with the size of the input, in practice one would perform a sequential repetition of this protocol in order to boost this gap. It is easy to see that, for a protocol with a completeness-soundness gap of $1/p(|x|)$, for some polynomial p, in order to achieve a constant gap of at least $1-\epsilon$, where $\epsilon > 0$, the protocol needs to be repeated $O(p(|x|) \cdot log(1/\epsilon))$ times. It is shown in [30, 68] that $p(|x|)$ is $O(|\mathcal{C}|^2)$, hence the protocol should be repeated $O(|\mathcal{C}|^2 \cdot log(1/\epsilon))$ times and this also gives us the total number of measurements for the verifier.[33] Note, however, that this assumes that each run of the protocol is independent of the previous one (in other words, that the states sent by the prover to the verifier in each run are uncorrelated). Therefore, the $O(|\mathcal{C}|^2 \cdot log(1/\epsilon))$ overhead should be taken as an i.i.d. (independent and identically distributed states) estimate. This is, in fact, mentioned explicitly in the Hangleiter et al result, where they explain

[33] As a side note, the total number of measurements is not the same as the communication complexity for this protocol, since the prover would have to send $O(|\mathcal{C}|^3 \cdot log(1/\epsilon))$ qubits in total. This is because, for each repetition, the prover sends a state of $O(|\mathcal{C}|)$ qubits, but the verifier only measures 2 qubits from each such state.

that the prover should prepare "*a number of independent and identical copies of a quantum state*" [30]. Thus, when considering the most general case of a malicious prover that does not obey the i.i.d. constraint, one requires a more thorough analysis involving non-independent runs, as is done in the measurement-only protocol [31] or the steering-based VUBQC protocol [33].

3.3 Summary of Receive-and-Measure Protocols

Receive-and-measure protocols are quite varied in the way in which they perform verification. The measurement-only protocols use stabilizers to test that the prover prepared a correct graph state and then has the verifier use this state to perform an MBQC computation. The 1S-Post-hoc protocol relies on the entirely different approach of estimating the ground state energy of a local Hamiltonian. Lastly, the steering-based VUBQC protocol, which we detail in Section 4.1, is different from these other two approaches by having the verifier remotely prepare the VUBQC states on the prover's side and then doing trap-based verification. Having such varied techniques leads to significant differences in the total number of measurements performed by the verifier, as we illustrate in Table 2.

Of course, the number of measurements is not the only metric we use in comparing the protocols. Another important aspect is how many observables the verifier should be able to measure. The 1S-Post-hoc protocol is optimal in that sense, since the verifier need only measure X and Z observables. Next is the hypergraph state measurement-only protocol which requires all three Pauli observables. Lastly, the other two protocols require the verifier to be able to measure the XY-plane observables X, Y, $(\mathsf{X}+\mathsf{Y})/\sqrt{2}$ and $(\mathsf{X}-\mathsf{Y})/\sqrt{2}$ plus the Z observable.

Finally, we compare the protocols in terms of blindness, which we have seen plays an important role in prepare-and-send protocols. For receive-and-measure protocols, the 1S-Post-hoc protocol is the only one that is not blind. While this is our first example of a verification protocol that does not hide the computation and input from the prover, it is not the only one. In the next section, we review two other post hoc protocols that are also not blind.

Table 2 Comparison of receive-and-measure protocols

Protocol	Measurements	Observables	Blind
Measurement-only	$O(N \cdot 1/\alpha \cdot 1/\epsilon^2)$	5	Y
Hypergraph measurement-only	$O(max(N, 1/\epsilon^2)^{22})$	3	Y
1S-Post-hoc	$O(N^2 \cdot log(1/\epsilon))$	2	N
Steering-based VUBQC	$O(N^{13} log(N) \cdot log(1/\epsilon))$	5	Y

We denote $N = |\mathcal{C}|$ to be the size of the delegated quantum computation. The number of measurements is computed for a target gap between completeness and soundness of $1-\epsilon$ for some constant $\epsilon > 0$. For the first measurement-only protocol, α denotes the confidence level of the verifier in the hypothesis test

4 Entanglement-Based Protocols

The protocols discussed in the previous sections have been either prepare-and-send or receive-and-measure protocols. Both types employ a verifier with some minimal quantum capabilities interacting with a single BQP prover. In this section we explore protocols which utilize multiple non-communicating provers that share entanglement and a fully classical verifier. The main idea will be for the verifier to distribute a quantum computation among many provers and verify its correct execution from correlations among the responses of the provers.

We classify the entanglement-based approaches as follows:

1. **Section 4.1** three protocols which make use of the CHSH game, the first one developed by Reichardt et al. [18], the second by Gheorghiu et al. [19] and the third by Hajdušek, Pérez-Delgado and Fitzsimons.
2. **Section 4.2** a protocol based on self-testing graph states, developed by McKague [21].
3. **Section 4.3** two post hoc protocols, one developed by Fitzsimons and Hajdušek [29] and another by Natarajan and Vidick [23].

Unlike the previous sections where, for the most part, each protocol was based on a different underlying idea for performing verification, entanglement-based protocols are either based on some form of rigid self-testing or on testing local Hamiltonians via the post hoc approach. In fact, as we will see, even the post hoc approaches employ self-testing. Of course, there are distinguishing features within each of these broad categories, but due to their technical specificity, we choose to label the protocols in this section by the initials of the authors.

Since self-testing plays such a crucial role in entanglement-based protocols, let us provide a brief description of the concept. The idea of self-testing was introduced by Mayers and Yao in [69], and is concerned with characterising the shared quantum state and observables of n non-communicating players in a *non-local game*. A non-local game is one in which a referee (which we will later identify with the verifier) will ask questions to the n players (which we will identify with the provers) and, based on their responses, decide whether they win the game or not. Importantly, we are interested in games where there is a quantum strategy that outperforms a classical strategy. By a classical strategy, we mean that the players can only produce local correlations.[34] Conversely, in a quantum strategy, the players are allowed to share entanglement in order to produce non-local correlations and achieve a higher win rate. Even so, there is a limit to how well the players can perform in the game. In other words, the optimal quantum strategy has a certain probability of winning the game, which may be less than 1. Self-testing results are concerned with non-local games in which the optimal quantum strategy is *unique*, up to local isometries on the

[34] To define local correlations, consider a setting with two players, Alice and Bob. Each player receives an input, x for Alice and y for Bob and produces an output, denoted a for Alice and b for Bob. We say that the players' responses are locally correlated if: $Pr(a, b|x, y) = \sum_\lambda Pr(a|x, \lambda)Pr(b|y, \lambda)Pr(\lambda)$. Where λ is known as a *hidden variable*. In other words, given this hidden variable, the players' responses depend only on their local inputs.

players' systems. This means that if the referee observes a near maximal win rate for the players, in the game, she can conclude that they are using the optimal strategy and can therefore characterise their shared state and their observables, up to a local isometries. More formally, we give the definition of self-testing, adapted from [70] and using notation similar to that of [23]:

Definition 4 (**Self-testing**) Let G denote a game involving n non-communicating players denoted $\{P_i\}_{i=1}^n$. Each player will receive a question from a set, Q and reply with an answer from a set A. Thus, each P_i can be viewed as a mapping from Q to A. There exists some condition establishing which combinations of answers to the questions constitutes a win for the game. Let $\omega^*(G)$ denote the maximum winning probability of the game for players obeying quantum mechanics.

The mappings P_i are implemented by a measurement strategy $S = (|\psi\rangle, \{O_i^j\}_{ij})$ consisting of a state $|\psi\rangle$ shared among the n players and local observables $\{O_i^j\}_j$, for each player P_i. We say that the game G **self-tests** the strategy S, with robustness $\epsilon = \epsilon(\delta)$, for some $\delta > 0$, if, for any strategy $\tilde{S} = (\left|\tilde{\psi}\right\rangle, \{\tilde{O}_i^j\}_{ij})$ achieving winning probability $\omega^*(G) - \epsilon$ there exists a local isometry $\Phi = \bigotimes_{i=1}^n \Phi_i$ and a state $|junk\rangle$ such that:

$$TD(\Phi(\left|\tilde{\psi}\right\rangle), |junk\rangle|\psi\rangle) \leq \delta \tag{63}$$

and for all j:

$$TD\left(\Phi\left(\bigotimes_{i=1}^n \tilde{O}_i^j \left|\tilde{\psi}\right\rangle\right), |junk\rangle \bigotimes_{i=1}^n O_i^j |\psi\rangle\right) \leq \delta \tag{64}$$

Note that TD denotes trace distance, and is defined in Section 1.

4.1 Verification Based on CHSH Rigidity

RUV Protocol In [71], Tsirelson gave an upper bound for the total amount of non-local correlations shared between two non-communicating parties, as predicted by quantum mechanics. In particular, consider a two-player game consisting of Alice and Bob. Alice is given a binary input, labelled a, and Bob is given a binary input, labelled b. They each must produce a binary output and we label Alice's output as x and Bob's output as y. Alice and Bob win the game iff $a \cdot b = x \oplus y$. The two are *not* allowed to communicate during the game, however they are allowed to share classical or quantum correlations (in the form of entangled states). This defines a non-local game known as the *CHSH game* [72]. The optimal *classical* strategy for winning the game achieves a success probability of 75%, whereas, what Tsirelson proved, is that any *quantum* strategy achieves a success probability of at most $cos^2(\pi/8) \approx 85.3\%$. This maximal winning probability, in the quantum case, can in fact be achieved by having Alice and Bob do the following. First, they will share the state $|\Phi_+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$. If Alice receives input $a = 0$, then she will measure the Pauli X observable on her half of the $|\Phi_+\rangle$ state, otherwise (when $a = 1$) she measures the Pauli Z observable. Bob, on input $b = 0$ measures $(\mathsf{X} + \mathsf{Z})/\sqrt{2}$, on his half of the Bell pair,

and on input $b = 1$, he measures $(\mathsf{X} - \mathsf{Z})/\sqrt{2}$. We refer to this strategy as the *optimal quantum strategy* for the CHSH game.

McKague, Yang and Scarani proved a converse of Tsierlson's result, by showing that if one observes two players winning the CHSH game with a near $cos^2(\pi/8)$ probability, then it can be concluded that the players' shared state is close to a Bell pair and their observables are close to the ideal observables of the optimal strategy (Pauli X and Z, for Alice, and $(\mathsf{X} + \mathsf{Z})/\sqrt{2}$ and $(\mathsf{X} - \mathsf{Z})/\sqrt{2}$, for Bob) [73]. This is effectively a self-test for a Bell pair. Reichardt, Unger and Vazirani then proved a more general result for self-testing a *tensor product* of multiple Bell states as well as the observables acting on these states [18].[35] It is this latter result that is relevant for the RUV protocol so we give a more formal statement for it:

Theorem 8 *Suppose two players, Alice and Bob, are instructed to play n sequential CHSH games. Let the inputs, for Alice and Bob, be given by the n-bit strings* $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$. *Additionally, let* $S = (\left|\tilde{\psi}\right\rangle, \tilde{A}(\mathbf{a}), \tilde{B}(\mathbf{b}))$ *be the strategy employed by Alice and Bob in playing the n CHSH games, where* $\left|\tilde{\psi}\right\rangle$ *is their shared state and* $\tilde{A}(\mathbf{a})$ *and* $\tilde{B}(\mathbf{b})$ *are their respective observables, for inputs* $\mathbf{a}, \mathbf{b}$.

Suppose Alice and Bob win at least $n(1-\epsilon)cos^2(\pi/8)$ *games, with* $\epsilon = poly(\delta, 1/n)$ *for some* $\delta > 0$, *such that* $\epsilon \to 0$ *as* $\delta \to 0$ *or* $n \to \infty$. *Then, there exist a local isometry* $\Phi = \Phi_A \otimes \Phi_B$ *and a state* $|junk\rangle$ *such that:*

$$TD(\Phi(\left|\tilde{\psi}\right\rangle), |junk\rangle|\Phi_+\rangle^{\otimes n}) \leq \delta \tag{65}$$

and:

$$TD\left(\Phi\left(\tilde{A}(\mathbf{a}) \otimes \tilde{B}(\mathbf{b})\left|\tilde{\psi}\right\rangle\right), |junk\rangle A(\mathbf{a}) \otimes B(\mathbf{b})|\Phi_+\rangle^{\otimes n}\right) \leq \delta \tag{66}$$

where $A(\mathbf{a}) = \bigotimes_{i=1}^{n} P(\mathbf{a}(\mathbf{i}))$, $B(\mathbf{b}) = \bigotimes_{i=1}^{n} Q(\mathbf{b}(\mathbf{i}))$ *and* $P(0) = \mathsf{X}$, $P(1) = \mathsf{Z}$, $Q(0) = (\mathsf{X} + \mathsf{Z})/\sqrt{2}$, $Q(1) = (\mathsf{X} - \mathsf{Z})/\sqrt{2}$.

What this means is that, up to a local isometry, the players share a state which is close in trace distance to a tensor product of Bell pairs and their measurements are close to the ideal measurements. This result, known as *CHSH game rigidity*, is the key idea for performing multi-prover verification using a classical verifier. We will refer to the protocol in this section as the *RUV protocol*.

Before giving the description of the protocol, let us first look at an example of *gate teleportation*, which we also mentioned when presenting the Poly-QAS VQC protocol of Section 2.1. Suppose two parties, Alice and Bob, share a Bell state $|\Phi_+\rangle$. Bob applies a unitary U on his share of the entangled state so that the joint state becomes

[35] Note that the McKague, Yang and Scarani result could also be used to certify a tensor product of Bell pairs, by repeating the self-test of a single Bell pair multiple times. However, this would require each repetition to be independent of the previous one. In other words the states shared by Alice and Bob, as well as their measurement outcomes, should be independent and identically distributed (i.i.d.) in each repetition. The Reichardt, Unger and Vazirani result makes no such assumption.

$(I \otimes U)\,|\Phi_+\rangle$. Alice now takes an additional qubit, labelled $|\psi\rangle$ and measures this qubit and the one from the $|\Phi_+\rangle$ state in the Bell basis given by the states:

$$|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad |\Phi_-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$
$$|\Psi_+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad |\Psi_-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

The outcome of this measurement will be two classical bits which we label b_1 and b_2. After the measurement, the state on Bob's system will be $U\mathsf{X}^{b_1}\mathsf{Z}^{b_2}\,|\psi\rangle$. Essentially, Bob has a one-time padded version of $|\psi\rangle$ with the U gate applied.

We now describe the RUV protocol. It uses two quantum provers but can be generalized to any number of provers greater than two. Suppose that Alice and Bob are the two provers. They are allowed to share an unbounded amount of quantum entanglement but are not allowed to communicate during the protocol. A verifier will interact classically with both of them in order to delegate and check an arbitrary quantum computation specified by the quantum circuit $\mathcal{C}$. The protocol consists in alternating randomly between four sub-protocols:

- **CHSH games.** In this subprotocol, the verifier will simply play CHSH games with Alice and Bob. To be precise, the verifier will repeatedly instruct Alice and Bob to perform the ideal measurements of the CHSH game. She will collect the answers of the two provers (which we shall refer to as CHSH statistics) and after a certain number of games, will compute the win rate of the two provers. The verifier is interested in the case when Alice and Bob win close to the maximum number of games as predicted by quantum mechanics. Thus, at the start of the protocol she takes $\epsilon = poly(1/|\mathcal{C}|)$ and accepts the statistics produced by Alice and Bob if and only if they win at least a fraction $(1-\epsilon)cos^2(\pi/8)$ of the total number of games. Using the rigidity result, this implies that Alice and Bob share a state which is close to a tensor product of perfect Bell states (up to a local isometry). This step is schematically illustrated in Fig. 11.

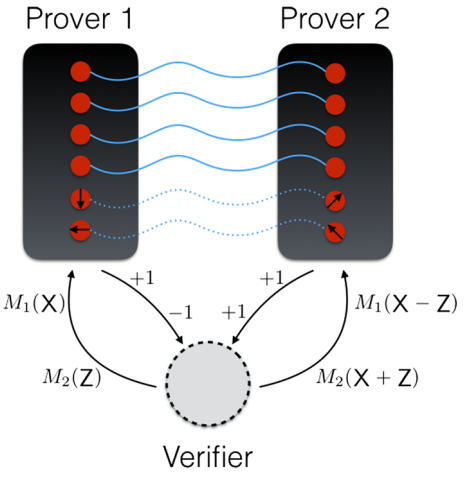


Fig. 11 Ideal CHSH game strategy

- **State tomography.** This time the verifier will instruct Alice to perform the ideal CHSH game measurements, as in the previous case. However, she instructs Bob to measure his halves of the entangled states so that they collapse to a set of *resource states* which will be used to perform gate teleportation. The resource states are chosen so that they are universal for quantum computation. Specifically, in the RUV protocol, the following resource states are used: $\{\mathsf{P}\,|0\rangle\,,(\mathsf{HP})_2\,|\Phi_+\rangle\,,(\mathsf{GY})_2\,|\Phi_+\rangle\,,\mathsf{CNOT}_{2,4}\mathsf{P}_2\mathsf{Q}_4(|\Phi_+\rangle\otimes|\Phi_+\rangle):\mathsf{P},\mathsf{Q}\in\{\mathsf{X},\mathsf{Y},\mathsf{Z},I\}\}$, where $\mathsf{G}=exp\left(-i\frac{\pi}{8}\mathsf{Y}\right)$ and the subscripts indicate on which qubits the operators act. Assuming Alice and Bob do indeed share Bell states, Bob's measurements will collapse Alice's states to the same resource states (up to a one-time padding known to the verifier). Alice's measurements on these states are used to check Bob's preparation, effectively performing state tomography on the resource states.
- **Process tomography.** This subprotocol is similar to the state tomography one, except the roles of Alice and Bob are reversed. The verifier instructs Bob to perform the ideal CHSH game measurements. Alice, on the other hand, is instructed to perform Bell basis measurements on pairs of qubits. As in the previous subprotocol, Bob's measurement outcomes are used to tomographically check that Alice is indeed performing the correct measurements.
- **Computation.** The final subprotocol combines the previous two. Bob is asked to perform the resource preparation measurements, while Alice is asked to perform Bell basis measurements. This effectively makes Alice perform the desired computation through repeated gate teleportation.

An important aspect, in proving the correctness of the protocol, is the local similarity of pairs of subprotocols. For instance, Alice cannot distinguish between the CHSH subprotocol and the state tomography one, or between the process tomography one and computation. This is because, in those situations, she is asked to perform the same operations on her side, while being unaware of what Bob is doing. Moreover, since the verifier can test all but the computation part, if Alice deviates there will be a high probability of her deviation being detected. The same is true for Bob. In this way, the verifier can, essentially, enforce that the two players behave honestly and thus perform the correct quantum computation. Note, that this is not the same as the blindness property, discussed in relation to the previous protocols. The RUV protocol does, however, posses that property as well. This follows from a more involved argument regarding the way in which the computation by teleportation is performed.

It should be noted that there are only two constraints imposed on the provers: that they cannot communicate once the protocol has commenced and that they produce close to quantum optimal win-rates for the CHSH games. Importantly, there are no constraints on the quantum systems possessed by the provers, which can be arbitrarily large. Similarly, there are no constraints on what measurements they perform or what strategy they use in order to respond to the verifier. In spite of this, the rigidity result shows that for the provers to produce statistics that are accepted by the verifier, they must behave according to the ideal strategy (up to local isometry). Having the ability to fully characterise the prover's shared state and their strategies in this way is what

allows the verifier to check the correctness of the delegated quantum computation. This approach, of giving a full characterisation of the states and observables of the provers, is a powerful technique which is employed by all the other entanglement-based protocols, as we will see.

In terms of practically implementing such a protocol, there are two main considerations: the amount of communication required between the verfier and the provers and the required quantum capabilities of the provers. For the latter, it is easy to see that the RUV protocol requires both provers to be universal quantum computers (i.e. BQP machines), having the ability to store multiple quantum states and perform quantum circuits on these states. In terms of the communication complexity, since the verifier is restricted to BPP, the amount of communication must scale polynomially with the size of the delegated computation. It was computed in [19], that this communication complexity is of the order $O(|\mathcal{C}|^c)$, with $c > 8192$. Without even considering the constant factors involved, this scaling is far too large for any sort of practical implementation in the near future.[36]

There are essentially two reasons for the large exponent in the scaling of the communication complexity. The first, as mentioned by the authors, is that the bounds derived in the rigidity result are not tight and could possibly be improved. The second and, arguably more important reason, stems from the rigidity result itself. In Theorem 8, notice that $\epsilon = poly(\delta, 1/n)$ and $\epsilon \to 0$ as $n \to \infty$. We also know that the provers need to win a fraction $(1 - \epsilon)cos^2(\pi/8)$ of CHSH games, in order to pass the verifier's checks. Thus, the completeness-soundness gap of the protocol will be determined by ϵ. But since, for fixed δ, ϵ is essentially inverse polynomial in n, the completeness-soundness gap will also be inverse polynomial in n. Hence, one requires polynomially many repetition in order to boost the gap to constant.

We conclude with:

Theorem 9 *The RUV protocol is an* MIP* *protocol achieving an inverse polynomial gap between completeness and soundness.*

GKW Protocol As mentioned, in the RUV protocol the two quantum provers must be universal quantum computers. One could ask whether this is a necessity or whether there is a way to reduce one of the provers to be non-universal. In a paper by Gheorghiu, Kashefi and Wallden it was shown that the latter option is indeed possible. This leads to a protocol which we shall refer to as the *GKW protocol*. The protocol is based on the observation that one could use the state tomography subprotocol of RUV in such a way so that one prover is remotely preparing single qubit states for the other prover. The preparing prover would then only be required to perform single qubit measurements and, hence, not need the full capabilities of a universal quantum computer. The specific single qubit states that are chosen, can be the ones used in the VUBQC protocol of Section 2.2. This latter prover can then be instructed to

[36] However, with added assumptions (such as i.i.d. states and measurement statistics for the two provers), the scaling can become small enough that experimental testing is possible. A proof of concept experiment of this is realized in [74].

perform the VUBQC protocol with these states. Importantly, because the provers are not allowed to communicate, this would preserve the blindness requirement of VUBQC. We will refer to the preparing prover as the *sender* and the remaining prover as the *receiver*. Once again, we assume the verifier wishes to delegate to the provers the evaluation of some quantum circuit $\mathcal{C}$.

The protocol, therefore, has a two-step structure:

(1) **Verified preparation.** This part is akin to the state tomography subprotocol of RUV. The verifier is trying to certify the correct preparation of states $\{|+_\theta\rangle\}_\theta$ and $|0\rangle$, $|1\rangle$, where $\theta \in \{0, \pi/4, ..., 7\pi/4\}$. Recall that these are the states used in VUBQC. We shall refer to them as the *resource states*. This is done by self-testing a tensor product of Bell pairs and the observables of the two provers using CHSH games and the rigidity result of Theorem 8.[37] As in the RUV protocol, the verifier will play multiple CHSH games with the provers. This time, however, each game will be an extended CHSH game (as defined in [18]) in which the verifier will ask each prover to measure an observable from the set $\{\mathsf{X}, \mathsf{Y}, \mathsf{Z}, (\mathsf{X} \pm \mathsf{Z})/\sqrt{2}, (\mathsf{Y} \pm \mathsf{Z})/\sqrt{2}, (\mathsf{X} \pm \mathsf{Y})/\sqrt{2}\}$. Alternatively, this can be viewed as the verifier choosing to play one of 6 possible CHSH games defined by the observables in that set[38] These observables are sufficient for obtaining the desired resource states. In particular, measuring the X, Y, and $(\mathsf{X} \pm \mathsf{Y})/\sqrt{2}$ observables on the Bell pairs will collapse the entangled qubits to states of the form $\{|+_\theta\rangle\}_\theta$, while measuring Z will collapse them to $|0\rangle$, $|1\rangle$. The verifier accepts if the provers win a fraction $(1 - \epsilon)cos^2(\pi/8)$ of the CHSH games, where $\epsilon = poly(\delta, 1/|\mathcal{C}|)$, and $\delta > 0$ is the desired trace distance between the reduced state on the receiver's side and the ideal state consisting of the required resource states in tensor product, up to a local isometry ($\epsilon \to 0$ as $\delta \to 0$ or $|\mathcal{C}| \to \infty$). The verifier will also instruct the sender prover to perform additional measurements so as to carry out the remote preparation on the receiver's side. This verified preparation is illustrated in Fig. 12.

(2) **Verified computation.** This part involves verifying the actual quantum computation, $\mathcal{C}$. Once the resource states have been prepared on the receiver's side, the verifier will perform the VUBQC protocol with that prover as if she had sent him the resource states. She accepts the outcome of the computation if all trap measurements succeed, as in VUBQC.

Note the essential difference, in terms of the provers' requirements, between this protocol and the RUV protocol. In the RUV protocol, both provers had to perform entangling measurements on their side. However, in the GKW protocol, the sender prover is required to only perform single qubit measurements. This means that the sender prover can essentially be viewed as an untrusted measurement device, whereas the receiver is the only universal quantum computer. For this reason, the

[37] In fact, what is used here is a more general version of Theorem 8 involving an extended CHSH game. See the appendix section of [18].

[38] For instance, one game would involve Alice measuring either X or Y, whereas Bob should measure $(\mathsf{X} + \mathsf{Y})/\sqrt{2}$ or $(\mathsf{X} - \mathsf{Y})/\sqrt{2}$. Similar games can be defined by suitably choosing observables from the given set.

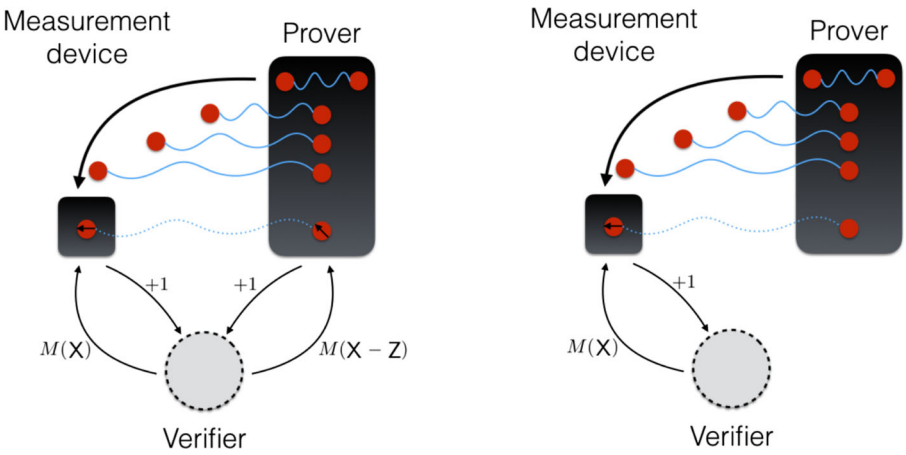


Fig. 12 Verified preparation

GKW protocol is also described as a *device-independent* [75, 76] verification protocol. This stems from comparing it to VUBQC or the receive-and-measure protocols, of Section 3, where the verifier had a trusted preparation or measurement device. In this case, the verifier essentially has a measurement device (the sender prover) which is untrusted.

Of course, performing the verified preparation subprotocol and combining it with VUBQC raises some questions. For starters, in the VUBQC protocol, the state sent to the prover is assumed to be an ideal state (i.e. an exact tensor product of states of the form $|+_\theta\rangle$ or $|0\rangle$, $|1\rangle$). However, in this case the preparation stage is probabilistic in nature and therefore the state of the receiver will be δ-close to the ideal tensor product state, for some $\delta > 0$. How is the completeness-soundness gap of the VUBQC protocol affected by this? Stated differently, is VUBQC robust to deviations in the input state? A second aspect is that, since the resource state is prepared by the untrusted sender, even though it is δ-close to ideal, it can, in principle, be correlated with the receiving prover's system. Do these initial correlations affect the security of the protocol?

Both of these issues are addressed in the proofs of the GKW protocol. Firstly, assume that in the VUBQC protocol the prover receives a state which is δ-close to ideal and uncorrelated with his private system. Any action of the prover can, in the most general sense, be modelled as a CPTP map. This CPTP map is of course distance preserving and so the output of this action will be δ-close to the output in the ideal case. It follows from this that the probabilities of the verifier accepting a correct or incorrect result change by at most $O(\delta)$. As long as $\delta > 1/poly(|\mathcal{C}|)$ (for a suitably chosen polynomial), the protocol remains a valid QPIP protocol.

Secondly, assume now that the δ-close resource state is correlated with the prover's private system, in VUBQC. It would seem that the prover could, in principle, exploit this correlation in order to convince the verifier to accept an incorrect outcome.

However, it is shown that this is, in fact, not the case, as long as the correlations are small. Mathematically, let ρ_{VP} be the state comprising of the resource state and the prover's private system. In the ideal case, this state should be a product state of the form $\rho_V \otimes \rho_P$, where $\rho_V = |\psi_{id}\rangle\langle\psi_{id}|$ is the ideal resource state and ρ_P the prover's system. However, in the general case the state can be entangled. In spite of this, it is known that:

$$TD(Tr_P(\rho_{VP}), |\psi_{id}\rangle\langle\psi_{id}|) \leq \delta \tag{67}$$

Using a result known as the *gentle measurement lemma* [18], one can show that this implies:

$$TD(\rho_{VP}, |\psi_{id}\rangle\langle\psi_{id}| \otimes Tr_V(\rho_{VP})) \leq O(\sqrt{\delta}) \tag{68}$$

In other words, the joint system of resource states and the prover's private memory is $O(\sqrt{\delta})$-close to the ideal system. Once again, as long as $\delta > 1/poly(|\mathcal{C}|)$ (for a suitably chosen polynomial), the protocol is a valid QPIP protocol.

These two facts essentially show that the GKW protocol is a valid entanglement-based protocol, as long as sufficient tests are performed in the verified preparation stage so that the system of resource states is close to the ideal resource states. As with the RUV protocol, this implies a large communication overhead, with the communication complexity being of the order $O(|\mathcal{C}|^c)$, where $c > 2048$. One therefore has:

Theorem 10 *The GKW protocol is an* MIP* *protocol achieving an inverse polynomial gap between completeness and soundness.*

Before concluding this section, we describe the steering-based VUBQC protocol that we referenced in Section 3. As mentioned, the GKW protocol can be viewed as a protocol involving a verifier with an untrusted measurement device interacting with a quantum prover. In a subsequent paper, Gheorghiu, Wallden and Kashefi addressed the setting in which the verifier's device becomes trusted [33]. They showed that one can define a self-testing game for Bell states which involves *steering correlations* [77] as opposed to non-local correlations. Steering correlations arise in a two-player setting in which one of the players is trusted to measure certain observables. This extra piece of information allows for the characterisation of Bell states with comparatively fewer statistics than in the non-local case. The steering-based VUBQC protocol, therefore, has exactly the same structure as the GKW protocol. First, the verifier uses this steering-based game, between her measurement device and the prover, to certify that the prover prepared a tensor product of Bell pairs. She then measures some of the Bell pairs so as to remotely prepare the resource states of VUBQC on the prover's side and then performs the trap-based verification. As mentioned in Section 3, the protocol has a communication complexity of $O(|\mathcal{C}|^{13} log(|\mathcal{C}|))$ which is clearly an improvement over $O(|\mathcal{C}|^{2048})$. This improvement stems from the trust added to the measurement device. However, the overhead is still too great for any practical implementation.

HPDF Protocol Independently from the GKW approach, Hajdušek, Pérez-Delgado and Fitzsimons developed a protocol which also combines the CHSH rigidity

result with the VUBQC protocol. This protocol, which we refer to as the *HPDF protocol* has the same structure as GKW in the sense that it is divided into a verified preparation stage and a verified computation stage. The major difference is that the number of non-communicating provers is on the order $O(poly(|\mathcal{C}|))$, where $\mathcal{C}$ is the computation that the verifier wishes to delegate. Essentially, there is one prover for each Bell pair that is used in the verified preparation stage. This differs from the previous two approaches in that the verifier knows, a priori, that there is a tensor product structure of states. She then needs to certify that these states are close, in trace distance, to Bell pairs. The advantage of assuming the existence of the tensor product structure, instead of deriving it through the RUV rigidity result, is that the overhead of the protocol is drastically reduced. Specifically, the total number of provers, and hence the total communication complexity of the protocol is of the order $O(|\mathcal{C}|^4 log(|\mathcal{C}|))$.

We now state the steps of the HPDF protocol. We will refer to one of the provers as the verifier's untrusted measurement device. This is akin to the sender prover in the GKW protocol. The remaining provers are the ones which will "receive" the states prepared by the verifier and subsequently perform the quantum computation.

(1) Verified preparation. The verifier is trying to certify the correct preparation of the resource states $\{|+_\theta\rangle\}_\theta$ and $|0\rangle, |1\rangle$, where $\theta \in \{0, \pi/4, ..., 7\pi/4\}$. The verifier instructs each prover to prepare a Bell pair and send one half to her untrusted measurement device. For each received state, she will randomly measure one of the following observables $\{\mathsf{X}, \mathsf{Y}, \mathsf{Z}, (\mathsf{X}+\mathsf{Z})/\sqrt{2}, (\mathsf{Y}+\mathsf{Z})/\sqrt{2}, (\mathsf{X}+\mathsf{Y})/\sqrt{2}, (\mathsf{X}-\mathsf{Y})/\sqrt{2}\}$. Each prover is either instructed to randomly measure an observable from the set $\{\mathsf{X}, \mathsf{Y}, \mathsf{Z}\}$ or to not perform any measurement at all. The latter case corresponds to the qubits which are prepared for the computation stage. The verifier will compute correlations between the measurement outcomes of her device and the provers and accept if these correlations are above some threshold parametrized by $\epsilon = poly(\delta, 1/|\mathcal{C}|)$ ($\epsilon \to 0$ as $\delta \to 0$ or $|\mathcal{C}| \to \infty$), where $\delta > 0$ is the desired trace distance between the reduced state on the receiving provers' sides and the ideal state consisting of the required resource states in tensor product, up to a local isometry.

(2) Verified computation. Assuming the verifier accepted in the previous stage, she instructs the provers that have received the resource states to act as a single prover. The verifier then performs the VUBQC protocol with that prover as if she had sent him the resource states. She accepts the outcome of the computation if all trap measurements succeed, as in VUBQC.

In their paper, Hajdušek et al have proved that the procedure in the verified preparation stage of their protocol constitutes a self-testing procedure for Bell states. This procedure self-tests individual Bell pairs, as opposed to the CHSH rigidity theorem which self-tests a tensor product of Bell pairs. In this case, however, the tensor product structure is already given by having the $O(|\mathcal{C}|^4 log(|\mathcal{C}|))$ non-communicating provers. The correctness of the verified computation stage follows from the robustness of the VUBQC protocol, as mentioned in the previous section. One therefore has the following:

Theorem 11 *The HPDF protocol is an* MIP*[poly] *protocol achieving an inverse polynomial gap between completeness and soundness.*

4.2 Verification Based on Self-Testing Graph States

We saw, in the HPDF protocol, that having multiple non-communicating provers presents a certain advantage in characterising the shared state of these provers, due to the tensor product structure of the provers' Hilbert spaces. This approach not only leads to simplified proofs, but also to a reduced overhead in characterising this state, when compared to the CHSH rigidity Theorem 8, from [18].

Another approach which takes advantage of this tensor product structure is the one of McKague from [21]. In his protocol, as in HPDF, the verifier will interact with $O(poly(|\mathcal{C}|))$ provers. Specifically, there are multiple groups of $O(|\mathcal{C}|)$ provers, each group jointly sharing a graph state $|G\rangle$. In particular, each prover should hold only one qubit from $|G\rangle$. The central idea is for the verifier to instruct the provers to measure their qubits to either test that the provers are sharing the correct graph state or to perform an MBQC computation of $\mathcal{C}$. This approach is similar to the stabilizer measurement-only protocol of Section 3.1 and, just like in that protocol or the Test-or-Compute or RUV protocols, the verifier will randomly alternate between tests and computation.

Before giving more details about this verification procedure, we first describe the type of graph state that is used in the protocol and the properties which allow this state to be useful for verification. McKague considers $|G\rangle$ to be a triangular graph state, which is a type of universal cluster state. What this means is that the graph G on which the state is based on, is a triangular lattice (a planar graph with triangular faces). An example is shown in Fig. 13. For each vertex v in G we have that:

$$X_v Z_{N(v)} |G\rangle = |G\rangle \tag{69}$$

Where $N(v)$ denotes the neighbors of the vertex v. In other words, $|G\rangle$ is stabilized by the operators $K_v = X_v Z_{N(v)}$, for all vertices v. This is important for the testing part of the protocol as it means that measuring the observable S_v will always yield the outcome 1. Another important property is:

$$X_\tau Z_{N(\tau)} |G\rangle = -|G\rangle \tag{70}$$

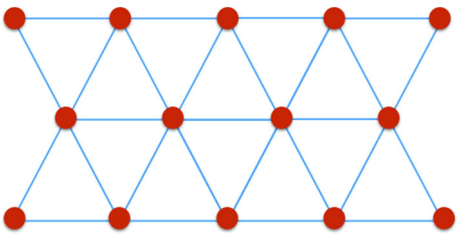


Fig. 13 Triangular lattice graph

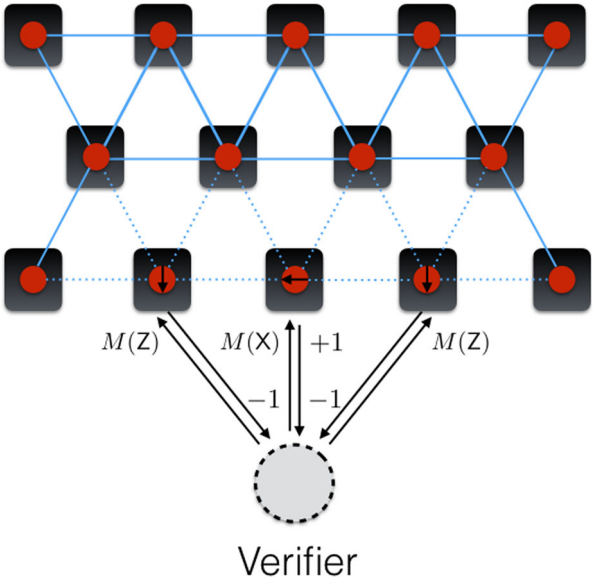


Fig. 14 Verifier instructing some of the provers to perform measurements in McKague's protocol

where τ is a set of 3 neighboring vertices which comprise a triangle in the graph G (and $N(\tau)$ are the *odd* neighbors of those vertices[39]). This implies that measuring $T_\tau = X_\tau Z_{N(\tau)}$ produces the outcome -1. Triangular graph states are universal for quantum computation, as explained in [21, 78], by performing local measurements (with corrections) on the vertex qubits using the observables $\mathsf{R}(\theta) = cos(\theta)\mathsf{X} + sin(\theta)\mathsf{Z}$, where $\theta \in \{0, \pi/4, ..., 7\pi/4\}$.

We now have the necessary elements to describe McKague's protocol. The verifier considers a triangular graph state $|G\rangle$ for the computation she wishes to verify. Let $n = O(|\mathcal{C}|)$ denote the number of vertices in G. In the ideal case, there will be multiple groups of n provers and, in each group, every prover should have one of the qubits of this graph (entangled with its neighbors). Denote T as the number of triangles (consisting of 3 neighboring vertices) in G and $N_G = 3n+T$. The protocol's setting is shown in Fig. 14.

The verifier will choose one of the n groups of provers at random to perform the computation $\mathcal{C}$. The computation is performed in an MBQC fashion. In other words, the verifier will pick appropriate measurement angles $\{\theta_v\}_{v\in V(G)}$, for all vertices in G, as well as a partial order for the vertices. To perform the computation $\mathcal{C}$, the verifier instructs the provers to measure the qubits of $|G\rangle$ with the observables $\mathsf{R}(\theta_v)$, defined above. The partial order establishes the temporal ordering of these measurements. Additionally, the θ_v angles, for the $\mathsf{R}(\theta_v)$ measurements, should be updated so as to account for corrections arising from previous measurement outcomes. In other words, the angles $\{\theta_v\}_{v\in V(G)}$, which we shall refer to as *computation angles*,

[39] In other words, $N(\tau)$ consists of those vertices that are connected to an odd number of vertices from τ.

are the ideal angles assuming no corrections. See Section 1 for more details about measurement-based quantum computations.

The remaining groups of provers are used to test that the correct graph state, $|G\rangle$, was prepared. This testing phase consists in the verifier randomly choosing to run one of the following sub-protocols:

- **Vertex test**, chosen with probability $\frac{n}{N_G}$. The verifier randomly picks a vertex v of the graph. She instructs the prover having the qubit corresponding to v to measure it in the X basis. At the same time, she asks the provers having the neighboring qubits, $N(v)$, to measure them in the Z basis. The ± 1 outcomes of the provers are multiplied by the verifier and she checks to see whether this product is 1 (since she is essentially measuring the S_v stabilizer). If this is the case, she accepts, otherwise she rejects.
- **Triangle test**, chosen with probability $\frac{T}{N_G}$. Similar to the previous test, the verifier picks a random triangle in G, consisting of the vertices τ. She instructs the provers having these vertices to measure them in the X basis, while their neighbors ($N(\tau)$) are instructed to measure in the Z basis. The verifier checks if the product of their outcomes is -1 and if so, she accepts, otherwise she rejects.
- $R(\theta)$ **test**, chosen with probability $\frac{2n}{N_G}$. In this case the verifier first randomly picks a vertex v of the graph, a neighbor u of v (so $u \in N(v)$) and t in $\{-1, +1\}$. She then randomly picks X with probability $p = \frac{cos(\theta_v)}{cos(\theta_v)+|sin(\theta_v)|}$ or Z with probability $1 - p$, where θ_v is the computation angle associated with v. If she chose X, then she queries the prover holding v to measure $\mathsf{R}(t\theta_v)$, and his neighbors ($N(v)$) to measure Z. She accepts if the product of their replies is $+1$. If the verifier instead chose Z, then she instructs the prover holding v to measure $t\mathsf{R}(t\theta_v)$, the prover holding u to measure X and the neighbors of u and v to measure Z. She accepts if the product of their outcomes is $+1$.

Together, these three tests are effectively performing a self-test of the graph state $|G\rangle$ and the prover's observables. Specifically, McKague showed the following:

Theorem 12 *For a triangular graph G, having n vertices, suppose that n provers, performing the strategy $S = (\left|\tilde{\psi}\right\rangle, \{\tilde{O}_i^j\}_{ij})$ succeed in the test described above with probability $1 - \epsilon$, where $\epsilon = poly(\delta, 1/n)$ for some $\delta > 0$ and $\epsilon \to 0$ as $\delta \to 0$ or $n \to \infty$. The strategy involves sharing the state $\left|\tilde{\psi}\right\rangle$ and measuring the observables $\{\tilde{O}_i^j\}_{ij}$, where each prover, i, has observables $\{\tilde{O}_i^j\}_j$. Then there exists a local isometry $\Phi = \bigotimes_{i=1}^n \Phi_i$ and a state $|junk\rangle$ such that:*

$$TD(\Phi(\left|\tilde{\psi}\right\rangle), |junk\rangle|G\rangle) \leq \delta \tag{71}$$

and for all j:

$$TD\left(\Phi\left(\bigotimes_{i=1}^n \tilde{O}_i^j \left|\tilde{\psi}\right\rangle\right), |junk\rangle \bigotimes_{i=1}^n O_i^j |G\rangle\right) \leq \delta \tag{72}$$

where for all i, $O_i^j \in \{\mathsf{R}(\theta) | \ \theta \in \{0, \pi/4, ..., 7\pi/4\}\}$.[40]

Note that the verifier will ask the provers to perform the same types of measurements in both the testing phase and the computation phase of the protocol. This means that, at the level of each prover, the testing and computation phases are indistinguishable. Moreover, the triangular state $|G\rangle$, being a universal cluster state, will be the same for computations of the same size. Therefore, the protocol is blind in the sense that each prover, on its own, is unaware of what computation is being performed. In summary, the protocol consists of the verifier choosing to perform one of the following:

- **Computation.** In this case, the verifier instructs the provers to perform the MBQC computation of $\mathcal{C}$ on the graph state $|G\rangle$, as described above.
- **Testing** $|G\rangle$**.** In this case, the verifier will randomly choose between one of the three tests described above accepting if an only if the test succeeds.

It is therefore the case that:

Theorem 13 *McKague's protocol is an* $\mathsf{MIP}^*[poly]$ *protocol having an inverse polynomial gap between completeness and soundness.*

As with the previous approaches, the reason for the inverse polynomial gap between completeness and soundness is the use of a self-test with robustness $\epsilon = poly(1/n)$ (and $\epsilon \to 0$ as $n \to \infty$). In turn, this leads to a polynomial overhead for the protocol as a whole. Specifically, McKague showed that the total number of required provers and communication complexity, for a quantum computation $\mathcal{C}$, is of the order $O(|\mathcal{C}|^{22})$. Note, however, that each of the provers must only perform a single-qubit measurement. Hence, apart from the initial preparation of the graph state $|G\rangle$, the individual provers are not universal quantum computers, merely single-qubit measurement devices.

4.3 Post Hoc Verification

In Section 3.2 we reviewed a protocol by Morimae and Fitzsimons for post hoc verification of quantum computation. Of course, that protocol involved a single quantum prover and a verifier with a measurement device. In this section, we review two post hoc protocols for the multi-prover setting having a classical verifier. We start with the first post hoc protocol by Fitzsimons and Hajdušek.

FH Protocol Similar to the 1S-Post-hoc protocol from Section 3.2, the protocol of Fitzsimons and Hajdušek, which we shall refer to as the *FH protocol*, also makes use of the local Hamiltonian problem stated in Definition 9. As mentioned, this problem is complete for the class QMA, which consists of problems that can be decided by a

[40]The measurement angles need not be restricted to this set, however, as in VUBQC, this set of angles is sufficient for performing universal MBQC computations.

BQP verifier receiving a witness state from a prover. Importantly, the size of the witness state is *polynomial* in the size of the input to the problem. However, Fitzsimons and Vidick proposed a protocol for the k-local Hamiltonian problem (and hence any QMA problem), involving 5 provers, in which the quantum state received by the verifier is of *constant* size [79]. That protocol is the basis for the FH protocol and so we start with a description of it.

Suppose that the k-local Hamiltonian is of the form $H = \sum_i H_i$, acting on a system of n qubits and each H_i is a k-local, n-qubit projector. For fixed a and b, such that $b - a \geq 1/poly(n)$, the verifier should accept if there exists a state $|\psi\rangle$ such that $\langle\psi|H|\psi\rangle \leq a$ and reject if for all states $|\psi\rangle$ it is the case that $\langle\psi|H|\psi\rangle \geq b$. Suppose we are in the acceptance case and let $|\psi\rangle$ be the witness state. The 5 provers must share a version of $|\psi\rangle$ encoded in the 5-qubit error correcting code,[41] denoted $|\psi\rangle_L$. Specifically, for each logical qubit of $|\psi\rangle_L$, each prover should hold one of its constituent physical qubits. The verifier will then check that the provers are indeed sharing this state, accepting if this is the case and rejecting otherwise. She will also perform an energy measurement on the state, to estimate if it has energy above b or below a. To do this she will, with equal probability, choose to either test that the shared state of the provers has energy below a or that the provers share a state encoded in the 5-qubit code:

- **Energy measurement.** In this case, the verifier will pick a random term H_i, from H, and ask each prover for k qubits corresponding to the logical states on which H_i acts. The verifier will then perform a two-outcome measurement, defined by the operators $\{H_i, I - H_i\}$ on the received qubits. As in the 1S-Post-hoc protocol, this provides an estimate for the energy of $|\psi\rangle$. The verifier accepts if the measurement outcome indicates the state has energy below a.
- **Encoding measurement.** In this case the verifier will choose at random between two subtests. In the first subtest, she will choose j at random from 1 to n and ask each prover to return the physical qubits comprising the j'th logical qubit. She then measures these qubits to check whether their joint state lies within the code space, accepting if it does and rejecting otherwise. In the second subtest, the verifier chooses a random set, S, of 3 values between 1 and n. She also picks one of the values at random, labelled j. The verifier then asks a randomly chosen prover for the physical qubits of the logical states indexed by the values in S, while asking the remaining provers for their shares of logical qubit j. As an example, if the set contains the values $\{1, 5, 8\}$, then the verifier picks one of the 5 provers at random and asks him for his shares (physical qubits) of logical qubits 1, 5 and 8 from $|\psi\rangle$. Assuming that the verifier also picked the random value 8 from the set, then she will ask the remaining provers for their shares of logical qubit 8. The verifier then measures logical qubit j (or 8, in our example) and checks if it is in the code subspace, accepting if it is and rejecting otherwise. The purpose of this second subtest is to guarantee that the provers respond with different qubits when queried.

[41] The 5-qubit code is the smallest error correcting capable of correcting for arbitrary single-qubit errors [80].

Table 3 Generators for 5-qubit code

Generator	Name
$I\mathsf{XZZX}$	g_1
$\mathsf{X}I\mathsf{XZZ}$	g_2
$\mathsf{ZX}I\mathsf{XZ}$	g_3
$\mathsf{ZZX}I\mathsf{X}$	g_4

One can see that when the witness state exists and the provers follow the protocol, the verifier will indeed accept with high probability. On the other hand, Fitzsimons and Vidick show that when there is no witness state, the provers will fail at convincing the verifier to accept with high probability. This is because they cannot simultaneously provide qubits yielding the correct energy measurements and also have their joint state be in the correct code space. This also illustrates why their protocol required testing both of these conditions. If one wanted to simplify the protocol, so as to have a single prover providing the qubits for the verifier's $\{H_i, I - H_i\}$ measurement, then it is no longer possible to prove soundness. The reason is that even if there does not exist a $|\psi\rangle$ having energy less than a for H, the prover could still find a group of k qubits which minimize the energy constraint for the specific H_i that the verifier wishes to measure. The second subtest prevents this from happening, with high probability, since it forces the provers to consistently provide the requested indexed qubits from the state $|\psi\rangle$.

Note that for a BQP computation, defined by the quantum circuit $\mathcal{C}$ and input x, the state $|\psi\rangle$ in the Fitzsimons and Vidick protocol becomes the Feynman-Kitaev state of that circuit, as described in Section 3.2. The FH protocol essentially takes the Fitzsimons and Vidick protocol for BQP computations and alters it by making the verifier classical. This is achieved using an approach of Ji [81] which allows for the two tests to be performed using only classical interaction with the provers. The idea is based on self-testing and is similar to the rigidity of the CHSH game. To understand this approach let us first examine the stabilizer generators, $\{g_i\}_{i=1}^4$ for the code space of the 5-qubit code, shown in Table 3. Notice that they all involve only Pauli X, Z or identity operators. In particular, the operator acting on the fifth qubit is always either X or Z. Ji then considers rotating this operator so that $\mathsf{X} \to \mathsf{X}'$ and $\mathsf{Z} \to \mathsf{Z}'$, where $\mathsf{X}' = (\mathsf{X}+\mathsf{Z})/\sqrt{2}$ and $\mathsf{Z}' = (\mathsf{X}-\mathsf{Z})/\sqrt{2}$, resulting in the new operators $\{g_i'\}_{i=1}^4$ shown in Table 4.

Table 4 Generators with fifth operator rotated

Generator	Name
$I\mathsf{XZZX}'$	g_1'
$\mathsf{X}I\mathsf{XZZ}'$	g_2'
$\mathsf{ZX}I\mathsf{XZ}'$	g_3'
$\mathsf{ZZX}I\mathsf{X}'$	g_4'

The new operators satisfy a useful property. For any state $|\phi\rangle$ in the code space of the 5-qubit code, it is the case that:

$$\sum_i \langle\phi|g_i'|\phi\rangle = 4\sqrt{2} \tag{73}$$

This is similar to the CHSH game. In the CHSH game, the ideal strategy involves Alice measuring either X or Z and Bob measuring either X' or Z', respectively, on the maximally entangled state $|\Phi_+\rangle$. These observables and the Bell state satisfy:

$$\langle\Phi_+|\mathsf{X}\mathsf{X}' + \mathsf{X}\mathsf{Z}' + \mathsf{Z}\mathsf{X}' - \mathsf{Z}\mathsf{Z}'|\Phi_+\rangle = 2\sqrt{2} \tag{74}$$

It can be shown that having observables which satisfy this relation implies that Alice and Bob win the CHSH game with the (quantum) optimal probability of success $cos^2(\pi/8)$. Analogous to the CHSH game, the stabilizers $\{g_i'\}_{i=1}^4$, viewed as observables, can be used to define a 5-player non-local game, in which the optimal quantum strategy involves measuring these observables on a state encoded in the 5-qubit code. Moreover, just like in the CHSH game, observing the players achieve the maximum quantum win-rate for the game implies that, up to local isometry, the players are following the ideal quantum strategy. We will not detail the game, except to say that it inolves partitioning the 5 provers into two sets, one consisting of four provers and the other with the remaining prover. Such a bipartition of a state encoded in the 5-qubit code yields a state which is isometric to a Bell pair. This means that the 5-player game is essentially self-testing a maximally entangled state, hence the similarity to the CHSH game. This then allows a classical verifier, interacting with the 5 provers, to perform the encoding test of the Fitzsimons and Vidick protocol.

We have discussed how a classical verifier can test that the 5 provers share a state encoded in the logical space of the 5-qubit code. But to achieve the functionality of the Fitzsimons and Vidick protocol, one needs to also delegate to the provers the measurement of a local term H_i from the Hamiltonian. This is again possible using the 5-player non-local game. Firstly, it can be shown that, without loss of generality, that each H_i, in the k-local Hamiltonian, can be expressed as a linear combination of terms comprised entirely of I, X and Z. This means that the Hamiltonian itself is a linear combination of such terms, $H = \sum_i a_i S_i$, where a_i are real coefficients and S_i are k-local XZ-terms. This is akin to the XZ-Hamiltonian from the 1S-Post-hoc protocol.

Given this fact, the verifier can measure one of the S_i terms, in order to estimate the energy of the ground state, instead of measuring $\{H_i, I - H_i\}$. She will pick an S_i term at random and ask the provers to measure the constituent Pauli observables in S_i. However, the verifier will also alternate these measurements with the stabilizer measurements of the non-local game, rejecting if the provers do not achieve the maximal non-local value of the game. This essentially forces the provers to perform the correct measurements (Fig. 15).

To summarize, the FH protocol is a version of the Fitzsimons and Vidick protocol which restricts the provers to be BQP machines and uses Ji's techniques, based on non-local games, to make the verifier classical. The steps of the FH protocol are as follows:

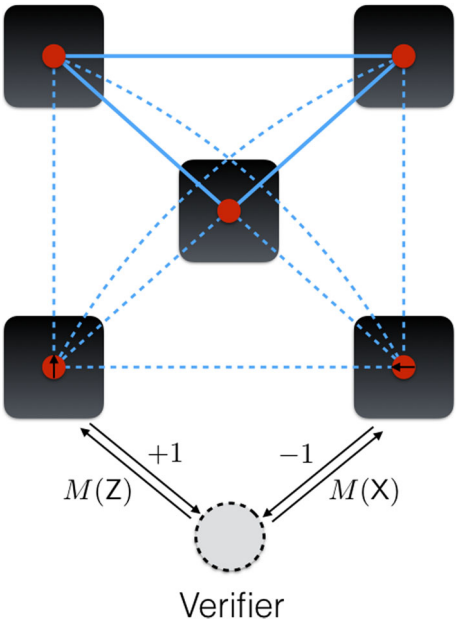


Fig. 15 Verifier interacting with the 5 provers

(1) The verifier instructs the provers to share the Feynman-Kitaev state, associated with her circuit $\mathcal{C}$, encoded in the 5-qubit error correcting code, as described above. We denote this state as $|\psi\rangle_L$. The provers are then split up and not allowed to communicate. The verifier then considers a k-local Hamiltonian having $|\psi_L\rangle$ as a ground state as well as the threshold values a and b, with $b - a > 1/poly(|\mathcal{C}|)$.

(2) The verifier chooses to either perform the energy measurement or the encoding measurement as described above. For the energy measurement she asks the provers to measure a randomly chosen XZ-term from the local Hamiltonian. The verifier accepts if the outcome indicates that the energy of $|\psi_L\rangle$ is below a. For the encoding measurement the verifier instructs the provers to perform the measurements of the 5-player non-local game. She accepts if the provers win the game, indicating that their shared state is correctly encoded.

One therefore has:

Theorem 14 *The FH protocol is an* MIP* *protocol achieving an inverse polynomial gap between completeness and soundness.*

There are two significant differences between this protocol and the previous entanglement-based approaches. The first is that the protocol does not use self-testing to enforce that the provers are performing the correct operations in order to implement the computation $\mathcal{C}$. Instead, the computation is checked indirectly by using the self-testing result to estimate the ground-state energy of the k-local Hamiltonian. This then provides an answer to the considered BQP computation viewed as a decision

problem.[42] The second difference is that the protocol is not blind. In all the previous approaches, the provers had to share an entangled state which was independent of the computation, up to a certain size. However, in the FH protocol, the state that the provers need to share depends on which quantum computation the verifier wishes to perform.

In terms of communication complexity, the protocol, as described, would involve only 2 rounds of interaction between the verifier and the provers. However, since the completeness-soundness gap is inverse polynomial, and therefore decreases with the size of the computation, it becomes necessary to repeat the protocol multiple times to properly differentiate between the accepting and rejecting cases. On the one hand, the local Hamiltonian itself has an inverse polynomial gap between the two cases of acceptance and rejection. As shown in [30, 68], for the Hamiltonian resulting from a quantum circuit, $\mathcal{C}$, that gap is $1/|\mathcal{C}|^2$. To boost this gap to constant, the provers must share $O(|\mathcal{C}|^2)$ copies of the Feynman-Kitaev state.

On the other hand, the self-testing result has an inverse polynomial robustness. This means that estimating the energy of the ground state is done with a precision which scales inverse polynomially in the number of qubits of the state. More precisely, according to Ji's result, the scaling should be $1/O(N^{16})$, where N is the number of qubits on which the Hamiltonian acts [81]. This means that the protocol should be repeated on the order of $O(N^{16})$ times, in order to boost the completeness-soundness gap to constant.

NV Protocol The second entanglement-based post hoc protocol was developed by Natarajan and Vidick [23] and we therefore refer to it as the *NV protocol*. The main ideas of the protocol are similar to those of the FH protocol. However, Natarajan and Vidick prove a self-testing result having *constant* robustness and use it in order to perform the energy estimation of the ground state for the local Hamiltonian.

The statement of their general self-testing result is too involved to state here, so instead we reproduce a corollary to their result (also from [23]) that is used for the NV protocol. This corollary involves self-testing a tensor product of Bell pairs:

Theorem 15 *For any integer n there exists a two-player non-local game, known as the Pauli braiding test (PBT), with $O(n)$-bit questions and $O(1)$-bit answers satisfying the following:*

Let $S = (\left|\tilde{\psi}\right\rangle, \tilde{A}(\mathbf{a}), \tilde{B}(\mathbf{b}))$ be the strategy employed by two players (Alice and Bob) in playing the game, where $\left|\tilde{\psi}\right\rangle$ is their shared state and $\tilde{A}(\mathbf{a})$ and $\tilde{B}(\mathbf{b})$ are their respective (multi-qubit) observables when given n-bit questions $\mathbf{a}$ and $\mathbf{b}$, respectively. Suppose Alice and Bob win the Pauli braiding test with probability $\omega^*(PBT) - \epsilon$, for some $\epsilon > 0$ (note that $\omega^*(PBT) = 1$). Then there exist $\delta = poly(\epsilon)$, a local isometry $\Phi = \Phi_A \otimes \Phi_B$ and a state $|junk\rangle$ such that:

$$TD(\Phi(\left|\tilde{\psi}\right\rangle), |junk\rangle|\Phi_+\rangle^{\otimes n}) \leq \delta \tag{75}$$

[42] In their paper, Fitzsimons and Hajdušek also explain how their protocol can be used to sample from a quantum circuit, rather than solve a decision problem [22].

$$TD\left(\Phi\left(\tilde{A}(\mathbf{a})\otimes\tilde{B}(\mathbf{b})\middle|\tilde{\psi}\right\rangle\right), |junk\rangle\mathsf{X}(\mathbf{a})\otimes\mathsf{Z}(\mathbf{b})|\Phi_+\rangle^{\otimes n}\right) \leq \delta \tag{76}$$

where $\mathsf{X}(\mathbf{a}) = \bigotimes_{i=1}^{n} X^{\mathbf{a(i)}}$ *and* $\mathsf{Z}(\mathbf{b}) = \bigotimes_{i=1}^{n} Z^{\mathbf{b(i)}}$.

This theorem is essentially a self-testing result for a tensor product of Bell states, and Pauli X and Z observables, achieving a constant robustness. The Pauli braiding test is used in the NV protocol in a similar fashion to Ji's result, from the previous subsection, in order to certify that a set of provers are sharing a state that is encoded in a quantum error correcting code. Again, this relies on a bi-partition of the provers into two sets, such that, an encoded state shared across the bi-partition is equivalent to a Bell pair.

Let us first explain the general idea of the Pauli braiding test for self-testing n Bell pairs and n-qubit observables. We have a referee that is interacting with two players, labelled Alice and Bob. The test consists of three subtests which are chosen at random by the referee. The subtests are:

- **Linearity test.** In this test, the referee will randomly pick a basis setting, W, from the set $\{\mathsf{X}, \mathsf{Z}\}$. She then randomly chooses two strings $\mathbf{a_1}, \mathbf{a_2} \in \{0,1\}^n$ and sends them to Alice. With equal probability, the referee takes $\mathbf{b_1}$ to be either $\mathbf{a_1}$, $\mathbf{a_2}$ or $\mathbf{a_1} \oplus \mathbf{a_2}$. She also randomly chooses a string $\mathbf{b_2} \in \{0,1\}^n$ and sends the pair $(\mathbf{b_1}, \mathbf{b_2})$ to Bob.[43] Alice and Bob are then asked to measure the observables $W(\mathbf{a_1})$, $W(\mathbf{a_2})$ and $W(\mathbf{b_1})$, $W(\mathbf{b_2})$, respectively, on their shared state. We denote Alice's outcomes as a_1, a_2 and Bob's outcomes as b_1, b_2. If $\mathbf{b_1} = \mathbf{a_1}$ (or $\mathbf{b_1} = \mathbf{a_2}$, respectively), the referee checks that $b_1 = a_1$ (or $b_1 = a_2$, respectively). If $\mathbf{b_1} = \mathbf{a_1} \oplus \mathbf{a_2}$, she checks that $b_1 = a_1 a_2$. This test is checking, on the one hand, that when Alice and Bob measure the same observables, they should get the same outcome (which is what should happen if they share Bell states). On the other hand, and more importantly, it is checking the commutation and linearity of their operators, i.e. that $W(\mathbf{a_1})W(\mathbf{a_2}) = W(\mathbf{a_2})W(\mathbf{a_1}) = W(\mathbf{a_1} + \mathbf{a_2})$ (and similarly for Bob's operators).
- **Anticommutation test.** The referee randomly chooses two strings $\mathbf{x}, \mathbf{z} \in \{0,1\}^n$, such that $\mathbf{x} \cdot \mathbf{z} = 1 \; mod \; 2$, and sends them to both players. These strings define the observables $\mathsf{X}(\mathbf{x})$ and $\mathsf{Z}(\mathbf{z})$ which are anticommuting because of the imposed condition on $\mathbf{x}$ and $\mathbf{z}$. The referee then engages in a non-local game with Alice and Bob designed to test the anticommutation of these observables for both of their systems. This can be any game that tests this property, such as the CHSH game or the *magic square* game, described in [82, 83]. As an example, if the referee chooses to play the CHSH game, then Alice will be instructed to measure either $\mathsf{X}(\mathbf{x})$ or $\mathsf{Z}(\mathbf{z})$ on her half of the shared state, while Bob would be instructed to measure either $(\mathsf{X}(\mathbf{x}) + \mathsf{Z}(\mathbf{z}))/\sqrt{2}$ or $(\mathsf{X}(\mathbf{x}) - \mathsf{Z}(\mathbf{z}))/\sqrt{2}$. The test is passed if the players achieve the win condition of the chosen anticommutation game. Note that for the case of the magic square game, the condition can be achieved with

[43] Note that pair can be either $(\mathbf{b_1}, \mathbf{b_2})$ or $(\mathbf{b_2}, \mathbf{b_1})$, so that Bob does not know which string is the one related to Alice's inputs.

probability 1 when the players implement the optimal quantum strategy. For this reason, if the chosen game is the magic square game, then $\omega^*(PBT) = 1$.

- **Consistency test.** This test combines the previous two. The referee randomly chooses a basis setting, $W \in \{\mathsf{X}, \mathsf{Z}\}$ and two strings $\mathbf{x}, \mathbf{z} \in \{0,1\}^n$. Additionally, let $\mathbf{w} = \mathbf{x}$, if $W = \mathsf{X}$ and $\mathbf{w} = \mathbf{z}$ if $W = \mathsf{Z}$. The referee sends W, $\mathbf{x}$ and $\mathbf{z}$ to Alice. With equal probability the referee will then choose to perform one of two subtests. In the first subtest, the referee sends $\mathbf{x}, \mathbf{z}$ to Bob as well and plays the anticommutation game with both, such that Alice's observable is $W(\mathbf{w})$. As an example, if $W = \mathsf{X}$ and the game is the CHSH game, then Alice would be instructed to measure $\mathsf{X}(\mathbf{x})$, while Bob is instructed to measure either $(\mathsf{X}(\mathbf{x}) + \mathsf{Z}(\mathbf{z}))/\sqrt{2}$ or $(\mathsf{X}(\mathbf{x}) - \mathsf{Z}(\mathbf{z}))/\sqrt{2}$. This subtest essentially mimics the anticommutation test and is passed if the players achieve the win condition of the game. In the second subtest, which mimics the linearity test, the referee sends W, $\mathbf{w}$ and a random string $\mathbf{y} \in \{0,1\}^n$ to Bob, instructing him to measure $W(\mathbf{w})$ and $W(\mathbf{y})$. Alice is instructed to measure $W(\mathbf{x})$ and $W(\mathbf{z})$. The test if passed if Alice and Bob obtain the same result for the $W(\mathbf{w})$ observable. For instance, if $W = \mathsf{X}$, then both Alice and Bob will measure $\mathsf{X}(\mathbf{x})$ and their outcomes for that measurement must agree.

Having observables that satisfy the linearity conditions of the test as well as the anticommutation condition implies that they are isometric to the actual X and Z observables acting on a maximally entangled state. This is what the Pauli braiding test checks and what is proven by the self-testing result of Natarajan and Vidick.

We can now describe the NV protocol. Similar to the FH protocol, for a quantum circuit, $\mathcal{C}$, and an input, x, one considers the associated Feynman-Kitaev state, denoted $|\psi\rangle$. This is then used to construct a 2-local XZ-Hamiltonian such that the ground state of this Hamiltonian is $|\psi\rangle$. As before, for some a and b, with $b - a > 1/poly(|\mathcal{C}|)$, when $\mathcal{C}$ accepts x we have that $\langle\psi|H|\psi\rangle < a$, otherwise $\langle\psi|H|\psi\rangle > b$. The verifier will instruct 7 provers to share a copy of $|\psi\rangle$ state, encoded in a 7-qubit quantum error correcting code known as *Steane's code*. The provers are then asked to perform measurements so as to self-test an encoded state or perform an energy measurement on this state. The code space, for Steane's code, is the 7-qubit subspace stabilized by all operators generated by $\{g_i\}_{i=1}^6$, where the generators are listed in Table 5.

The reason Natarajan and Vidick use this specific error correcting code is because it has two properties that are necessary for the application of their self-testing result.

Table 5 Generators for Steane's 7-qubit code

Generator	Name
$III\mathsf{XXXX}$	g_1
$I\mathsf{XX}II\mathsf{XX}$	g_2
$\mathsf{X}I\mathsf{X}I\mathsf{X}I\mathsf{X}$	g_3
$III\mathsf{ZZZZ}$	g_4
$I\mathsf{XX}II\mathsf{XZ}$	g_5
$\mathsf{Z}I\mathsf{Z}I\mathsf{Z}I\mathsf{Z}$	g_6

The first property is that each stabilizer generator is a tensor product of only the I, X and Z operators. This, of course, is true for the 5-qubit code as well. The second property is a symmetry condition: for each index $i \in \{1, ..., 6\}$, there exists a pair of stabilizer generators, S_{X} and S_{Z}, such that S_{X} consists exclusively of I and X operators and has an X on position i, whereas S_{Z} is identical to S_{X} but with all X operators replaced with Z. This property is not satisfied by the 5-qubit code and will allow the verifier to delegate to the provers measurements of the form $\mathsf{X}(\mathbf{x})$ and $\mathsf{Z}(\mathbf{z})$, where $\mathbf{x}$ and $\mathbf{z}$ are binary strings, as in the Pauli braiding test.

Putting everything together, the protocol works as follows. The verifier instructs the 7 provers to share an encoded instance of the Feynman-Kitaev state, $|\psi\rangle_L$, such that, for each logical qubit in $|\psi_L\rangle$, each prover will hold one of 7 the constituent physical qubits. She then chooses at random to perform one of the following:

- **Pauli braiding test.** The verifier chooses one of the 7 provers at random to be Alice, while the remaining provers will take on the role of Bob. The verifier then performs the Pauli braiding test with Alice and Bob in order to self-test the logical qubits in $|\psi_L\rangle$. As mentioned, each logical qubit, encoded in the 7 qubit code, is equivalent to a Bell pair under the chosen bi-partition. The Pauli braiding test is essentially checking that the provers have correctly encoded each of the qubits in $|\psi\rangle$ and that they are correctly measuring X and Z observables. The verifier rejects if the provers do not pass the test.
- **Energy test.** In this case, the verifier will pick an XZ-term, S, from H, at random, and instruct the provers to measure this term on their shared state. Note that S consists of *logical* X and Z operators. This means that each prover will need to perform local measurements so that their joint measurement acts as either X_L or Z_L, respectively. Additionally, X_L and Z_L, for the 7 qubit code, are expressed as tensor products of physical X and Z operations. This means that each prover will be instructed to measure an operators of the form $\mathsf{X}(\mathbf{x})$ and $\mathsf{Z}(\mathbf{z})$, on its physical qubits, where $\mathbf{x}, \mathbf{z} \in \{0, 1\}^N$, and N is the total number of logical qubits on which H acts. The product $\mathsf{X}(\mathbf{x})\mathsf{Z}(\mathbf{z})$ is the outcome for that prover's share of S. The verifier then takes all of these ± 1 outcomes and multiplies them together, thus obtaining the outcome of measuring S itself. She accepts if the outcome of the measurement indicates that the estimated energy of $|\psi\rangle$ is below a and rejects otherwise.
- **Energy consistency test.** This test is a combination of the previous two. As in the Pauli braiding test, the provers are bi-partitioned into two sets, one consisting of one prover, denoted Alice, and the other consisting of the other 6 provers, jointly denoted as Bob. The verifier now performs a test akin to the linearity test from Pauli braiding. She randomly chooses $W \in \{\mathsf{X}, \mathsf{Z}\}$, and let $\mathbf{w} = \mathbf{x}$, if $W = \mathsf{X}$ and $\mathbf{w} = \mathbf{z}$ if $W = \mathsf{Z}$. She also chooses $\mathbf{x}, \mathbf{z} \in \{0, 1\}^N$ according to the same distribution as in the energy test (i.e. as if she were instructing the provers to measure a random XZ-term from H). The verifier then does one of the following:
    - With probability $1/2$, instructs Alice to measure the observables $\mathsf{X}(\mathbf{x})$ and $\mathsf{Z}(\mathbf{z})$. Additionally, the verifier chooses $\mathbf{y} \in \{0, 1\}^N$ at random and instructs Bob to measure $W(\mathbf{y})$ and $W(\mathbf{y} \oplus \mathbf{w})$. If $W = \mathsf{X}$, the verifier accepts if the product of Bob's answers agrees with Alice's answer for

the $\mathsf{X}(\mathbf{x})$ observable. If $W = \mathsf{Z}$, the verifier accepts if the product of Bob's answers agrees with Alice's answer for the $\mathsf{Z}(\mathbf{z})$ observable. Note that this is the case since the product of Bob's observables should be $W(\mathbf{w})$ if he is behaving honestly.

- With probability $1/4$, instructs Alice to measure $W(\mathbf{y})$ and $W(\mathbf{v})$, where $\mathbf{y}, \mathbf{w} \in \{0,1\}^N$ are chosen at random. Bob is instructed to measure $W(\mathbf{y})$ and $W(\mathbf{y} \oplus \mathbf{w})$. The verifier accepts if the outcomes of Alice and Bob for $W(\mathbf{y})$ agree.
- With probability $1/4$, instructs Alice to measure $W(\mathbf{y} \oplus \mathbf{w})$ and $W(\mathbf{v})$, where $\mathbf{y}, \mathbf{w} \in \{0,1\}^N$ are chosen at random. Bob is instructed to measure $W(\mathbf{y})$ and $W(\mathbf{y} \oplus \mathbf{w})$. The verifier accepts if the outcomes of Alice and Bob for $W(\mathbf{y} \oplus \mathbf{w})$ agree.

The self-testing result guarantees that if these tests succeed, the verifier obtains an estimate for the energy of the ground state. Importantly, unlike the FH protocol, her estimate has constant precision. However, the protocol, as described up to this point, will still have an inverse polynomial completeness-soundness gap given by the local Hamiltonian. Recall that this is because the Feynman-Kitaev state will have energy below a when $\mathcal{C}$ accepts x with high probability, and energy above b otherwise, where $b - a > 1/|\mathcal{C}|^2$. But one can easily boost the protocol to a constant gap between completeness and soundness by simply requiring the provers to share $M = O(|\mathcal{C}|^2)$ copies of the ground state. This new state, $|\psi\rangle^{\otimes M}$, would then be the ground state of a new Hamiltonian H'.[44] One then runs the NV protocol for this Hamiltonian. It should be mentioned that this Hamiltonian is no longer 2-local, however, all of the tests in the NV protocol apply for these general Hamiltonians as well (as long as each term is comprised of I, X and Z operators, which is the case for H'). Additionally, the new Hamiltonian has a constant gap. The protocol therefore achieves a constant number of rounds of interaction with the provers (2 rounds) and we have that:

Theorem 16 *The NV protocol is an* MIP^* *protocol achieving a constant gap between completeness and soundness.*

To then boost the completeness-soundness gap to $1 - \epsilon$, for some $\epsilon > 0$, one can perform a parallel repetition of the protocol $O(log(1/\epsilon))$ times.

4.4 Summary of Entanglement-Based Protocols

We have seen that having non-communicating provers sharing entangled states allows for verification protocols with a classical client. What all of these protocols have in common is that they all make use of self-testing results. These essentially state that if a number of non-communicating players achieve a near optimal win rate in a

[44] Note that the state still needs to be encoded in the 7 qubit code.

non-local game, the strategy they employ in the game is essentially fixed, up to a local isometry. The strategy of the players consists of their shared quantum state as well as their local observables. Hence, self-testing results provide a precise characterisation for both.

This fact is exploited by the surveyed protocols in order to achieve verifiability. Specifically, we have seen that one approach is to define a number of non-local games so that by combining the optimal strategies of these games, the provers effectively perform a universal quantum computation. This is the approach employed by the RUV protocol [18]. Alternatively, the self-testing result can be used to check only for the correct preparation of a specific resource state. This resource state is then used by the provers to perform a quantum computation. How this is done depends on the type of resource state and on how the computation is delegated to the provers. For instance, one possibility is to remotely prepare the resource state used in the VUBQC protocol and then run the verification procedure of that protocol. This is the approach used by the GKW and HPDF protocols [19, 20]. Another possibility is to prepare a cluster state shared among many provers and then have each of those provers measure their states so as to perform an MBQC computation. This approach was used by McKague in his protocol [21]. Lastly, the self-tested resource state can be the ground state of a local Hamiltonian leading to the post hoc approaches employed by the FH and NV protocols.

We noticed that, depending on the approach that is used, there will be different requirements for the quantum operations of the provers. Of course, all protocols require that collectively the provers can perform BQP computations, however, individually some provers need not be universal quantum computers. Related to this is the issue of blindness. Again, based on what approach is used some protocols utilize blindness and some do not. In particular, the post hoc protocols are not blind since the computation and the input are revealed to the provers so that they can prepare the Feynman-Kitaev state.

We have also seen that the robustness of the self-testing game impacts the communication complexity of the protocol. Specifically, having robustness which is inverse polynomial in the number of qubits of the self-tested state, leads to an inverse polynomial gap between completeness and soundness. In order to make this gap constant, the communication complexity of the protocol has to be made polynomial. This means that most protocols will have a relatively large overhead, when compared to prepare-and-send or receive-and-measure protocols. Out of the surveyed protocols, the NV protocol is the only one which utilizes a self-testing result with constant robustness and therefore has a constant completeness-soundness gap. We summarize all of these facts in Table 6.[45]

[45] Note that for the HPDF protocol we assumed that there is one prover with quantum memory, comprised of the individual provers that come together in order to perform the MBQC computation at the end of the protocol. Since, to achieve a completeness-soundness gap of $1-\epsilon$. the protocol is repeated $O(log(1/\epsilon))$ times, this means there will be $O(log(1/\epsilon))$ provers with quantum memory in total.

Table 6 Comparison of entanglement-based protocols

Protocol	Provers	Qmem provers	Rounds	Communication	Blind
RUV	2	2	$O(N^{8192} \cdot log(1/\epsilon))$	$O(N^{8192} \cdot log(1/\epsilon))$	Y
McKague	$O(N^{22} \cdot log(1/\epsilon))$	0	$O(N^{22} \cdot log(1/\epsilon))$	$O(N^{22} \cdot log(1/\epsilon))$	Y
GKW	2	1	$O(N^{2048} \cdot log(1/\epsilon))$	$O(N^{2048} \cdot log(1/\epsilon))$	Y
HPDF	$O(N^4 log(N) \cdot log(1/\epsilon))$	$O(log(1/\epsilon))$	$O(N^4 log(N) \cdot log(1/\epsilon))$	$O(N^4 log(N) \cdot log(1/\epsilon))$	Y
FH	5	5	$O(N^{16} \cdot log(1/\epsilon))$	$O(N^{19} \cdot log(1/\epsilon))$	N
NV	7	7	$O(1)$	$O(N^3 \cdot log(1/\epsilon))$	N

We denote $N = |\mathcal{C}|$ to be the size of the delegated quantum computation together with the input to that computation. The listed values are given assuming a completeness-soundness gap of at least $1 - \epsilon$, for some $\epsilon > 0$. For the "Qmem provers" column, the numbers indicate how many provers need to have a quantum memory that is not of constant size, with respect to $|\mathcal{C}|$ (if we ignore the preparation of the initial shared entangled state). The "Rounds" column quantifies how many rounds of interaction are performed between the verifier and the provers, whereas "Communication" quantifies the total amount of communication (number of rounds times the size of the messages). Note that a similar table can be found in [24]

5 Outlook

5.1 Sub-Universal Protocols

So far we have presented protocols for the verification of universal quantum computations, i.e. protocols in which the provers are assumed to be BQP machines. In the near future, however, quantum computers might be more limited in terms of the type of computations that they can perform. Examples of this include the class of so-called *instantaneous quantum computations*, denoted IQP, *boson sampling* or the *one-pure qubit model* of quantum computation [1, 2, 84]. While not universal, these examples are still highly relevant since, assuming some plausible complexity theoretic conjectures hold, they could solve certain problems or sample from certain distributions that are intractable for classical computers. One is therefore faced with the question of how to verify the correctness of outcomes resulting from these models. In particular, when considering an interactive protocol, the prover should be restricted to the corresponding sub-universal class of problems and yet still be able to prove statements to a computationally limited verifier. We will see that many of the considered approaches are adapted versions of the VUBQC protocol from Section 2.2. It should be noted, however, that the protocols themselves are not direct applications of VUBQC. In each instance, the protocol was constructed so as to adhere to the constraints of the model.

The first sub-universal verification protocol is for the one-pure (or one-clean) qubit model. A machine of this type takes as input a state of limited purity (for instance, a system comprising of the totally mixed state and a small number of single qubit pure states), and is able to coherently apply quantum gates. The model was considered in order to reflect the state of a quantum computer with noisy storage. In [85], Kapourniotis, Kashefi and Datta introduced a verification protocol for this model by adapting VUBQC to the one-pure qubit setting. The verifier still prepares individual pure qubits, as in the original VUBQC protocol, however the prover holds a mixed state of limited purity at all times.[46] Additionally, the prover can inject or remove pure qubits from his state, during the computation, as long as it does not increase the total purity of the state. The resulting protocol has an inverse polynomial completeness-soundness gap. However, unlike the universal protocols we have reviewed, the constraints on the prover's state do not allow for the protocol to be repeated. This means that the completeness-soundness gap cannot be boosted through repetition.

Another model, for which verification protocols have been proposed, is that of instantaneous quantum computations, or IQP [2, 86]. An IQP machine is one which can only perform unitary operations that are diagonal in the X basis and therefore commute with each other. The name "instantaneous quantum computation" illustrates that there is no temporal structure to the quantum dynamics [2]. Additionally, the machine is restricted to measurements in the computational basis. It is important to mention that IQP does not represent a decision class, like BQP, but rather a

[46]The purity of a d-qubit state, ρ, is quantified by the *purity parameter* defined in [85] as: $\pi(\rho) = log(Tr(\rho^2)) + d$.

sampling class. The input to a sampling problem is a specification of a certain probability distribution and the output is a sample from that distribution. The class IQP, therefore, contains all distributions which can be sampled efficiently (in polynomial time) by a machine operating as described above. Under plausible complexity theoretic assumptions, it was shown that this class is not contained in the set of distributions which can be efficiently sampled by a classical computer [86].

In [2], Shepherd and Bremner proposed a hypothesis test in which a classical verifier is able to check that the prover is sampling from an IQP distribution. The verifier cannot, however, check that the prover sampled from the correct distributions. Nevertheless, the protocol serves as a practical tool for demonstrating a quantum computational advantage. The test itself involves an encoding, or obfuscation scheme which relies on a computational assumption (i.e. it assumes that a particular problem is intractable for IQP machines).

Another test of IQP problems is provided by the Hangleiter et al approach, from Section 3.2 [30]. Recall that this was essentially the 1S-Post-hoc protocol for certifying the ground state of a local Hamiltonian. Hangleiter et al have the prover prepare multiple copies of a state which is the Feynman-Kitaev state of an IQP circuit. They then use the post hoc protocol to certify that the prover prepared the correct state (measuring local terms from the Hamiltonian associated with that state) and then use one copy to sample from the output of the IQP circuit. This is akin to the measurement-only approach of Section 3.1. In a subsequent paper, by Bermejo-Vega et al, they consider a subclass of sampling problems that are contained in IQP and prove that this class is also hard to classically simulate (subject to standard complexity theory assumptions). The problems can be viewed as preparing a certain entangled state and then measuring all qubits in a fixed basis. The authors provide a way to certify that the state prepared is close to the ideal one, by giving an upper bound on the trace distance. Moreover, the measurements required for this state certification can be made using local stabilizer measurements, for the considered architectures and settings [5].

Recently, another scheme has been proposed, by Mills et al. [87], which again adapts the VUBQC protocol to the IQP setting. This eliminates the need for computational assumptions, however it also requires the verifier to have a single qubit preparation device. In contrast to VUBQC, however, the verifier need only prepare eigenstates of the Y and Z operators.

Yet another scheme derived from VUBQC was introduced in [88] for a model known as the *Ising spin sampler*. This is based on the *Ising model*, which describes a lattice of interacting spins in the presence of a magnetic field [89]. The Ising spin sampler is a translation invariant Ising model in which one measures the spins thus obtaining samples from the partition function of the model. Just like with IQP, it was shown in [90] that, based on complexity theoretic assumptions, sampling from the partition function is intractable for classical computers.

Lastly, Disilvestro and Markham proposed a verification protocol [91] for *Spekkens' toy model* [92]. This is a local hidden variable theory which is phenomenologically very similar to quantum mechanics, though it cannot produce non-local correlations. The existence of the protocol, again inspired by VUBQC, suggests that

Bell non-locality is not a necessary feature for verification protocols, at least in the setting in which the verifier has a trusted quantum device.

5.2 Fault Tolerance

The protocols reviewed in this paper have all been described in an ideal setting in which all quantum devices work perfectly and any deviation from the ideal behaviour is the result of malicious provers. This is not, however, the case in the real world. The primary obstacle, in the development of scalable quantum computers, is noise which affects quantum operations and quantum storage devices. As a solution to this problem, a number of fault tolerant techniques, utilizing quantum error detection and correction, have been proposed. Their purpose is to reduce the likelihood of the quantum computation being corrupted by imperfect gate operations. But while these techniques have proven successful in minimizing errors in quantum computations, it is not trivial to achieve the same effect for verification protocols. To clarify, while we have seen the use of quantum error correcting codes in verification protocols, their purpose was to either boost the completeness-soundness gap (in the case of prepare-and-send protocols), or to ensure an honest behaviour from the provers (in the case of entanglement-based post hoc protocols). The question we ask, therefore, is: how can one design a fault-tolerant verification protocol? Note that this question pertains primarily to protocols in which the verifier is not entirely classical (such as the prepare-and-send or receive-and-measure approaches) or in which one or more provers are assumed to be single-qubit devices (such as the GKW and HPDF protocols). For the remaining entanglement-based protocols, one can simply assume that the provers are performing all of their operations on top of a quantum error correcting code.

Let us consider what happens if, in the prepare-and-send and receive-and-measure protocols, the devices of the verifier and the prover are subject to noise.[47] If, for simplicity, we assume that the errors on these devices imply that each qubit will have a probability, p, of producing the same outcome as in the ideal setting, when measured, we immediately notice that the probability of n qubits producing the same outcomes scales as $O(p^n)$. This means that, even if the prover behaves honestly, the computation is very unlikely to result in the correct outcome [19].

Ideally, one would like the prover to perform his operations in a fault tolerant manner. In other words, the prover's state should be encoded in a quantum error correcting code, the gates he performs should result in logical operations being applied on his state and he should, additionally, perform error-detection (syndrome) measurements and corrections. But we can see that this is problematic to achieve. Firstly, in prepare-and-send protocols, the computation state of the prover is provided by the verifier. Who should then encode this state in the error-correcting code, the verifier or

[47] Different noise models have been examined when designing fault tolerant protocols, however, a very common model and one which can be considered in our case, is *depolarizing noise* [93, 94]. This can be single-qubit depolarizing noise, which acts as $\mathcal{E}(\rho) = (1-p)[I] + p/3([\mathsf{X}] + [\mathsf{Y}] + [\mathsf{Z}])$, or two-qubit depolarizing noise, which acts as $\mathcal{E}(\rho) = (1-p)[I \otimes I] + p/15([I \otimes \mathsf{X}] + ... [\mathsf{Z} \otimes \mathsf{Z}])$, for some probability $p > 0$. The square bracket notation indicates the action of an operator.

the prover? It is known that in order to suppress errors in a quantum circuit, $\mathcal{C}$, each qubit should be encoded in a logical state having $O(polylog(|\mathcal{C}|))$-many qubits [93]. This means that if the encoding is performed by the verifier, she must have a quantum computer whose size scales poly-logarithmically with the size of the circuit that she would like to delegate. It is preferable, however, that the verifier has a constant-size quantum computer. Conversely, even if the prover performs the encoding, there is another complication. Since the verifier needs to encrypt the states she sends to the prover, and since her operations are susceptible to noise, the errors acting on these states will have a dependency on her secret parameters. This means that when the prover performs error-detection procedures he could learn information about these secret parameters and compromise the protocol.

For receive-and-measure protocols, one encounters a different obstacle. While the verifier's measurement device is not actively malicious, if the errors occurring in this device are correlated with the prover's operations in preparing the state, this can compromise the correctness of the protocol.

A number of fault tolerant verification protocols have been proposed, however, they all overcome these limitations by making additional assumptions. For instance, one proposal, by Kapourniotis and Datta [88], for making VUBQC fault tolerant, uses a topological error-correcting code described in [58, 59]. The error-correcting code is specifically designed for performing fault tolerant MBQC computations, which is why it is suitable for the VUBQC protocol. In the proposed scheme, the verifier still prepares single qubit states, however there is an implicit assumption that the errors on these states are independent of the verifier's secret parameters. The prover is then instructed to perform a blind MBQC computation in the topological code. The protocol described in [88] is used for a specific type of MBQC computation designed to demonstrate a quantum computational advantage. However, the authors argue that the techniques are general and could be applied for universal quantum computations.

A fault-tolerant version of the measurement-only protocol from Section 3.1 has also been proposed in [95]. The graph state prepared by the prover is encoded in an error-correcting code, such as the topological lattice used by the previous approaches. As in the 'non-fault-tolerant' version of the protocol, the prover is instructed to send many copies of this state which the verifier will test using stabilizer measurements. The verifier also uses one copy in order to perform her computation in an MBQC fashion. The protocol assumes that the errors occurring on the verifier's measurement device are independent of the errors occurring on the prover's devices.

More details, regarding the difficulties with achieving fault tolerance in QPIP protocols, can be found in [26].

5.3 Experiments and Implementations

Protocols for verification will clearly be useful for benchmarking experiments implementing quantum computations. Experiments implementing quantum computations on a small number of qubits can be verified with brute force simulation on a classical computer. However, as we have pointed out that this is not scalable, in the long-term it is worthwhile to try and implement verification protocols on these devices. As a

result, there have been proof of concept experiments that demonstrate the components necessary for verifiable quantum computing.

Inspired by the prepare-and-send VUBQC protocol, Barz et al implemented a four-photon linear optical experiment, where the four-qubit linear cluster state was constructed from entangled pairs of photons produced through parametric down-conversion [96]. Within this cluster state, in runs of the experiment, a trap qubit was placed in one of two possible locations, thus demonstrating some of the elements of the VUBQC protocol. However, it should be noted that the trap qubits are placed in the system through measurements on non-trap qubits within the cluster state, i.e. through measurements made on the the other three qubits. Because of this, the analysis of the VUBQC protocol cannot be directly translated over to this setting, and bespoke analysis of possible deviations is required. In addition, the presence of entanglement between the photons was demonstrated through Bell tests that are performed blindly. This work also builds on a previous experimental implementation of blind quantum computation by Barz et al. [97].

With regards to receive-and-measure protocols, and in particular the measurement-only protocol of Section 3.1, Greganti et al. implemented [98] some of the elements of these protocols with a four-photon experiment, similar to the experiment of Barz et al mentioned above [96]. This demonstration builds on previous work in the experimental characterisation of stabiliser states [99]. In this case, two four-qubit cluster states were generated: the linear cluster state and the star graph state, where in the latter case the only entanglement is between one central qubit and pairwise with every other qubit. In order to demonstrate the elements for measurement-only verification, by suitable measurements made by the client, traps can be placed in the state. Furthermore, the linear cluster state and star graph state can be used as computational resources for implementing single qubit unitaries and an entangling gate respectively.

Finally, preliminary steps have been taken towards an experimental implementation of the RUV protocol, from Section 4.1. Huang et al implemented a simplified version of this protocol using sources of pairs of entangled photons [74]. Repeated CHSH tests were performed on thousands of pairs of photons demonstrating a large violation of the CHSH inequality; a vital ingredient in the protocol of RUV. In between the many rounds of CHSH tests, state tomography, process tomography, and a computation were performed, with the latter being the factorisation of the number 15. Again, all of these elements are ingredients in the protocol, however, the entangled photons are created 'on-the-fly'. In other words, in RUV, two non-communicating provers share a large number of maximally entangled states prior to the full protocol, but in this experiment these states are generated throughout.

6 Conclusions

The realization of the first quantum computers capable of outperforming classical computers at non-trivial tasks is fast approaching. All signs indicate that their development will follow a similar trajectory to that of classical computers. In other words, the first generation of quantum computers will comprise of large servers that are maintained and operated by specialists working either in academia, industry or a

combination of both. However, unlike with the first super-computers, the Internet opens up the possibility for users, all around the world, to interface with these devices and delegate problems to them. This has already been the case with the 5-qubit IBM machine [100], and more powerful machines are soon to follow [101, 102]. But how will these computationally restricted users be able to verify the results produced by the quantum servers? That is what the field of quantum verification aims to answer. Moreover, as mentioned before and as is outlined in [12], the field also aims to answer the more foundational question of: how do we verify the predictions of quantum mechanics in the large complexity regime?

In this paper, we have reviewed a number of protocols that address these questions. While none of them achieve the ultimate goal of the field, which is to have a classical client verify the computation performed by a single quantum server, each protocol provides a unique approach for performing verification and has its own advantages and disadvantages. We have seen that these protocols combine elements from a multitude of areas including: cryptography, complexity theory, error correction and the theory of quantum correlations. We have also seen that proof-of-concept experiments, for some of these protocols, have already been realized.

What all of the surveyed approaches have in common, is that none of them are based on computational assumptions. In other words, they all perform verification unconditionally. However, recently, there have been attempts to reduce the verifier's requirements by incorporating computational assumptions as well. What this means is that the protocols operate under the assumption that certain problems are intractable for quantum computers. We have already mentioned an example: a protocol for verifying the sub-universal sampling class of IQP computations, in which the verifier is entirely classical. Other examples include protocols for *quantum fully homomorphic encryption* [103, 104]. In these protocols, a client is delegating a quantum computation to a server while trying to keep the input to the computation hidden. The use of computational assumptions allows these protocols to achieve this functionality using only one round of back-and-forth communication. However, in the referenced schemes, the client does require some minimal quantum capabilities. A recent modification of these schemes has been proposed in order to make the protocols verifiable as well [105]. Additionally, an even more recent paper introduces a protocol for quantum fully homomorphic encryption with an entirely classical client (again, based on computational assumptions) [106]. We can therefore see a new direction emerging in the field of delegated quantum computations. This recent success in developing protocols based on computational assumptions could very well lead to the first single-prover verification protocol with a classical client.

Another new direction, especially pertaining to entanglement-based protocols, is given by the development of self-testing results achieving constant robustness. This started with the work of Natarajan and Vidick, which was the basis of their protocol from Section 4.3 [23]. We saw, in Section 4, that all entanglement-based protocols rely, one way or another, on self-testing results. Consequently, the robustness of these results greatly impacts the communication complexity and overhead of these protocols. Since most protocols were based on results having inverse polynomial robustness, this led to prohibitively large requirements in terms of quantum resources (see Table 6). However, subsequent work by Coladangelo et al, following up on the

Natarajan and Vidick result, has led to two entanglement-based protocols, which achieve near linear overhead [24].[48] This is a direct consequence of using a self-testing result with constant robustness and combining it with the Test-or-Compute protocol of Broadbent from Section 2.3. Of course, of the two protocols proposed by Coladangelo et al, only one is blind and so an open problem, of their result, is whether the second protocol can also be made blind. Another question is whether the protocols can be further optimized so that only one prover is required to perform universal quantum computations, in the spirit of the GKW protocol from Section 4.1.

We conclude by listing a number of other open problems that have been raised by the field of quantum verification. The resolution of these problems is relevant not just to quantum verification but to quantum information theory as a whole.

- While the problem of a classical verifier delegating computations to a single prover is the main open problem of the field, we emphasize a more particular instance of this problem: can the proof that any problem in PSPACE[49] admits an interactive proof system, be adapted to show that any problem in BQP admits an interactive proof system with a BQP prover? The proof that PSPACE = IP (in particular the PSPACE $\subseteq$ IP direction) uses error-correcting properties of low-degree polynomials to give a verification protocol for a PSPACE-complete problem [107]. We have seen that the Poly-QAS VQC scheme, presented in Section 2.1, also makes use of error-correcting properties of low-degree polynomials in order to perform quantum verification (albeit, with a quantum error correcting code and a quantum verifier). Can these ideas lead to a classical verifier protocol for BQP problems with a BQP prover?
- In all existing entanglement-based protocols, one assumes that the provers are not allowed to communicate during the protocol. However, this assumption is not enforced by physical constraints. Is it, therefore, possible to have an entanglement-based verification protocol in which the provers are *space-like separated*?[50] Note, that since all existing protocols require the verifier to query the two (or more) provers adaptively, it is not directly possible to make the provers be space-like separated.
- What is the optimal overhead (in terms of either communication complexity, or the resources of the verifier) in verification protocols? For all types of verification protocols we have seen that, for a fixed completeness-soundness gap, the best achieved communication complexity is linear. For the prepare-and-send case is it possible to have a protocol in which the verifier need only prepare a poly-logarithmic number of single qubits (in the size of the computation)? For the entanglement-based case, can the classical verifier send only poly-logarithmic sized questions to the provers? This latter question is related to the quantum PCP conjecture [108].

[48] The result from [24] appeared on the arxiv close to the completion of this work, which is why we did not review it.

[49] PSPACE is the class of problems which can be solved in polynomial space by a classical computer.

[50] In an experiment, two regions are space-like separated if the time it takes light to travel from one region to the other is longer than the duration of the experiment. Essentially, according to relativity, this means that there is no causal ordering between events occurring in one region and events occurring in the other.

- Are there other models of quantum computation that are suitable for developing verification protocols? We have seen that the way in which we view quantum computations has a large impact on how we design verification protocols and what characteristics those protocols will have. Specifically, the separation between classical control and quantum resources in MBQC lead to VUBQC, or the QMA-completeness of the local Hamiltonian problem lead to the post hoc approaches. Of course, all universal models are equivalent in terms of the computations which can be performed, however each model provides a particular insight into quantum computation which can prove useful when devising new protocols. Can other models of quantum computation, such as the adiabatic model, the anyon model etc, provide new insights?
- We have seen that while certain verification protocols employ error-correcting codes, these are primarily used for boosting the completeness-soundness gap. Alternatively, for the protocols that do in fact incorporate fault tolerance, in order to cope with noisy operations, there are additional assumptions such as the noise in the verifier's device being uncorrelated with the noise in the prover's devices. Therefore, the question is: can one have a fault tolerant verification protocol, with a minimal quantum verifier, in the most general setting possible? By this we mean that there are no restrictions on the noise affecting the quantum devices in the protocol, other than those resulting from the standard assumptions of fault tolerant quantum computations (constant noise rate, local errors etc). This question is addressed in more detail in [26]. Note that the question refers in particular to prepare-and-send and receive-and-measure protocols, since entanglement-based approaches are implicitly fault tolerant (one can assume that the provers are performing the computations on top of error correcting codes).

Acknowledgements The authors would like to thank Petros Wallden, Alex Cojocaru, Thomas Vidick for very useful comments and suggestions for improving this work, and Dan Mills for TEX support. AG would also like to especially thank Matty Hoban for many helpful remarks and comments and Vivian Uhlir for useful advice in improving the figures in the paper. EK acknowledges funding through EPSRC grant EP/N003829/1 and EP/M013243/1. TK acknowledges funding through EPSRC grant EP/K04057X/2.

Appendix

Quantum Information and Computation

In this section, we provide a few notions regarding the basics of quantum information and quantum computation and refer the reader to the appropriate references for a more in depth presentation [93, 109, 110].

Basics of Quantum Mechanics A quantum state (or a quantum register) is a unit vector in a complex Hilbert space, $\mathcal{H}$. We denote quantum states, using standard

Dirac notation, as $|\psi\rangle \in \mathcal{H}$, called a 'ket' state. The dual of this state is denoted $\langle\psi|$, called a 'bra', and is a member of the dual space $\mathcal{H}^{\perp}$. We will only be concerned with finite-dimensional Hilbert spaces. Qubits are states in two-dimensional Hilbert spaces. Traditionally, one fixes an orthonormal basis for such a space, called *computational basis*, and denotes the basis vectors as $|0\rangle$ and $|1\rangle$. Gluing together systems to express the states of multiple qubits is achieved through *tensor product*, denoted $\otimes$. The notation $|\psi\rangle^{\otimes n}$ denotes a state comprising of n copies of $|\psi\rangle$. If a state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ cannot be expressed as $|a\rangle \otimes |b\rangle$, for any $|a\rangle \in \mathcal{H}_1$ and any $|b\rangle \in \mathcal{H}_2$, we say that the state is *entangled*. As a shorthand, we will sometimes write $|a\rangle\,|b\rangle$ instead of $|a\rangle \otimes |b\rangle$. As a simple example of an entangled state one can consider the *Bell state*:

$$|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{77}$$

Quantum mechanics postulates that there are two ways to change a quantum state: *unitary evolution* and *measurement*. Unitary evolution involves acting with some unitary operation U on $|\psi\rangle$, thus producing the mapping $|\psi\rangle \rightarrow U\,|\psi\rangle$. Note that any such operation is reversible through the application of the hermitian conjugate of U, denoted $U^{\dagger}$, since $UU^{\dagger} = U^{\dagger}U = I$.

Measurement, in its most basic form, involves expressing a state $|\psi\rangle$ in a particular orthonormal basis, $\mathcal{B}$, and then choosing one of the basis vectors as the state of the system post-measurement. The index of that vector is the classical outcome of the measurement. The post-measurement vector is chosen at random and the probability of obtaining a vector $|v\rangle \in \mathcal{B}$ is given by $|\,\langle v|\psi\rangle\,|^2$.

More generally, a measurement involves a collection of operators $\{M_i\}_i$ acting on the state space of the system to be measured and satisfying:

$$\sum_i M_i^{\dagger} M_i = I \tag{78}$$

The label i indicates a potential measurement outcome. Given a state $|\psi\rangle$ to be measured, the probability of obtaining outcome i is $p(i) = \langle\psi|M_i^{\dagger}M_i|\psi\rangle$ and the state of the system after the measurement will be $M_i|\psi\rangle/\sqrt{p(i)}$. If we are only interested in the probabilities of the different outcomes and not in the post-measurement state then we can denote $E_i = M_i^{\dagger}M_i$ and we will refer to the set $\{E_i\}_i$ as a *positive-operator valued measure* (POVM). When performing a measurement in an orthonormal basis $\mathcal{B} = \{|i\rangle\}_i$, we are essentially choosing $M_i = |i\rangle\langle i|$. This is known as a *projective measurement* and in general consists of operators M_i satisfying the property that $M_i^2 = M_i$.

Lastly, when discussing measurements we will sometimes use *observables*. These are hermitian operators which define a measurement specified by the diagonal basis of the operator. Specifically, for some hermitian operator O, we know that there exists an orthonormal basis $\mathcal{B} = \{|i\rangle\}_i$ such that:

$$O = \sum_i \lambda_i |i\rangle \tag{79}$$

where $\{\lambda_i\}_i$ is the set of eigenvalues of O. Measuring the O observable on some state $|\psi\rangle$ is equivalent to performing a projective measurement of $|\psi\rangle$ in the basis $\mathcal{B}$.[51] When using observables, one takes the measurement outcomes to be the eigenvalues of O, rather than the basis labels. In other words, if when measuring O the state is projected to $|i\rangle$, then the measurement outcome is taken to be λ_i.

Density Matrices States denoted by kets are also referred to as *pure states*. Quantum mechanics tells us that for an isolated quantum system the complete description of that system is given by a pure state.[52] This is akin to classical physics where pure states are points in phase space, which provide a complete characterisation of a classical system. However, unlike classical physics where knowing the pure state uniquely determines the outcomes of all possible measurements of the system, in quantum mechanics measurements are probabilistic even given the pure state. It is also possible that the state of a quantum system is specified by a probability distribution over pure states. This is known as a *mixed state* and can be represented using *density matrices*. These are positive semidefinite, trace one, hermitian operators.

The density matrix of a pure state $|\psi\rangle$ is $\rho = |\psi\rangle\langle\psi|$. For an ensemble of states $\{|\psi_i\rangle\}_i$, each occurring with probability p_i, such that $\sum_i p_i = 1$, the corresponding density matrix is:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{80}$$

It can be shown that if ρ corresponds to a pure state then $Tr(\rho^2) = 1$, whereas when ρ is a mixed state $Tr(\rho^2) < 1$. One of the most important mixed states, which we encounter throughout this review, is the *maximally mixed state*. The density matrix for this state is I/d, where I is the identity matrix and d is the dimension of the underlying Hilbert space. As an example, the maximally mixed state for a one qubit system is $I/2$. This state represents the state of maximal uncertainty about quantum system. What this means is that for any basis $\{|v_i\rangle\}_i$ of the Hilbert space of dimension d, the maximally mixed state is:

$$\frac{I}{d} = \frac{1}{d}\sum_{i=1}^{d} |v_i\rangle\langle v_i| \tag{81}$$

Equivalently, any non-degenerate projective measurement, specified by an orthonormal basis $\mathcal{B}$, of the maximally mixed state will have all outcomes occurring with equal probability. We will denote the set of all density matrices over some Hilbert space $\mathcal{H}$ as $\mathcal{D}(\mathcal{H})$.

When performing a measurement on a state ρ, specified by operators $\{M_i\}_i$, the probability of outcome i is given by $p(i) = Tr(M_i^\dagger M_i \rho)$ and the post-measurement state will be $M_i \rho M_i^\dagger / p(i)$.

[51] Note that if the operator is degenerate (i.e. has repeating eigenvalues) then the projectors for degenerate eigenvalues will correspond to projectors on the subspaces spanned by the associated eigenvectors.

[52] It should be noted that this is the case provided that quantum mechanics is a *complete* theory in terms of its characterisation of physical systems. See [111] for more details.

Purification An essential operation concerning density matrices is the *partial trace*. This provides a way of obtaining the density matrix of a subsystem that is part of a larger system. Partial trace is linear, and is defined as follows. Given two density matrices ρ_1 and ρ_2 with Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, we have that:

$$\rho_1 = Tr_2(\rho_1 \otimes \rho_2) \qquad \rho_2 = Tr_1(\rho_1 \otimes \rho_2) \tag{82}$$

In the first case one is 'tracing out' system 2, whereas in the second case we trace out system 1. This property together with linearity completely defines the partial trace. For if we take any general density matrix, ρ, on $\mathcal{H}_1 \otimes \mathcal{H}_2$, expressed as:

$$\rho = \sum_{i,i',j,j'} a_{ii'jj'} \, |i\rangle_1 \langle i'|_1 \otimes |j\rangle_2 \langle j'|_2 \tag{83}$$

where $\{|i\rangle\}$ ($\{|i'\rangle\}$) and $\{|j\rangle\}$ ($\{|j'\rangle\}$) are orthonormal bases for $\mathcal{H}_1$ and $\mathcal{H}_2$, if we would like to trace out subsystem 2, for example, we would then have:

$$Tr_2(\rho) = Tr_2\left(\sum_{i,i',j,j'} a_{ii'jj'} \, |i\rangle_1 \langle i'|_1 \otimes |j\rangle_2 \langle j'|_2\right) = \sum_{i,i',j} a_{ii'jj} \, |i\rangle_1 \langle i'|_1 \tag{84}$$

An important fact, concerning the relationship between mixed states and pure states, is that any mixed state can be *purified*. In other words, for any mixed state ρ over some Hilbert space $\mathcal{H}_1$ one can always find a pure state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ such that $dim(\mathcal{H}_1) = dim(\mathcal{H}_2)$[53] and:

$$Tr_2(|\psi\rangle \langle\psi|) = \rho \tag{85}$$

Moreover, the purification $|\psi\rangle$ is not unique and so another important result is the fact that if $|\phi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ is another purification of ρ then there exists a unitary U, acting only on $\mathcal{H}_2$ (the additional system that was added to purify ρ) such that:

$$|\phi\rangle = (I \otimes U)|\psi\rangle \tag{86}$$

We will refer to this as the *purification principle*.

CPTP Maps and Isometries All operations on quantum states can be viewed as maps from density matrices on an input Hilbert space to density matrices on an output Hilbert space, $\mathcal{O} : \mathcal{D}(\mathcal{H}_{in}) \to \mathcal{D}(\mathcal{H}_{out})$, which may or may not be of the same dimension. Quantum mechanics dictates that such a map, must satisfy three properties:

1. **Linearity**: $\mathcal{O}(a\rho_1 + b\rho_2) = a\mathcal{O}(\rho_1) + b\mathcal{O}(\rho_2)$.
2. **Complete positivity**: the map $\mathcal{O} \otimes I : \mathcal{D}(\mathcal{H}_{in} \otimes \mathcal{H}_E) \to \mathcal{D}(\mathcal{H}_{out} \otimes \mathcal{H}_E))$ takes positive states to positive states, for all extensions $\mathcal{H}_E$.
3. **Trace preserving**: $Tr(\mathcal{O}(\rho)) = Tr(\rho)$.

[53] One could allow for purifications in larger systems, but we restrict attention to same dimensions.

For this reason, such maps are referred to as *completely positive trace-preserving* (CPTP) maps. It can be shown that any CPTP map can be equivalently expressed as:

$$\mathcal{O}(\rho) = \sum_i K_i \rho K_i^\dagger \tag{87}$$

for some set of linear operators $\{K_i\}_i$, known as *Kraus operators*, satisfying:

$$\sum_i K_i^\dagger K_i = I \tag{88}$$

CPTP maps are also referred to as *quantum channels*.

Let us also define *isometries*. First, let $\Phi : \mathcal{H}_{in} \to \mathcal{H}_{out}$ be a bounded linear map. The adjoint of Φ, denoted $\Phi^\dagger$ is the unique linear map $\Phi^\dagger : \mathcal{H}_{out} \to \mathcal{H}_{in}$ such that for all $|\psi\rangle \in \mathcal{H}_{in}$, $|\phi\rangle \in \mathcal{H}_{out}$:

$$\langle \Phi(\psi)|\phi\rangle = \left\langle \psi|\Phi^\dagger(\phi)\right\rangle \tag{89}$$

An isometry is a bounded linear map, $\Phi : \mathcal{H}_{in} \to \mathcal{H}_{out}$ such that:

$$\Phi^\dagger \circ \Phi = id \tag{90}$$

where id is the identity map (on $\mathcal{H}_{in}$).

Trace Distance We will frequently be interested in comparing the "closeness" of quantum states. To do so we will use the notion of *trace distance* which generalizes *variation distance* for probability distributions. Recall that if one has two probability distributions $p(x)$ and $q(x)$, over a finite sample space, the variation distance between them is defined as:

$$D(p, q) = \frac{1}{2} \sum_x |p(x) - q(x)| \tag{91}$$

Informally, this represents the largest possible difference between the probabilities that the two distributions can assign to some even x. The quantum analogue of this, for density matrices, is:

$$TD(\rho_1, \rho_2) = \frac{1}{2} Tr\left(\sqrt{(\rho_1 - \rho_2)^2}\right) \tag{92}$$

One could think that the trace distance simply represents the variation distance between the probability distributions associated with measuring ρ_1 and ρ_2 in the same basis (or using the same POVM). However, there are infinitely many choices of a measurement basis. So, in fact, the trace distance is the *maximum* over all possible measurements of the variation distance between the corresponding probability distributions.

Similar to variation distance, the trace distance takes values between 0 and 1, with 0 corresponding to identical states and 1 to perfectly distinguishable states. Additionally, like any other distance measure, it satisfies the triangle inequality.

Quantum Computation Quantum computation is most easily expressed in the *quantum gates model*. In this framework, gates are unitary operations which act on

groups of qubits. As with classical computation, universal quantum computation is achieved by considering a fixed set of quantum gates which can approximate any unitary operation up to a chosen precision. The most common universal set of gates is given by:

$$\mathsf{X}=\begin{bmatrix}0&1\\1&0\end{bmatrix}\ \mathsf{Z}=\begin{bmatrix}1&0\\0&-1\end{bmatrix}\ \mathsf{H}=\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\ \mathsf{T}=\begin{bmatrix}1&0\\0&e^{i\pi/4}\end{bmatrix}\ \mathsf{CNOT}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} \tag{93}$$

In order, the operations are known as Pauli X and Pauli Z, Hadamard, the T-gate and controlled-NOT. Note that general controlled-U operations are operations performing the mapping $|0\rangle|\psi\rangle \to |0\rangle|\psi\rangle$, $|1\rangle|\psi\rangle \to |1\rangle U|\psi\rangle$. The first qubit is known as a *control qubit*, whereas the second is known as *target qubit*. The matrices express the action of each operator on the computational basis. A classical outcome for a particular quantum computation can be obtained by measuring the quantum state resulting from the application of a sequence of quantum gates. Another gate, which we will encounter, is the *Toffoli* gate, or the controlled-controlled-NOT gate, described by the matrix:

$$\mathsf{CCNOT}=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix} \tag{94}$$

The effect of this gate is to apply an X on a target qubit if both control qubits are in the $|1\rangle$ state.

We also mention an important class of quantum operations known as *Clifford operations*. To define them, consider first the n-qubit Pauli group:

$$\mathbb{P}_n = \{\alpha\ \sigma_1 \otimes ... \otimes \sigma_n | \alpha \in \{+1, -1, +i, -i\}, \sigma_i \in \{I, \mathsf{X}, \mathsf{Y}, \mathsf{Z}\}\} \tag{95}$$

As a useful side note, the n-qubit Pauli group forms a basis for all $2^n \times 2^n$ matrices. The *Clifford group* is then defined as follows:

$$\mathfrak{C}_n = \{U \in U(2^n) | \sigma \in \mathbb{P}_n \implies U\sigma U^\dagger \in \mathbb{P}_n\} \tag{96}$$

Where $U(2^n)$ is the set of all $2^n \times 2^n$ unitary matrices. Clifford operations, therefore, are operations which leave the Pauli group invariant under conjugation (in other words, they normalise the Pauli group). Operationally they can be obtained through combinations of the Pauli gates together with H, CNOT and $\mathsf{S} = \mathsf{T}^2$, in which case they are referred to as *Clifford circuits*. We note that the T and Toffoli gates are not Clifford operations. However, Clifford circuits combined with either of these two gates gives a universal set of quantum operations.

Bloch Sphere The final aspect we mention is the *Bloch sphere*, which offers a useful geometric picture for visualizing single qubit states. Any such state is represented

as a point on the surface of the sphere. In Fig. 16, one can see a visualization of the Bloch sphere together with the states $|0\rangle, |1\rangle$, the eigenstates of Z, as well as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, the eigenstates of X and $\left|+_{\pi/2}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle), \left|-_{\pi/2}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)$, the eigenstates of Y. All of the previously mentioned single-qubit operations can be viewed as rotations on this sphere. The Pauli $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ gates correspond to rotations by π radians around the corresponding $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ axes. The Hadamard gate, which can be expressed as $\mathsf{H} = \frac{1}{\sqrt{2}}(\mathsf{X} + \mathsf{Z})$ acts as a rotation by π radians around the $\mathsf{X} + \mathsf{Z}$ axis. Lastly, the T gate, corresponds to a rotation by $\pi/4$ radians around the Z axis.

We will frequently mention the states $\left|+_\phi\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\phi}\,|1\rangle)$ and $\left|-_\phi\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\phi}\,|1\rangle)$ which all lie in the XY-plane of the Bloch sphere, represented in blue in the above figure. These states can be viewed as rotations of the $|+\rangle, |-\rangle$ states by ϕ radians around the Z axis. For example, the $\left|+_{\pi/2}\right\rangle, \left|-_{\pi/2}\right\rangle$ states are rotations by $\pi/2$ around the Z axis of the $|+\rangle, |-\rangle$ states. One can also consider measurements in the XY-plane. Any two diametrically opposed states in this plane form a basis for a one-qubit Hilbert space and therefore define a projective measurement. Suppose we choose the basis $(\left|+_\phi\right\rangle, \left|-_\phi\right\rangle)$ and wish to measure the state $|+_\theta\rangle$. It can be shown that the probability of the state being projected to $\left|+_\phi\right\rangle$ is $cos^2((\phi - \theta)/2)$, whereas the probability of it being projected to $\left|-_\phi\right\rangle$ is $sin^2((\phi - \theta)/2)$. In other words, the probabilities only depend on the *angle difference* between ϕ and θ. This fact will prove very useful later on.

Quantum Error Correction One important consideration, when discussing quantum protocols, is that any implementation of quantum operations will be subject to noise stemming from interactions with the external environment. For this reason, one needs a *fault tolerant* way of performing quantum computation. This is achieved using protocols for quantum error detection and correction, for which we give a simplified description.

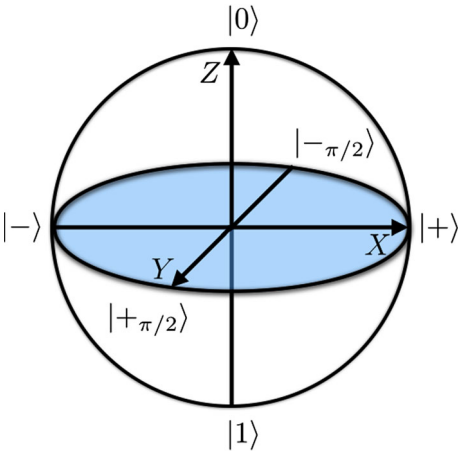


Fig. 16 Bloch sphere

Suppose we have a k-qubit quantum state $|\psi\rangle$ on which we want to perform some quantum gate G. The quantum memory storing $|\psi\rangle$ as well as the implementation of G are subject to noise. This means that if we were to apply G directly on $|\psi\rangle$ the result would be $\mathcal{E}(G|\psi\rangle)$, where $\mathcal{E}$ is a CPTP error map associated with the noisy application of G. Using the Kraus decomposition, the action of $\mathcal{E}$ can be expressed as:

$$\mathcal{E}(G|\psi\rangle) = \sum_j E_j \, G|\psi\rangle\langle\psi|G^\dagger \, E_j^\dagger \tag{97}$$

where $\{E_j\}_j$ is a set of Kraus operators. If one can correct for all E_j's then one can correct for $\mathcal{E}$ as well [112].

To detect and correct for errors from the set $\{E_j\}_j$, one first performs an encoding procedure on $|\psi\rangle$ mapping it to a so-called *logical state* $|\psi\rangle_L$ on n qubits, where $n > k$. This procedure involves the use of $n-k$ auxiliary qubits known as *ancilla* qubits. If we denote the state of these $n-k$ ancillas as $|anc\rangle$ we then have the encoding procedure $Enc(|\psi\rangle|anc\rangle) \rightarrow |\psi\rangle_L$. This state is part of a 2^k-dimensional subspace of the 2^n-dimensional Hilbert space of all n qubits, denoted $\mathcal{H}$. The subspace is usually referred to as the *code space* of the error correcting code. One way to represent this space is by giving a set of operators such that the code space is the intersection of the $+1$ eigenspaces of all the operators.

As an example, consider the 3-qubit *flip code*. We will take $k = 1$ and $n = 3$, so that one qubit is encoded in 3 qubits. The code is able to detect and correct for Pauli X errors occurring on *a single* qubit. The encoding procedure for a state $|\psi\rangle = a|0\rangle + b|1\rangle$ maps it to the state $|\psi\rangle_L = a|000\rangle + b|111\rangle$. The code space is therefore defined by $span(|000\rangle, |111\rangle)$. It is also the unique $+1$ eigenspace of the operators $g_1 = Z \otimes Z \otimes I$ and $g_2 = I \otimes Z \otimes Z$.[54] All valid operations on $|\psi\rangle_L$ must be invariant on this subspace, whereas any error from the set $\{E_j\}_j$ should map the state to a different subspace. In this case, valid operations, or *logical operations*, are the analogues of the single-qubit unitaries that map $|\psi\rangle \rightarrow |\phi\rangle = U|\psi\rangle$. Thus, a logical operation U_L would map $|\psi\rangle_L \rightarrow |\phi\rangle_L$. The error set simply consists of $\{\mathsf{X} \otimes I \otimes I, I \otimes \mathsf{X} \otimes I, I \otimes I \otimes \mathsf{X}\}$. We can see that any of these errors will map a state inside $span(|000\rangle, |111\rangle)$ to a state outside of this code space. One then defines a projective measurement in which the projectors are associated with each of the 2^{n-k} subspaces of $\mathcal{H}$. This is called a *syndrome measurement*. Its purpose is to detect whether an error has occurred and, if so, which error. Knowing this, the effect of the error can be undone by simply applying the inverse operation. For the 3-qubit code, there are $2^{3-1} = 4$ possible subspaces in which the state can be mapped to, meaning that we need a 4-outcome measurement. The syndrome measurement is defined by jointly measuring the observables g_1 and g_2. An outcome of $+1$ for both observables indicates that the state is in the correct subspace, $span(|000\rangle, |111\rangle)$. Conversely, if either of the two observables produces a -1 outcome, then this corresponds to one of the 3 possible errors. For instance, an outcome of $+1$ for the first observable

[54] These are known as *stabilizer* operators for the states in the code spaces. We also encounter these operators in Section 1. The operators form a group under multiplication and so, when specifying the code space, it is sufficient to provide the generators of the group.

and -1 for the second, indicates that the state is in the subspace $span(|001\rangle, |110\rangle)$, corresponding to an X error on the third qubit. The error is corrected by applying another X operation on that qubit.

Since Kraus operators can be expressed in terms of Pauli matrices acting on the individual qubits, one often speaks about the *weight* of an error correcting code. If the code can correct non-identity Pauli operations on at most w qubits, then w is the weight of the code.

The smallest error correcting code which can correct for *any* single-qubit error is the *5-qubit code* (i.e. one qubit is encoded as 5 qubits) [80]. This code is used Section 4.3.

Measurement-Based Quantum Computation

Since some of the protocols we review are expressed in the model of *measurement-based quantum computation* (MBQC), defined in [113–115], we provide a brief description of this model.

Unlike the quantum gates model of computation, in MBQC a given computation is performed by measuring qubits from a large entangled state. Traditionally, this state consists of qubits prepared in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, entangled using the CZ (controlled-Z) operation, where:

$$\mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

They are then measured in the basis $(|+_\phi\rangle, |-_\phi\rangle)$. These measurements are denoted as $M(\phi)$, and depending on the value of ϕ chosen for each qubit one can perform universal quantum computation. For this to work, the entangled qubits need to form a *universal graph state*. A graph state, denoted $|G\rangle$, is one in which the qubits have been entangled according to the structure of a graph G. Given some fixed constant k, a universal graph state is a family of graph states, denoted $\{|G_N\rangle\}_N$, with $N > 0$, and having kN qubits, such that, for any quantum circuit $\mathcal{C}$, consisting of N gates, there exists a measurement pattern[55] on $|G_N\rangle$ that implements $\mathcal{C}|00..0\rangle$. In other words, for each quantum circuit of size N, there is an MBQC computation using $|G_N\rangle$ that performs that circuit.

An example of such a state is the *brickwork state*, defined in [36] from which we illustrate Fig. 17. To be more precise, suppose we would like to perform some quantum computation described by a circuit consisting of N gates. The corresponding MBQC computation consists of the following steps:

1. **Initialization**. Prepare $O(N)$ qubits, each in the state $|+\rangle$.
2. **Entanglement**. Entangle the qubits according to some universal graph state structure, such as the brickwork state.

[55] A measurement pattern is simply a tuple consisting of the measurement angles, for the qubits in $|G_N\rangle$, and the partial ordering of these measurements.

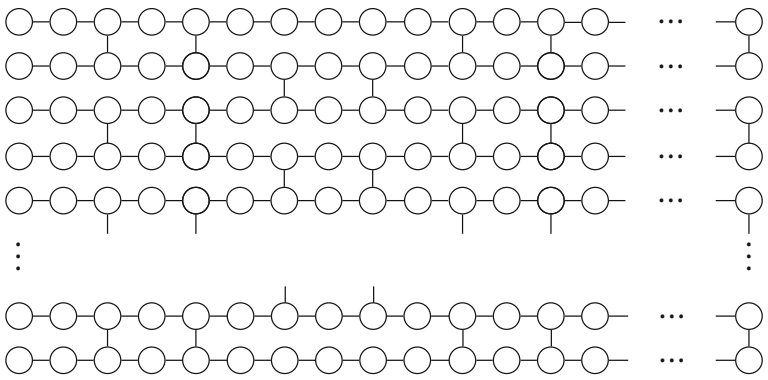


Fig. 17 Brickwork state, reproduced from [36]

3. **Measurement**. Measure each qubit, i using $M(\phi_i)$, for some angle ϕ_i determined based on the computation we would like to perform. The angles ϕ_i are referred to as the *computation angles*.
4. **Correction**. Apply appropriate corrections (Pauli X and Z operations) to the qubits, based on the measurement outcomes.

The last two steps can be performed together. This is because if we would like to apply a Pauli X correction to a qubit, i, before measuring it, we can simply measure it using $M(-\phi_i)$. Similarly, if we would like to apply a Pauli Z correction to that same qubit we measure it using $M(\phi_i + \pi)$. Therefore, the general measurement performed on a particular qubit will be $M((-1)^s\phi_i + r\pi)$, where $s, r \in \{0, 1\}$ are determined by previous measurement outcomes.

One element concerning graph states, which we will encounter in some protocols, is the representation of these states using *stabilizers*. A stabilizer state for a unitary hermitian operator, O, is some state $|\psi\rangle$ such that $O|\psi\rangle = |\psi\rangle$. O is referred to as a stabilizer of $|\psi\rangle$. It is possible to specify a state, $|\psi\rangle$, by giving a set of operators, such that $|\psi\rangle$ is the unique state (up to global phase) which is stabilized by all the operators in the set. As an example, the state $|\Phi_+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is uniquely stabilized by the set $\{\mathsf{X}\otimes\mathsf{X}, \mathsf{Z}\otimes\mathsf{Z}\}$. Note that the set of all stabilizers for a state forms a group, since if $O_1|\psi\rangle = |\psi\rangle$ and $O_2|\psi\rangle = |\psi\rangle$, then clearly $O_1O_2|\psi\rangle = |\psi\rangle$. So, it is sufficient to specify a set of generators for that group in order to describe the stabilizer group for a particular state.

To specify the generators for the stabilizer group of a graph state $|G\rangle$, let us first denote $V(G)$ as the set of vertices in the graph G and $N_G(v)$ as the set of neighbouring vertices for some vertex v (i.e. all vertices in G that are connected to v through an edge). Additionally, for some operator O, when we write O_v we mean that O is acting on the qubit from $|G\rangle$ associated with vertex v in G. The generators for the stabilizer group of $|G\rangle$ are then given by:

$$K_v = \mathsf{X}_v \prod_{w\in N_G(v)} \mathsf{Z}_w \tag{98}$$

for all $v \in V(G)$.

As a final remark, it should be noted that one can translate quantum circuits into MBQC patterns in a canonical way. For instance, the universal gate set mentioned in the previous subsection, and hence any quantum circuit comprising of those gates, can be translated directly into MBQC. See for example [36, 115] for more details.

Complexity Theory

As mentioned in the introduction, the questions regarding verification of quantum computation can be easily expressed in the language of complexity theory. To that end, we provide definitions for the basic complexity classes used in this paper. We let $\{0,1\}^*$ denote the set of all binary strings of finite length and $\{0,1\}^n$ the set of all binary strings of length n. We use standard complexity theory notation and assume familiarity with the concepts of Turing machines and uniform circuits. For a more general introduction into the subject we refer the reader to [116, 117].

Definition 5 A language $L \subseteq \{0,1\}^*$ belongs to BPP if there exists a polynomial p and a probabilistic Turing machine M, whose running time on inputs of size n is bounded by $p(n)$, such that for any $x \in \{0,1\}^n$ the following is true:

- when $x \in L$, $M(x)$[56] accepts with probability at least c,
- when $x \notin L$, $M(x)$ accepts with probability at most s,

where $c - s \geq 1/p(n)$.

Here, and in all subsequent definitions, c is referred to as *completeness* and s is referred to as *soundness*. Traditionally, one takes $c = 2/3$ and $s = 1/3$, however, in full generality, the only requirement is that there exists an inverse polynomial gap between c and s.

Definition 6 A language $L \subseteq \{0,1\}^*$ belongs to BQP if there exists a polynomial p and a uniform quantum circuit family $\{C_n\}_n$, where each circuit has at most $p(n)$ gates, 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 c,
- when $x \notin L$, $C_n(x)$ accepts with probability at most s,

where $c - s \geq 1/p(n)$.

For the quantum circuit C_n, acceptance can be defined as having one of its output qubits outputting 1 when measured in the computational basis.

Definition 7 A language $L \subseteq \{0,1\}^*$ belongs to MA if there exists a polynomial p and a probabilistic Turing machine V, whose running time on inputs of size n is bounded by $p(n)$, such that for any $x \in \{0,1\}^n$ the following is true:

[56] The notation $M(x)$ means running the Turing machine M on input x.

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

where $c - s \geq 1/p(n)$.

For this class, V is traditionally referred to as the verifier (or Arthur), whereas w, which is the witness string, is provided by the prover (or Merlin). Essentially, the verifier and is tasked with checking a purported proof that $x \in L$, provided by the prover. There is also a quantum version of this class:

Definition 8 A language $L \subseteq \{0,1\}^*$ belongs to QMA if there exists a polynomial p and a uniform quantum circuit family $\{V_n\}_n$ taking x and a quantum state $|\psi\rangle$ as inputs, such that for any $x \in \{0,1\}^n$ the following are true:

- when $x \in L$, there exists a quantum state $|\psi\rangle \in \mathcal{H}$, such that $V_n(x, |\psi\rangle)$ accepts with probability at least c, and
- when $x \notin L$, for all quantum states $|\psi\rangle \in \mathcal{H}$, $V_n(x, |\psi\rangle)$ accepts with probability at most s,

where $dim(\mathcal{H}) \leq 2^{p(|x|)}$ and $c - s \geq 1/p(|x|)$.

For QMA we also provide the definition of a complete problem[57] since this will be referenced in some of the protocols we review. The specific problem we state was defined by Kitaev et al and is known as the *k-local Hamiltonian problem* [66]. 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. We give the definition of the k-local Hamiltonian problem from [108]:

Definition 9 (**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 $\Gamma = b - a > 1/poly(n)$. Γ is referred to as the *absolute promise gap* of the problem.
- **Output:** Is the smallest eigenvalue of $H = H_1 + H_2 + ... + H_m$ smaller than a or are all its eigenvalues larger than b?

Essentially, 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 a and whenever $x \notin L$, all of its eigenvalues are greater than b. The witness $|\psi\rangle$,

[57] A problem, P, is complete for the complexity class QMA if $P \in$ QMA and all problems in QMA can be reduced in quantum polynomial time to P.

when $x \in L$, is the eigenstate of H corresponding to its lowest eigenvalue (or one such eigenstate if the Hamiltonian is degenerate). The uniform circuit family $\{V_n\}_n$ represents a BQP verifier, whereas the state $|\psi\rangle$ is provided by a prover. The verifier receives this witness from the prover and measures one of the local terms H_i (which is an observable) on that state. This can be done with a polynomial-size quantum circuit and 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 a. Conversely, when $x \notin L$, no matter what state the prover sends, with high probability, the verifier will measure a value above b. The constant k, in the definition, is not arbitrary. In the initial construction of Kitaev, k had to be at least 5 for the problem to be QMA-complete. Subsequent work has shown that even with $k = 2$ the problem remains QMA-complete [64].

Definition 10 A language $L \subseteq \{0, 1\}^*$ belongs to IP if there exists a polynomial p and a probabilistic Turing machine V, whose running time on inputs of size n is bounded by $p(n)$, such that for any $x \in \{0, 1\}^n$ the following is true:

- when $x \in L$, there exists a prover P which exchanges at most $p(n)$ messages (of length at most $p(n)$) with V and makes V accept with probability at least c,
- when $x \notin L$, any prover P which exchanges at most $p(n)$ messages (of length at most $p(n)$) with V, makes V accept with probability at most s,

where $c - s \geq 1/p(n)$.

While the previous are fairly standard complexity classes, we now state the definition of a more non-standard class, which first appeared in [25]:

Definition 11 A language $L \subseteq \{0, 1\}^*$ belongs to QPIP if there exists a polynomial p, a constant κ and a probabilistic Turing machine V, whose running time on inputs of size n is bounded by $p(n)$, and which is augmented with the ability to prepare and measure groups of κ qubits, such that for any $x \in \{0, 1\}^n$ the following is true:

- when $x \in L$, there exists a BQP prover P which exchanges at most $p(n)$ classical or quantum messages (of length at most $p(n)$) with V and makes V accept with probability at least c,
- when $x \notin L$, any BQP prover P which exchanges at most $p(n)$ classical or quantum messages (of length at most $p(n)$) with V, makes V accept with probability at most s,

where $c - s \geq 1/p(n)$.

Some clarifications are in order. The class QPIP differs from IP in two ways. Firstly, while computationally the verifier is still restricted to the class BPP, operationally it has the additional ability of preparing or measuring groups of κ qubits. Importantly, κ is a constant which is independent of the size of the input. This is why this extra ability does not add to the verifier's computational power, since a constant-size quantum device can be simulated in constant time by a BPP machine. Secondly,

unlike IP, in QPIP the prover is restricted to BQP computations. This constraint on the prover is more in line with Problem 1 and it also has the direct implication that QPIP $\subseteq$ BQP.

As we will see, all the protocols in Sections 2 and 3 are QPIP protocols. And since these protocols allow for the delegation of arbitrary BQP problems, it follows that QPIP = BQP.

We now proceed to the multi-prover setting and define the multi-prover generalization of IP:

Definition 12 A language $L \subseteq \{0,1\}^*$ belongs to MIP[k] if there exists a polynomial p and a probabilistic Turing machine V, whose running time on inputs of size n is bounded by $p(n)$, such that for any $x \in \{0,1\}^n$ the following is true:

- when $x \in L$, there exists a k-tuple of provers $(P_1, P_2, ...P_k)$ which are not allowed to communicate and which exchange at most $p(n)$ messages (of length at most $p(n)$) with V and make V accept with probability at least c,
- when $x \notin L$, any k-tuple of provers $(P_1, P_2, ...P_k)$ which are not allowed to communicate and which exchange at most $p(n)$ messages (of length at most $p(n)$) with V, make V accept with probability at most s,

where $c - s \geq 1/p(n)$.

Note that MIP[1] = IP and it was shown that for all $k > 2$, MIP[k] = MIP[2] [118]. The latter class is simply denoted MIP. If the provers are allowed to share entanglement then we obtain the class:

Definition 13 A language $L \subseteq \{0,1\}^*$ belongs to MIP*[k] if there exists a polynomial p and a probabilistic Turing machine V, whose running time on inputs of size n is bounded by $p(n)$, such that for any $x \in \{0,1\}^n$ the following is true:

- when $x \in L$, there exists a k-tuple of provers $(P_1, P_2, ...P_k)$ which can share arbitrarily many entangled qubits, are not allowed to communicate and which exchange at most $p(n)$ messages (of length at most $p(n)$) with V and make V accept with probability at least c,
- when $x \notin L$, any k-tuple of provers $(P_1, P_2, ...P_k)$ which can share arbitrarily many entangled qubits, are not allowed to communicate and which exchange at most $p(n)$ messages (of length at most $p(n)$) with V, make V accept with probability at most s,

where $c - s \geq 1/p(n)$.

As before it is the case that MIP*[k] = MIP*[2] and this class is denoted as MIP* [119]. It is not known whether MIP = MIP*, however, it is known that both classes contain BQP. Importantly, for MIP* protocols, if the provers are restricted to BQP computations, the resulting complexity class is equal to BQP [18]. Most of the protocols presented in Section 4 are of this type.

Note that while the protocols we review can be understood in terms of the listed complexity classes, we will often give a more fine-grained description of their functionality and resources than is provided by complexity theory. To give an example, for a QPIP protocol, from the complexity theoretic perspective, we are interested in the verifier's ability to delegate arbitrary BQP decision problems to the prover by interacting with it for a polynomial number of rounds. In practice, however, we are interested in a number of other characteristics of the protocol such as:

- whether the verifier can delegate not just decision problems, but also sampling problems (i.e. problems in which the verifier wishes to obtain a sample from a particular probability distribution and is able to certify that, with high probability, the sample came from the correct distribution),
- whether the prover can receive a particular quantum input for the computation or return a quantum output to the verifier,
- having minimal quantum communication between the verifier and the prover,
- whether the verifier can "hide" the input and output of the computation from the prover.

References

1. Aaronson, S., Arkhipov, A.: The computational complexity of linear optics. In: Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing. STOC '11, pp. 333–342. ACM, New York (2011)
2. Shepherd, D., Bremner, M.J.: 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)
3. Boixo, S., Isakov, S.V., Smelyanskiy, V.N., Babbush, R., Ding, N., Jiang, Z., Martinis, J.M., Neven, H.: Characterizing quantum supremacy in near-term devices. arXiv:1608.00263 (2016)
4. Aaronson, S., Chen, L.: Complexity-theoretic foundations of quantum supremacy experiments. arXiv:1612.05903 (2016)
5. Bermejo-Vega, J., Hangleiter, D., Schwarz, M., Raussendorf, R., Eisert, J.: Architectures for quantum simulation showing a quantum speedup (2017)
6. Tillmann, M., Dakić, B., Heilmann, R., Nolte, S., Szameit, A., Walther, P.: Experimental boson sampling. Nat. Photon. **7**(7), 540–544 (2013)
7. Spagnolo, N., Vitelli, C., Bentivegna, M., Brod, D.J., Crespi, A., Flamini, F., Giacomini, S., Milani, G., Ramponi, R., Mataloni, P., et al: Experimental validation of photonic boson sampling. Nat. Photon. **8**(8), 615–620 (2014)
8. Bentivegna, M., Spagnolo, N., Vitelli, C., Flamini, F., Viggianiello, N., Latmiral, L., Mataloni, P., Brod, D.J., Galvão, E.F., Crespi, A., et al.: Experimental scattershot boson sampling. Sci. Adv. **1**(3), e1400255 (2015)
9. Lanyon, B., Barbieri, M., Almeida, M., White, A.: Experimental quantum computing without entanglement. Phys. Rev. Lett. **101**(20), 200501 (2008)
10. Aaronson, S.: The Aaronson $25.00 prize. http://www.scottaaronson.com/blog/?p=284
11. Vazirani, U.: Workshop on the computational worldview and the sciences http://users.cms.caltech.edu/schulman/Workshops/CS-Lens-2/report-comp-worldview.pdf (2007)
12. Aharonov, D., Vazirani, U.: Is Quantum Mechanics Falsifiable? A Computational Perspective on the Foundations of Quantum Mechanics. Computability: Turing, Gödel, Church, and Beyond. MIT Press (2013)

13. Impagliazzo, R., Wigderson, A.: P= bpp if e requires exponential circuits: Derandomizing the xor lemma. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, pp. 220–229. ACM (1997)
14. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999)
15. Bernstein, E., Vazirani, U.: Quantum complexity theory. SIAM J. Comput. **26**(5), 1411–1473 (1997)
16. Watrous, J.: Succinct quantum proofs for properties of finite groups. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science. FOCS '00, pp. 537–. IEEE Computer Society, Washington, DC (2000)
17. Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by a quantum walk. In: Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, pp. 59–68. ACM (2003)
18. Reichardt, B.W., Unger, F., Vazirani, U.: Classical command of quantum systems. Nature **496**(7446), 456 (2013)
19. Gheorghiu, A., Kashefi, E., Wallden, P.: Robustness and device independence of verifiable blind quantum computing. J. Phys. **17**(8), 083040 (2015)
20. Hajdušek, M., Pérez-Delgado, C.A., Fitzsimons, J.F.: Device-independent verifiable blind quantum computation. arXiv:1502.02563 (2015)
21. McKague, M.: Interactive proofs for BQP via self-tested graph states. Theory Comput. **12**(3), 1–42 (2016)
22. Fitzsimons, J.F., Hajdušek, M.: Post hoc verification of quantum computation. arXiv:1512.04375 (2015)
23. Natarajan, A., Vidick, T.: Robust self-testing of many-qubit states. arXiv:1610.03574 (2016)
24. Coladangelo, A., Grilo, A., Jeffery, S., Vidick, T.: Verifier-on-a-leash: new schemes for verifiable delegated quantum computation, with quasilinear resources. arXiv:1708.07359 (2017)
25. Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In: Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings, pp. 453–469 (2010)
26. Aharonov, D., Ben-Or, M., Eban, E., Mahadev, U.: Interactive proofs for quantum computations. arXiv:1704.04487 (2017)
27. Fitzsimons, J.F., Kashefi, E.: Unconditionally verifiable blind quantum computation. Phys. Rev. A **96**, 012303 (2017)
28. Broadbent, A.: How to verify a quantum computation. Theory of Computing. arXiv:1509.09180 (2018)
29. Morimae, T., Fitzsimons, J.F.: Post hoc verification with a single prover. arXiv:1603.06046 (2016)
30. Hangleiter, D., Kliesch, M., Schwarz, M., Eisert, J.: Direct certification of a class of quantum simulations. Quant. Sci. Technol. **2**(1), 015004 (2017)
31. Hayashi, M., Morimae, T.: Verifiable measurement-only blind quantum computing with stabilizer testing. Phys. Rev. Lett. **115**(22), 220502 (2015)
32. Morimae, T., Takeuchi, Y., Hayashi, M.: Verification of hypergraph states. Phys. Rev. A **96**, 062321 (2017)
33. Gheorghiu, A., Wallden, P., Kashefi, E.: Rigidity of quantum steering and one-sided device-independent verifiable quantum computation. J. Phys. **19**(2), 023043 (2017)
34. Fitzsimons, J.F.: Private quantum computation: An introduction to blind quantum computing and related protocols. npj Quant. Inf. **3**(1), 23 (2017)
35. Childs, A.M.: Secure assisted quantum computation. Quant. Info. Comput. **5**(6), 456–466 (2005)
36. Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: Proceedings of the 50th Annual Symposium on Foundations of Computer Science. FOCS '09, pp. 517–526. IEEE Computer Society (2009)
37. Arrighi, P., Salvail, L.: Blind quantum computation. Int. J. Quant. Inf. **04**(05), 883–898 (2006)
38. Giovannetti, V., Maccone, L., Morimae, T., Rudolph, T.G.: Efficient universal blind quantum computation. Phys. Rev. Lett. **111**, 230501 (2013)
39. Mantri, A., Pérez-Delgado, C.A., Fitzsimons, J.F.: Optimal blind quantum computation. Phys. Rev. Lett. **111**, 230502 (2013)
40. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Sec. Comput. **4**(11), 169–180 (1978)

41. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing. STOC '09, pp. 169–178. ACM, New York (2009)
42. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science. FOCS '11, pp. 97–106. IEEE Computer Society, Washington, DC (2011)
43. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ITCS '12, pp. 309–325. ACM, New York (2012)
44. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques. EUROCRYPT'10, pp. 24–43. Springer, Berlin (2010)
45. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. CRC press (2014)
46. Danos, V., Kashefi, E.: Determinism in the one-way model. Physical Review A **74**(5), 052310 (2006)
47. Aaronson, S., Cojocaru, A., Gheorghiu, A., Kashefi, E.: On the implausibility of classical client blind quantum computing. arXiv:1704.08482 (2017)
48. Dunjko, V., Kashefi, E.: Blind quantum computing with two almost identical states. arXiv:1604.01586 (2016)
49. Dunjko, V., Fitzsimons, J.F., Portmann, C., Renner, R.: Composable security of delegated quantum computation. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 406–425. Springer (2014)
50. Kashefi, E., Wallden, P.: Garbled quantum computation. Cryptography **1**(1), 6 (2017)
51. Kapourniotis, T., Dunjko, V., Kashefi, E.: On optimising quantum communication in verifiable quantum computing. arXiv:1506.06943 (2015)
52. Barnum, H., Crėpeau, C., Gottesman, D., Smith, A.D., Tapp, A.: Authentication of quantum messages. In: 43rd Symposium on Foundations of Computer Science (FOCS 2002), 16–19 November 2002, Vancouver, BC, Canada, Proceedings, pp. 449–458 (2002)
53. Aharonov, D., Ben-Or, M.: Fault-tolerant quantum computation with constant error rate. SIAM J. Comput. **38**(4), 1207–1282 (2008)
54. Gottesman, D., Chuang, I.L.: Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. Nature **402**(6760), 390–393 (1999)
55. Broadbent, A., Gutoski, G., Stebila, D.: Quantum one-time programs. In: Advances in Cryptology–CRYPTO 2013, pp. 344–360. Springer (2013)
56. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd IEEE Symposium on Foundations of Computer Science, 2001. Proceedings, pp. 136–145. IEEE (2001)
57. Kashefi, E., Wallden, P.: Optimised resource construction for verifiable quantum computation. J. Phys. A: Math. Theor. **50**(14), 145306 (2017)
58. Raussendorf, R., Harrington, J., Goyal, K.: A fault-tolerant one-way quantum computer. Ann. Phys. **321**(9), 2242–2270 (2006)
59. Raussendorf, R., Harrington, J., Goyal, K.: Topological fault-tolerance in cluster state quantum computation. J. Phys. **9**(6), 199 (2007)
60. Fisher, K., Broadbent, A., Shalm, L., Yan, Z., Lavoie, J., Prevedel, R., Jennewein, T., Resch, K.: Quantum computing on encrypted data. Nat. Commun. **5**, 3074 (2014)
61. Fitzsimons, J.F., Hajdušek, M., Morimae, T.: Post hoc verification of quantum computation. Phys. Rev. Lett. **120**(4), 040501 (2018)
62. Crépeau, C.: Cut-and-choose protocol. In: Encyclopedia of Cryptography and Security, pp. 290–291. Springer (2011)
63. Kashefi, E., Music, L., Wallden, P.: The quantum cut-and-choose technique and quantum two-party computation. arXiv:1703.03754 (2017)
64. Kempe, J., Kitaev, A., Regev, O.: The complexity of the local hamiltonian problem. SIAM J. Comput. **35**(5), 1070–1097 (2006)
65. Morimae, T., Nagaj, D., Schuch, N.: Quantum proofs can be verified using only single-qubit measurements. Phys. Rev. A **93**(2), 022326 (2016)
66. Kitaev, A.Y., Shen, A., Vyalyi, M.N.: Classical and Quantum Computation, vol. 47. American Mathematical Society, Providence (2002)

67. Biamonte, J.D., Love, P.J.: Realizable Hamiltonians for universal adiabatic quantum computers. Phys. Rev. A **78**, 012352 (2008)
68. Bausch, J., Crosson, E.: Increasing the quantum unsat penalty of the circuit-to-Hamiltonian construction. arXiv:1609.08571 (2016)
69. Mayers, D., Yao, A.: Self testing quantum apparatus. Quant. Info. Comput. **4**(4), 273–286 (2004)
70. Coladangelo, A., Stark, J.: Separation of finite and infinite-dimensional quantum correlations, with infinite question or answer sets. arXiv:1708.06522 (2017)
71. Cirel'son, B.: Quantum generalizations of Bell's inequality. Lett. Math. Phys. **4**(2), 93–100 (1980)
72. Clauser, J.F., Horne, M.A., Shimony, A., Holt, R.A.: Proposed experiment to test local hidden-variable theories. Phys. Rev. Lett. **23**, 880–884 (1969)
73. McKague, M., Yang, T.H., Scarani, V.: Robust self-testing of the singlet. J. Phys. A: Math. Theor. **45**(45), 455304 (2012)
74. Huang, H.L., Zhao, Q., Ma, X., Liu, C., Su, Z.E., Wang, X.L., Li, L., Liu, N.L., Sanders, B.C., Lu, C.Y., et al.: Experimental blind quantum computing for a classical client. Phys. Rev. Lett. **119**(5), 050503 (2017)
75. Barrett, J., Hardy, L., Kent, A.: No signaling and quantum key distribution. Phys. Rev. Lett. **95**(1), 010503 (2005)
76. Acín, A., Brunner, N., Gisin, N., Massar, S., Pironio, S., Scarani, V.: Device-independent security of quantum cryptography against collective attacks. Phys. Rev. Lett. **98**(23), 230501 (2007)
77. Schrödinger, E.: Probability relations between separated systems. Math. Proc. Cambridge Philos. Soc. **32**(10), 446–452 (1936)
78. MHALLA, M., PERDRIX, S.: Graph states, pivot minor, and universality of (x, z)-measurements. Int. J. Unconv. Comput., 9 (2013)
79. Fitzsimons, J., Vidick, T.: A multiprover interactive proof system for the local hamiltonian problem. In: Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, pp. 103–112. ACM (2015)
80. Laflamme, R., Miquel, C., Paz, J.P., Zurek, W.H.: Perfect quantum error correcting code. Phys. Rev. Lett. **77**(1), 198 (1996)
81. Ji, Z.: Classical verification of quantum proofs. In: Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, pp. 885–898. ACM (2016)
82. Mermin, N.D.: Simple unified form for the major no-hidden-variables theorems. Phys. Rev. Lett. **65**(27), 3373 (1990)
83. Peres, A.: Incompatible results of quantum measurements. Phys. Lett. A **151**(3-4), 107–108 (1990)
84. Knill, E., Laflamme, R.: Power of one bit of quantum information. Phys. Rev. Lett. **81**(25), 5672 (1998)
85. Kapourniotis, T., Kashefi, E., Datta, A.: Verified delegated quantum computing with one pure qubit. arXiv:1403.1438 (2014)
86. Bremner, M.J., Jozsa, R., Shepherd, D.J.: Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, The Royal Society, rspa20100301 (2010)
87. Mills, D., Pappa, A., Kapourniotis, T., Kashefi, E.: Information theoretically secure hypothesis test for temporally unstructured quantum computation. arXiv:1704.01998 (2017)
88. Kapourniotis, T., Datta, A.: Nonadaptive fault-tolerant verification of quantum supremacy with noise. arXiv:1703.09568 (2017)
89. Ising, E.: Beitrag zur theorie des ferromagnetismus. Zeitschrift für Physik A Hadrons and Nuclei **31**(1), 253–258 (1925)
90. Gao, X., Wang, S.T., Duan, L.M.: Quantum supremacy for simulating a translation-invariant ising spin model. Phys. Rev. Lett. **118**(4), 040502 (2017)
91. Disilvestro, L., Markham, D.: Quantum protocols within Spekkens' toy model. Phys. Rev. A **95**(5), 052324 (2017)
92. Spekkens, R.W.: Evidence for the epistemic view of quantum states: A toy theory. Phys. Rev. A **75**(3), 032110 (2007)
93. Nielsen, M.A., Chuang, I.L. Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th edn. Cambridge University Press, New York (2011)
94. Buhrman, H., Cleve, R., Laurent, M., Linden, N., Schrijver, A., Unger, F.: New limits on fault-tolerant quantum computation. In: 47th Annual IEEE Symposium on Foundations of Computer Science, 2006. FOCS'06, pp. 411–419. IEEE (2006)

95. Fujii, K., Hayashi, M.: Verifiable fault-tolerance in measurement-based quantum computation. arXiv:1610.05216 (2016)
96. Barz, S., Fitzsimons, J.F., Kashefi, E., Walther, P.: Experimental verification of quantum computation. Nat. Phys. **9**(11), 727–731 (2013). Article
97. Barz, S., Kashefi, E., Broadbent, A., Fitzsimons, J.F., Zeilinger, A., Walther, P.: Demonstration of blind quantum computing. Science **335**(6066), 303–308 (2012)
98. Greganti, C., Roehsner, M.C., Barz, S., Morimae, T., Walther, P.: Demonstration of measurement-only blind quantum computing. J. Phys. **18**(1), 013020 (2016)
99. Greganti, C., Roehsner, M.C., Barz, S., Waegell, M., Walther, P.: Practical and efficient experimental characterization of multiqubit stabilizer states. Phys. Rev. A **91**(2), 022325 (2015)
100. Ibm quantum experience. http://research.ibm.com/ibm-q/
101. Ibm 16-qubit processor. https://developer.ibm.com/dwblog/2017/quantum-computing-16-qubit-processor/
102. Google 49-qubit chip. https://spectrum.ieee.org/computing/hardware/google-plans-to-demonstrate-the-supremacy-of-quantum-computing
103. Broadbent, A., Jeffery, S.: Quantum homomorphic encryption for circuits of low T-gate complexity. In: Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015. Proceedings. Part II, pp. 609–629 (2015)
104. Dulek, Y., Schaffner, C., Speelman, F.: Quantum Homomorphic Encryption for Polynomial-Sized Circuits, pp. 3–32. Springer, Berlin (2016)
105. Alagic, G., Dulek, Y., Schaffner, C., Speelman, F.: Quantum fully homomorphic encryption with verification. arXiv:1708.09156 (2017)
106. Mahadev, U.: Classical homomorphic encryption for quantum circuits. arXiv:1708.02130 (2017)
107. Shamir, A.: Ip= pspace. J. ACM (JACM) **39**(4), 869–877 (1992)
108. Aharonov, D., Arad, I., Vidick, T.: Guest column: The quantum pcp conjecture. ACM Sigact News **44**(2), 47–79 (2013)
109. Watrous, J.: Guest column: An introduction to quantum information and quantum circuits 1. SIGACT News **42**(2), 52–67 (2011)
110. Watrous, J.: Quantum computational complexity. In: Encyclopedia of Complexity and Systems Science, pp 7174–7201. Springer (2009)
111. Harrigan, N., Spekkens, R.W.: Einstein, incompleteness, and the epistemic view of quantum states. Found. Phys. **40**(2), 125–157 (2010)
112. Gottesman, D.: An introduction to quantum error correction and fault-tolerant quantum computation. In: Quantum Information Science and its Contributions to Mathematics, Proceedings of Symposia in Applied Mathematics, vol 68, pp 13–58 (2009)
113. Raussendorf, R., Briegel, H.J.: A one-way quantum computer. Phys. Rev. Lett. **86**, 5188–5191 (2001)
114. Briegel, H.J., Browne, D.E., Dur, W., Raussendorf, R., Van den Nest, M.: Measurement-based quantum computation. Nat. Phys., 19–26 (2009)
115. Raussendorf, R., Browne, D.E., Briegel, H.J.: Measurement-based quantum computation on cluster states. Phys. Rev. A **68**(2), 022312 (2003)
116. Complexity Zoo. https://complexityzoo.uwaterloo.ca/Complexity_Zoo
117. Arora, S., Barak, B. Computational Complexity: A Modern Approach, 1st edn. Cambridge University Press, New York (2009)
118. Ben-Or, M., Goldwasser, S., Kilian, J., Wigderson, A.: Multi-prover interactive proofs: How to remove intractability assumptions. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, pp. 113–131. ACM (1988)
119. Cleve, R., Hoyer, P., Toner, B., Watrous, J.: Consequences and limits of nonlocal strategies. In: 19th IEEE Annual Conference on Computational Complexity, 2004. Proceedings, pp. 236–249. IEEE (2004)