Robust Verification of Quantum Computation

Alexandru Gheorghiu

Doctor of Philosophy
School of Informatics
University of Edinburgh
2018

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Alexandru Gheorghiu)

Abstract

Quantum computers promise to offer a considerable speed-up in solving certain problems, compared to the best classical algorithms. In many instances, the gap between quantum and classical running times is conjectured to be exponential. While this is great news for those applications where quantum computers would provide such an advantage, it also raises a significant challenge: how can classical computers *verify* the correctness of quantum computations? In attempting to answer this question, a number of protocols have been developed in which a classical client (referred to as *verifier*) can interact with one or more quantum servers (referred to as *provers*) in order to certify the correctness of a quantum computation performed by the server(s). These protocols are of one of two types: either there are multiple non-communicating provers, sharing entanglement, and the verifier is completely classical; or, there is a single prover and the classical verifier has a device for preparing or measuring quantum states. The latter type of protocols are, arguably, more relevant to near term quantum computers, since having multiple quantum computers that share a large amount of entanglement is, from a technological standpoint, extremely challenging.

Before the realisation of practical single-prover protocols, a number of challenges need to be addressed: how robust are these protocols to noise on the verifier's device? Can the protocols be made fault-tolerant without significantly increasing the requirements of the verifier? How do we know that the verifier's device is operating correctly? Could this device be eliminated completely, thus having a protocol with a fully classical verifier and a single quantum prover? Our work attempts to provide answers to these questions.

First, we consider a single-prover verification protocol developed by Fitzsimons and Kashefi and show that this protocol is indeed robust with respect to deviations on the quantum state prepared by the verifier. We show that this is true even if those deviations are the result of a correlation with the prover's system. We then use this result to give a verification protocol which is *device-independent*. The protocol consists of a verifier with a measurement device and a single prover. Device-independence means that the verifier need not trust the measurement device (nor the prover) which can be assumed to be fully malicious (though not communicating with the prover). A key element in realising this protocol is a robust technique of Reichardt, Unger and Vazirani for testing, using non-local correlations, that two *untrusted* devices share a large number of entangled states. This technique is referred to as *rigidity of non-local correlations*.

Our second result is to prove a rigidity result for a type of quantum correlations known as *steering correlations*. To do this, we first show that steering correlations

can be used in order to certify maximally entangled states, in a setting in which each test is independent of the previous one. We also show that the fidelity with which we characterise the state, in this specific test, is optimal. We then improve the previous result by removing the independence assumption. This then leads to our desired rigidity result. We make use of it, in a similar fashion to the device-independent case, in order to give a verification protocol that is *one-sided device-independent*. The importance of this application is to show how different trust assumptions affect the efficiency of the protocol.

Next, we describe a protocol for fault-tolerantly verifying quantum computations, with minimal "quantum requirements" for the verifier. Specifically, the verifier only requires a device for measuring single-qubit states. Both this device, and the prover's operations are assumed to be prone to errors. We show that under standard assumptions about the error model, it is possible to achieve verification of quantum computation using fault-tolerant principles. As a proof of principle, and to better illustrate the inner workings of the protocol, we describe a toy implementation of the protocol in a quantum simulator, and present the results we obtained, when running it for a small computation.

Finally, we explore the possibility of having a verification protocol, with a classical verifier and a single prover, such that the prover is *blind* with respect to the verifier's computation. We give evidence that this is not possible. In fact, our result is only concerned with blind quantum computation with a classical client, and uses complexity theoretic results to argue why it is improbable for such a protocol to exist. We then use these complexity theoretic techniques to show that a client, with the ability to prepare and send quantum states to a quantum server, would not be able to delegate arbitrary NP problems to that server. In other words, even a client with quantum capabilities cannot exploit those capabilities to delegate the computation of NP problems, while keeping the input, to that computation, private. This is again true, provided certain complexity theoretic conjectures are true.

Pentru tatăl meu

Lay summary

Quantum computation is a new form of computation that exploits the effects of quantum physics in order to solve certain problems much faster than regular (classical) computers. It is conjectured that, for some of these problems, even the most powerful supercomputers available today would take millennia to arrive at a solution, whereas a quantum computer could do it in mere minutes. This is great news for the prospect of solving currently intractable problems, and is the impetus for developing large scale quantum computers, but it also raises a significant challenge: how can classical computers verify the correctness of quantum computations? In other words, how can a quantum computer prove to a classical agent that the solution to a certain problem is correct? While checking the correct functionality of small quantum devices, such as lasers or photo-detectors, is relatively easy, the situation is far from trivial when it comes to quantum computers. In attempting to answer this question, a number of so-called quantum verification protocols have been developed, in which a classical computer, possibly utilizing small quantum devices, attempts to verify the computations performed by one or more quantum computers. In this thesis, we study a number of aspects pertaining to these protocols:

(1) The robustness of these protocols to noise and imperfections in the utilized quantum devices. To that end, we show that certain protocols are indeed robust to noise and that verification can be achieved by simply inspecting correlations between the results of these devices and the results of the quantum computer.

(2) The performance of these protocols under varying assumptions regarding the aforementioned correlations. Here we show that certain types of correlations lead to improved performance for the verification protocols, making them better suited for practical implementations. In certain cases we are also able to show that the improved performance is optimal.

(3) Coping with erroneous results and imperfections. For this, we propose a simple construction for making verification protocols fault-tolerant (i.e. having the ability to detect and correct for errors).

(4) Limitations for cryptographic applications. Certain types of verification protocols have been inspired by cryptographic primitives for performing computations on encrypted data. We show that there is a trade-off between the desired security of these protocols and their applicability in settings where the classical verifier does not have access to trusted quantum devices.

Publications and manuscripts

The majority of this work is based on the following publications, manuscripts and one article:

1. **Robustness and device independence of verifiable blind quantum computing**. *Alexandru Gheorghiu, Elham Kashefi, Petros Wallden. New Journal of Physics, Vol. 17, No. 8, 2015.* Eprint arxiv:1502.02571.

2. **Rigidity of quantum steering and one-sided device-independent verifiable quantum computation**. *Alexandru Gheorghiu, Petros Wallden, Elham Kashefi. New Journal of Physics, Vol. 19, No. 2, 2017.* Eprint arxiv:1512.07401.

3. **Verification of Quantum Computation: An overview of existing approaches**. *Alexandru Gheorghiu, Theodoros Kapourniotis, Elham Kashefi. Theory of Computing Systems, July 2018.* Eprint arXiv:1704.08482.

4. **On the implausibility of classical client blind quantum computing**. *Scott Aaronson, Alexandru Cojocaru, Alexandru Gheorghiu, Elham Kashefi. April 2017.* Eprint arXiv:1704.08482.

5. **A simple protocol for fault tolerant verification of quantum computation**. *Alexandru Gheorghiu, Matty J. Hoban, Elham Kashefi. April 2018.* Eprint arXiv:1804.06105.

6. **Keeping Quantum Computers Honest (or Verification of Quantum Computing)**. *Alexandru Gheorghiu, Elham Kashefi. January 2018.* Article in ERCIM News 112, special theme on Quantum Computing.

Acknowledgements

As Carl Sagan once said "*science is a collaborative enterprise*". In that spirit, the research I did during my PhD, that eventually (after many sleepless nights) culminated in this thesis, has also been a collaborative enterprise. I am therefore deeply grateful to a number of people that have been part of this enterprise.

First of all, I would like to thank my supervisor, Elham Kashefi for her unwavering and enthusiastic support since the MSc days, when I first met her and we started working together. I don't think I have met anyone who is as excited or full of energy, when it comes to science, as Elham. I like to think that some of her enthusiasm has rubbed off on me and has made me into a better researcher.

I am also extremely grateful to my two co-supervisors Petros Wallden and Myrto Arapinis. Since the beginning of the PhD, Petros and Myrto have been an endless source of wisdom, insight, advice, moral support and funny Greek expressions. Petros, in particular, has also been a source of *hilarious* jokes, which, for better or worse, have heavily influenced my own humour. Having spent much of my time in his office (more than in my own office), engaging in numerous discussions ranging from quantum gravity to chess, to the differences between Greek and Romanian drinks, I have come to consider him not only a mentor, but also a great friend.

While technically not one of my supervisors, I'd be lying if I didn't mention that a good chunk of my understanding of anything quantum related came from long (and often hilarious) discussions with Matty Hoban. I am immensely thankful to him for his limitless patience in answering my barrage of (mostly silly) questions, for always being up for a fun chat and for just being an all around great guy.

It goes without saying that I am also grateful to Elham, Petros, Myrto and Matty for helping me improve this thesis with many useful comments and suggestions. In that spirit, I would also like to thank my PhD examiners Chris Heunen, Ashley Montanaro and Thomas Vidick for reading my thesis thoroughly and giving me very useful feedback to improve it. Any mistakes or inaccuracies still present in this work are entirely my own.

Throughout my time in Edinburgh, I've been lucky enough to have a number of very supportive friends that were crucial to my successful completion of the PhD: my long time friend Alex Cojocaru with whom I've explored all the food places around my office, played numerous games of ping pong and football and had many productive discussions in both English and Romanian; my jolly good friend Dan Mills who has been a plentiful source of wisdom (especially when it comes to politics), humour and British etiquette, with whom I've had fruitful scientific and philosophical discussions/debates and to whom I am deeply grateful for reading my thesis and giving me feedback; my gamer friend Borislav Ikonomov (known to his friends as *crazy hair dude*) who has given me a renewed appreciation for boardgames and anime, has enlightened me on aspects pertaining to machine learning and probability theory and with whom I've had many fun philosophical discussions; my exhaustingly enthusiastic friend Stanislav Stratiev who shares my terrible sense of humour when it comes to puns and who is constantly schooling me on quantum field theory and condensed matter physics; my former flatmate Pia Kullik, with whom I've watched more sci-fi than anyone else and who has tolerated my endless ramblings and various quirks; my other former flatmate (and the first person I befriended in Edinburgh), Emma Chisholm, with whom I've watched more comicbook movies than anyone else, baked a cake and had endless fun discussions.

I am also grateful to all the "quantum people" that I've met and/or worked with in Edinburgh: Theodoros Kapourniotis, Leonardo Disilvestro, Chris Heunen, Martti Karvonen, Brian Coyle, Ellen Derbyshire, Ieva Cepaite, Mahshid Delavar as well as the members of the Paris quantum group: Tom Douce, Marc Kaplan, Damian Markham, Eleni Diamanti, Luka Music, Léo Colisson and Ulysse Chabaud.

I very much enjoyed my time in Edinburgh and will definitely miss it. The Informatics Forum has been an amazing place in which to work, to have engaging discussions, and to exchange ideas. With that, I'd like to thank the School of Informatics and LFCS for creating an excellent research environment. I would also like to thank Laboratoire d'Informatique de Paris 6, of Université Pierre et Marie Curie, for hosting me on my many trips there to visit the Paris quantum group.

There are also a number of friends, outside of Edinburgh, that I've been lucky to have and to whom I'd like to express my thanks: Andrei Sfrenţ, Liviu Chirondojan, Vladimir Diaconescu, Dan Moşanu, George-Daniel Mitra, Irina Grigorescu, Anaïs Baras, Cristina Opriceană and Vivian Marie.

Finally, I am grateful to the many wonderful people that I've met at conferences, workshops and universities, with whom I've had productive exchanges: Scott Aaronson, Thomas Vidick, Urmila Mahadev, Umesh Vazirani, Kousha Etessami, Dominique Unruh and Ronald de Wolf.

Contents

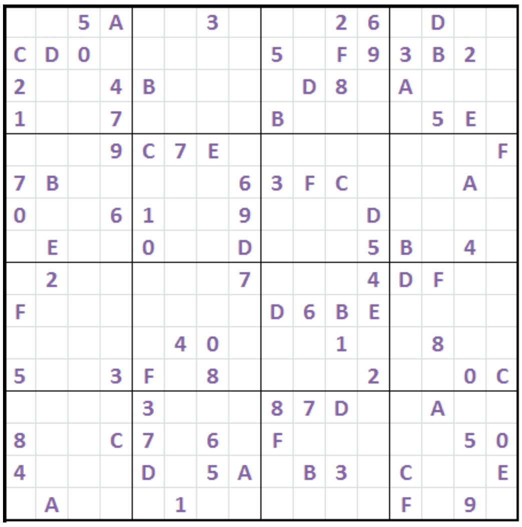
Chapter 1

Introduction

> **Principal Skinner:** There's nothing more exciting than science. You get all the fun of sitting still, being quiet, writing down numbers, paying attention. Science has it all.
>
> — The Simpsons, Season 6, Episode 14

Quantum information theory has radically altered our perspective on quantum mechanics. Initially, research into quantum mechanics was devoted to explaining phenomena as they are observed in nature. But recently a new direction has emerged, having to do with designing and creating quantum systems for computation, information processing, communication, and cryptography, among many other tasks [1]. For quantum computation in particular, there was a remarkable realisation that quantum interference, "*the heart of quantum mechanics*", as Feynman described it [2], could be harnessed in order to solve certain problems exponentially faster than with the best known classical algorithms. What kinds of problems? Probably the most well-known example is *factoring integers*, for which the fastest classical algorithm requires time which scales sub-exponentially in the size of the input, whereas Shor's quantum algorithm requires only polynomial time [3]. But factoring achieved its notoriety due, in large part, to its importance in cryptography. After all, cryptographic protocols which rely on the hardness of this problem, such as RSA, are ubiquitous when performing e-commerce, digital signatures and essentially any form of secure communication over the Internet. However, the problems that are, arguably, more natural for quantum computers to solve efficiently, concern simulating physical systems. Indeed, this was the original impetus for the development of quantum computing [4]. For instance, quantum computers could be used to dynamically simulate chemical reactions and efficiently compute quantities of interest for these reactions (such as thermal reaction rates) [5,6]. By "efficiently" we mean that the time it takes to solve the problem scales polynomially in the size of the input.

Let us, for a moment, consider a question that is seemingly unrelated to quantum computation: do all solvable problems have a *recognisable* solution? By recognisable, we mean that there exists some procedure through which a classical system, such as a person or a regular computer, when presented with a candidate solution to a problem, could say whether the solution is correct or not. Additionally, this process of *verifying* the candidate solution should be more



		5	A			3				2	6		D		
C	D	0						5		F	9	3	B	2	
2			4	B					D	8		A			
1			7					B					5	E	
			9	C	7	E									F
7	B						6	3	F	C				A	
0			6	1			9				D				
	E			0			D				5	B		4	
	2						7				4	D	F		
F								D	6	B	E				
					4	0				1			8		
5			3	F		8					2			0	C
				3				8	7	D			A		
8			C	7		6		F						5	0
4				D		5	A		B	3		C			E
	A				1							F		9	

Figure 1.1: Hexadecimal Sudoku puzzle from [7]. This is an example of a problem with a recognisable solution. It can be difficult to solve the Sudoku, but one can immediately check whether a given solution is correct or not. The reader might find it amusing to solve this puzzle.

efficient than the task of finding a solution to the problem. More formally, we require this verification procedure to run in polynomial time. Upon first thought, we realise that many of the problems that we encounter around us, do have recognisable solutions. Sudoku puzzles are one such example. Solving a Sudoku puzzle can take anywhere from a few minutes to an hour, depending on the difficulty of the puzzle. But if we're presented with a candidate solution to the puzzle, we can immediately check whether it is valid or not. This fact remains true if we scale up the size of the Sudoku puzzle (see Figure 1.1 for an example of a hexadecimal Sudoku). Other puzzles, such as crosswords, Rubik's cube and jigsaw puzzles share the same features. Proving (or disproving) a mathematical statement is another example. Given the statement, it can be quite challenging to produce a proof. In certain cases, this process can take hundreds of years, as was the case with Fermat's last theorem [8]. However, when presented with a purported proof, a mathematician would take far less time to check its validity. In spite of these, and many other examples, there are also problems that *do not* admit recognisable solutions. For instance, given two regular expressions[1], the problem of deciding whether they match the same set of strings does not, in general, have a recognisable solution [10]. Many other examples exist, but this one is sufficient to provide an answer to our question: not all problems admit a solution that can be checked efficiently.

So what does this have to do with quantum computation? Given the conjecture that for certain problems, that are efficiently solvable by quantum algorithms, there are no efficient classical algorithms, it is only natural to ask whether these problems at least have recognisable solutions. After all, what good is a machine that can solve difficult problems if we have no way of checking that the solutions

[1] A regular expression is a sequence of characters, defining a pattern. Certain strings of characters will match this pattern while others will not. See [9] for more details.

are correct? To make things fair, given that quantum mechanics involves randomness and probabilities, we relax "recognisable solution" to "probabilistically recognisable solution". By this we mean that the process of checking the solution is allowed to give an incorrect verdict as long as this happens with small probability (say, less than 1/3).

At this point, it is convenient to introduce some helpful notation, which will be formally defined in the next chapter. We will refer to the set of all problems that can be solved efficiently on a classical computer as BPP. The corresponding set for quantum computers is BQP and lastly, the set of problems with probabilistically recognisable solutions is denoted MA[2]. Clearly, $\mathsf{BPP} \subseteq \mathsf{BQP}$, since any problem that can be solved efficiently, classically, can also be solved efficiently on a quantum computer. Additionally, $\mathsf{BPP} \subseteq \mathsf{MA}$, since if one can compute a solution efficiently, then one can also verify it efficiently. We would like to know whether $\mathsf{BQP} \subseteq \mathsf{MA}$. We certainly know of problems that are contained in both BQP and MA but are not believed to be contained in BPP. Factoring integers is one such example. The solution is recognisable since, when given potential factors, one can simply multiply them and check if the result matches the number to be factored. In fact, factoring is a particular instance of a more general problem known as the *Abelian Hidden Subgroup Problem (AHSP)* [1] which also lies at the intersection of BQP and MA but is not believed to be contained in BPP [3].

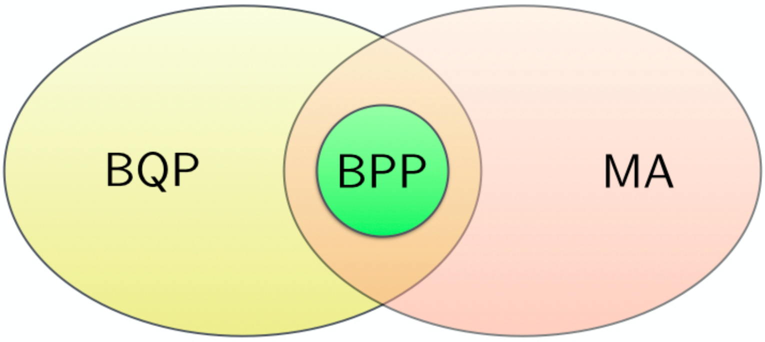


Figure 1.2: The suspected relationship between problems solvable efficiently classically (BPP), those solvable efficiently quantumly (BQP) and those having recognisable solutions (MA).

In general, however, it is believed that BQP is not contained in MA. Hence, the aforementioned natural problem for quantum computers, of simulating a quantum

[2] The names of these classes are abbreviations for bounded-error probabilistic polynomial time (BPP), bounded-error quantum polynomial time (BQP) and Merlin-Arthur (MA), respectively [11]. The name Merlin-Arthur was chosen in the idea that Merlin, an all-powerful wizard, could solve any problem and present the solution to Arthur who would have to check its correctness. Since Arthur is not a wizard, this captures the idea that the solution needs to be efficiently verifiable by limited agents, such as medieval kings or classical computers.

[3] Strictly speaking, factoring is not an instance of AHSP. Rather, computing the order of an element in the multiplicative group of integers modulo n is an instance of AHSP. However, factoring can be reduced efficiently, classically, to this latter problem. In other words, having a procedure to solve order finding leads to an efficient algorithm for factoring, using that procedure [1].

system, is believed to lie outside of the class MA. There is no known procedure to efficiently check such a simulation, on a classical computer. The separation between the two classes is conjectured due to results in computational complexity theory which provide compelling evidence that there are problems in BQP that are not contained in MA (as well as the converse) [12–14]. Thus, the assumed relationship between these classes is the one depicted in Figure 1.2. This raises an important question, which has spawned an entire field of research [15] and which sets the stage for our results:

If problems that can be solved efficiently by quantum computers do not admit recognisable solutions, how then can one verify the correctness of quantum computations?

We will refer to this as *the question of quantum verification.*

1.1 Quantum verification

At first, one might be tempted to dismiss the question of quantum verification and say that as long as each component of a quantum computer has been tested and works correctly, there is no need to worry about the validity of the device's results. However, this is not an adequate solution. Rather than giving a dry explanation as to why, let us consider a fictitious dialogue between a sceptic of a general quantum verification technique and a proponent of it:

Proponent: Since we don't believe that all problems in BQP have recognisable solutions, it is imperative to come up with some other technique through which we can verify quantum computations. Perhaps, a technique by which a *verifier* could interact with a quantum computer, while it is performing the computation, thereby verifying it in a step-by-step fashion.

Sceptic: That sounds complicated and unnecessary. We can simply test that each component of the quantum computer works as expected, before putting everything together. After all, that's what we do with classical computers and they work fine.

Proponent: Yes, but the point of verifying general quantum computations is much more profound than merely checking that the components work. Quantum computers would provide us with one of the most stringent tests of the laws of quantum mechanics.

Sceptic: How so?

Proponent: Well, when someone proposes a new physical theory, the way we test it is by making predictions from the theory and then checking to see whether the predictions are true in the real world. However, because of the exponential overhead in simulating quantum systems, we can't do that anymore, at least not in any reasonable amount of time. I mean, sure, we can do it for small enough systems and even come up with decent approximations for slightly larger ones, but once you set up an experiment involving a complex quantum system with many degrees of freedom it becomes completely infeasible to predict the outcome of that experiment. That's, basically, what a quantum computation is. Just like telescopes allow us to probe the regimes of large distances and time scales, or particle accelerators allow us to probe short distances and time scales and high energies, a quantum computer would allow us to probe the regime of *high complexity.* If there is new physics in that regime, the only way we can make sense of it is if we have a general verification technique.

Sceptic: Ok, that's all well and good, but it sounds very hypothetical and I'm more of a pragmatist. While I could agree that in the long term we might need such techniques to probe this "high complexity regime", for now it's fine to just test the components. There's no need to complicate things.

Proponent: Actually, even if you just care about the short term, or are still not convinced, I can think of a few more compelling reasons. For starters, you say "let's just test each individual component". That's fine, except that each individual component, when considered separately, isn't capable of producing the interesting features that seem to be necessary when running a quantum algorithm. For instance, it seems that *entanglement* is a key ingredient in these algorithms. But producing the types of highly entangled, multi-particle states that appear in, say, Shor's algorithm, requires putting the components together. So, in fact, each component working separately is not interesting. What we want to know is whether they work well when combined together.

Sceptic: Fair enough...

Proponent: Another reason has to do with security. IBM and Rigetti have made quantum computers available to users around the world over the Internet [16,17]. What if their devices are compromised remotely? What if you, as a user, delegate a problem to one of these computers (or, rather, to future and more powerful devices) and your communication is intercepted or altered by some malicious party in the middle. How do you know whether the results you're getting back are correct?

Sceptic: Ok, you've convinced me. I'm glad we had this discussion.

Presumably, the two then proceed to debate what subjective experience an AI would have, if it were running on a quantum computer [18].

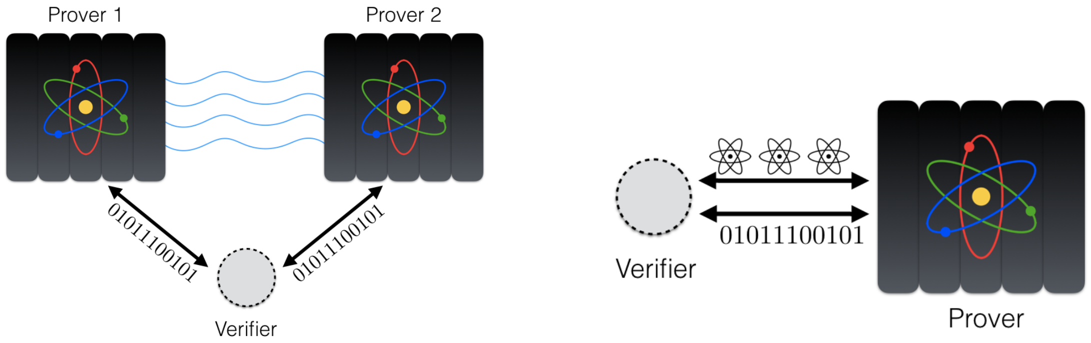
The question of verification was promoted as a complexity challenge with a \$25 prize, by Aaronson on his blog: "*If a quantum computer can efficiently solve a problem, can it also efficiently convince an observer that the solution is correct? More formally, does every language in the class of quantumly 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*)?*" [19]. He credits Gottesman with first formalising this question, in a complexity theoretic sense, at a 2004 conference. Vazirani, then emphasised the importance of the question, not only from the perspective of complexity theory, but from a philosophical point of view [20]. In 2007, he related it to the question of whether quantum mechanics is a *falsifiable theory*, and suggested that a computational approach could provide an answer. This perspective was explored in depth by Aharonov and Vazirani in [21].

So what is the answer? At the time of writing, we still do not know. Or rather, we do not know an answer to Gottesman and Aaronson's version of the question, i.e. whether a completely classical BPP verifier, interacting with a BQP server, can check the correctness of general quantum computations. We do know, however, that under slight alterations of this setup, an interactive verification protocol does exist. Before giving examples, we would like to re-emphasise that the question of verification only makes sense under certain, widely accepted, complexity theoretic assumptions. One assumption is that $\mathsf{BPP} \neq \mathsf{BQP}$, since otherwise the classical client could simply perform the BQP computations efficiently, without the need of a quantum computer. One would assume the second assumption is that BQP problems do not have recognisable solutions (i.e. $\mathsf{BQP} \not\subseteq \mathsf{MA}$). In fact, things are more subtle. It could very well be that $\mathsf{BQP} \subseteq \mathsf{MA}$, and yet a BQP server is unable to convince a BPP machine of the correctness of a particular solutions. Think back to the Sudoku example. When one is presented with a completion of a Sudoku puzzle, one also requires a *proof* that the completion is indeed correct. This proof entails showing that the candidate solution satisfies the constrains of the Sudoku puzzle (each row, column and local square contains all the digits exactly once). The BPP machine recognises the Sudoku solution by *verifying this proof*. For BQP problems, it could be that such a proof exists, and could be verified efficiently by a classical computer, but that *the proof cannot be efficiently generated by a quantum computer*. While this may seem counter-intuitive, it is still a logical possibility. A quantum computer could efficiently solve an MA problem and yet be unable to produce a proof that that solution is indeed correct. A candidate problem that is believed to be of this type is Childs et al's glued trees problem [22]. Our second assumption is therefore that there are BQP problems for which producing a proof, verifiable by a BPP machine, cannot be done in BQP.

Let us now return to how one might change the verification setup so as to have an interactive protocol. For convenience, we will refer to the classical verifier as Alice and to the quantum server, which we will also call *prover*, as Bob. The first approach was provided by Aharonov, Ben-Or and Eban [23]. They showed that if one equips Alice with the ability to prepare quantum states, which she then sends to Bob, then it is possible to verify arbitrary quantum computations. Crucially, the states that Alice prepares would not allow her to perform universal quantum computations. In other words, while she can prepare quantum states, her computational capability is still limited to BPP. Other protocols, that also relied on Alice's ability to prepare quantum states, were later developed [24, 25]. We will be especially interested in such a protocol, developed by Fitzsimons and Kashefi [24]. Collectively, all of these protocols are categorised as *single-prover prepare-and-send* protocols.

Another way of changing the original setting, of the verifier and the prover, is to endow Alice with a measurement device, instead of a preparation device.

This would allow her to measure quantum states, however she would not have the ability to prepare them and her computational power is still restricted to BPP. The first such protocol was proposed by Morimae [26] and, just as with the prepare-and-send setting, a number of other protocols in this model were later developed [27–30]. Such protocols are referred to as *single-prover receive-and-measure* protocols.



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

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

Figure 1.3: Models for verifiable quantum computation

The verification setting can also be changed by having multiple quantum provers, instead of one, while keeping the verifier (as well as her communication with the provers) entirely classical. In such a setting, the provers are assumed to be non-communicating but sharing entanglement. That is, they are allowed to share quantum correlations but are otherwise prevented from interacting with each other. The first approach of this type was proposed by Reichardt, Unger and Vazirani [31]. The family of such protocols is known as *multi-prover entanglement-based* protocols and, once again, includes a number of different approaches [32–36].

Throughout this thesis we will be primarily concerned with these three families of verification protocols, for which we give a pictorial representation in Figure 1.3 (prepare-and-measure and receive-and-measure protocols are merged together in Subfigure 1.3b).

Recently, a fourth option has also emerged. In a breakthrough result of Mahadev, it was shown that it is possible for a fully classical verifier to delegate and verify arbitrary BQP computations to a quantum prover [37], provided a certain problem, known as *Learning With Errors (LWE)*, is not contained in BQP [38]. Since this problem is indeed believed to not be efficiently solvable by quantum computers, Aaronson has awarded the $25 prize to Mahadev for the protocol [39]. A similar approach to that of Mahadev, had also been used by Bremner and Shepherd to verify a more restricted class of quantum computations known as instantaneous quantum computations [40]. Again, this was based on the assumption that a particular problem cannot be efficiently solved using instantaneous quantum computations. We will collectively refer to such approaches as approaches that are based on computational assumptions. Throughout this thesis, however, we will be interested in protocols that do not rely on computational assumptions and so, we will primarily reference protocols from the three families, mentioned above.

1.2 Robust verification

As mentioned, the field of quantum verification arose from the question of whether a classical BPP verifier can check efficient quantum computations performed by a BQP prover. While that still remains one of the major open problems of the field (at least for the case of no computational assumptions), the development of the aforementioned protocols has raised a number of other interesting problems and challenges. For instance, quantum systems are notoriously difficult to isolate from their environment. Such an isolation is necessary to preserve quantum information. In other words, the interaction with the environment acts as a sort of *noise* that can corrupt quantum states, leading to erroneous computations. How do verification protocols behave in the presence of noise? What can be done in order to mitigate the effects of it? A solution is provided by *quantum error-correcting codes*, which, together with specialised operations for preparing states, performing quantum gates and making measurements lead to *fault tolerant* quantum computation [1]. However, fault tolerance typically requires that quantum information is stored in larger quantum systems. This isn't a problem for the prover, that is assumed to be a general quantum computer, but it can become a problem for the verifier if we wish to keep her quantum capabilities as limited as possible. Thus, the challenge is to develop a fault tolerant verification protocol, while at the same time keeping the verifier as "quantum limited" as in the non-fault tolerant case.

Another challenge relates to trust assumptions. The whole point of verification is that the verifier does not trust the prover and is trying to validate his computations. But, in the prepare-and-send or receive-and-measure protocols, why should the verifier trust her own quantum device? An argument could be made that she can test her device prior to engaging in the protocol with the prover. However, such tests, performed in isolation, would not reveal any correlations that had been pre-established between that device and the prover. Such correlations could then be exploited by the prover, during the protocol, in order to convince the verifier to accept an incorrect result. One could object and say that it is unfounded to consider such deliberately malicious behaviour by the prover. However, as was pointed out in the Proponent vs. Sceptic discussion, quantum verification protocols serve two main purposes. The first one is to test quantum mechanics itself, and if we are to be fully scientific and not susceptible to biases, then this test requires the most minimal assumptions possible. It should be our job to convince even the most radical sceptic that one can indeed verify (or falsify) the predictions of quantum mechanics in the regime of high complexity. This will become especially important when experimentalists are confronted with this regime, as they improve their ability to control and manipulate quantum systems. The second purpose relates to the role of quantum computers in the Internet. As mentioned, quantum computers are already available for public or commercial use through the Internet. With the development of quantum communication, it is reasonable to imagine that in the future users will have the ability to send or receive quantum states to or from quantum servers. This then creates the necessary environment for using existing verification protocols, but it also raises the possibility that malicious agents would attempt to compromise these protocols. Devices for secure quantum communication are being continuously tested for the possibility of such attacks, most notably by so-called *quantum hackers* [41–43]. It is therefore a necessity to ensure the correctness of these protocols in a malicious setting and with minimal trust assumptions.

This latter reason has motivated a cryptographic approach to verification protocols, which will feature prominently throughout the thesis. In turn, this raises further challenges such as the verifier wanting to encrypt and hide her computation from the prover, a condition which is known as *blind quantum computation*. Under what conditions can this be achieved? How does this relate to the other aspects of verification, such as fault tolerance?

All of these are challenges in developing robust verification protocols. By "robust" we do not mean only robustness to noise, but also robustness with respect to varying trust assumptions. This thesis is concerned with addressing these challenges, in some cases providing solutions to them, while in other cases characterising their impact on existing protocols and informing the direction of future research.

1.3 Thesis overview

The thesis is organised as follows:

(1) Chapter 2 presents the background concepts and notation that will be used throughout the thesis. These include basics of quantum information and quantum computation, complexity theory, measurement-based quantum computation, universal blind quantum computation and a number of verification protocols that will be used in our results. The contents of the chapter are based on a survey paper of verification protocols which was completed in collaboration with Theodoros Kapourniotis and Elham Kashefi [15].

(2) In Chapter 3, we prove that the prepare-and-send protocol, developed by Fitzsimons and Kashefi, is robust with respect to an imperfect preparation device. This will also account for correlations between that device and the prover's system. We then use this to show how the protocol can be turned into a *device-independent* verification protocol. Device-independent, in this context, means that the verifier retains a quantum device (a measurement device) however, the device need not be trusted and can be assumed to be malicious and correlated with the prover. We show that, in this setting, verification is still possible. This is achieved, using a robust technique of Reichardt, Unger and Vazirani for testing that two *untrusted* devices share a large number of entangled states, using non-local correlations. We refer to this as *rigidity of non-local correlations*. The contents of the chapter are based on work done in collaboration with Petros Wallden and Elham Kashefi, in [30].

(3) In Chapter 4, we adapt the rigidity technique to the setting in which only one device is untrusted (the server) whereas the other is trusted (the measurement device). In this case, the types of correlations that allow for entanglement testing are known as *EPR-steering correlations*, or simply steering correlations. We prove a number of results concerning these correlations. First, that they can be used in order to test for pairs of entangled states, in a setting in which each test is independent of the previous one. We also show that the fidelity of this sort of test is optimal. Next, we prove how one can test for multiple entangled states, when removing the independence assumptions. We then use this result, in the same manner as in Chapter 3, to give a verification protocol which is *one-sided device-independent*. This will illustrate how different trust assumptions affect the efficiency of verification protocols. Lastly, we give an argument for why the entangled states that are useful for these types of verification protocols should be equivalent to maximally entangled states. This chapter is based on joint work with Petros Wallden and Elham Kashefi [33].

(4) Chapter 5 is dedicated to giving a fault tolerant verification protocol in which the verifier's quantum capabilities are minimal. We first discuss how noise might negatively affect a protocol that does not utilise a fault tolerant construction. We then give a simple construction for making a particular type of protocol, known as *post hoc verification protocol*, fault tolerant. A proof of principle example of the protocol, implemented in a quantum simulator, is also presented. This is based on joint work with Matty Hoban and Elham Kashefi [44].

(5) In Chapter 6 we explore the possibility of having a verification protocol, with a classical verifier and a single prover, such that the prover is blind with respect to the verifier's computation. We give evidence that this is not possible. In fact, we merely examine the case of blind quantum computation with a classical client, without verification, and use complexity theoretic results to argue that it is improbable for such a protocol to exist. This then implies that verification, in this setting, is also improbable. We then use these complexity theoretic techniques to show that a client, with the ability to prepare and send quantum states to a quantum server, would not be able to delegate arbitrary MA problems to that server. In other words, even a client with quantum capabilities cannot exploit those capabilities to delegate arbitrary problems with recognisable solutions, while also keeping her input private. This is again true, provided certain complexity theoretic statements are true. The work was done in collaboration with Scott Aaronson, Alexandru Cojocaru and Elham Kashefi, in [45].

(6) We conclude in Chapter 7, where we summarise the main findings of our work and discuss future directions and open problems.

Chapter 2

Preliminaries

> **Morty:** What is it, Rick? Is it the quantum carburettor or something?
> **Rick:** Quantum carburettor? Jeez, Morty. You can't just add a sci-fi word to a car word and hope it means something. Hmm, looks like something's wrong with the micro-verse battery.
>
> — Rick and Morty, Season 2, Episode 6

2.1 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 [1, 46, 47].

2.1.1 Basics of quantum mechanics

States, unitaries, measurement

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 $\mathcal{H}^{\perp}$. If $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ are two quantum states, then, taking the dual of $|\psi\rangle$ and acting on $|\phi\rangle$, a number which is denoted $\langle\psi|\phi\rangle$, is defined as:

$$\langle\psi|\phi\rangle \equiv \langle\psi, \phi\rangle \tag{2.1}$$

where $\langle\psi, \phi\rangle$ is the inner product of $|\psi\rangle$ and $|\phi\rangle$. In other words, the action of dual states is defined with respect to the inner product over $\mathcal{H}$.

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 $|1\rangle$. We will also be working in the basis $(|+_\theta\rangle, |-_\theta\rangle)$, where $\theta \in [0, 2\pi]$ and:

$$|+_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle) \quad |-_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\theta}|1\rangle) \tag{2.2}$$

When $\theta = 0$, we simply denote these states as $|+\rangle$ and $|-\rangle$, respectively.

Gluing together systems to express the states of multiple qubits is achieved through the *tensor product*, denoted $\otimes$. The notation $|\psi\rangle^{\otimes n}$ denotes a state comprising of n copies of $|\psi\rangle$. If, for a given state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$, there are no $|a\rangle \in \mathcal{H}_1$ and $|b\rangle \in \mathcal{H}_2$ such that $|\psi\rangle = |a\rangle \otimes |b\rangle$, then we say that such a 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* (also referred to as *EPR state*):

$$|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2.3}$$

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 $|\psi\rangle \to U |\psi\rangle$. 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 linear operators $\{M_i\}_i$ acting on the state space of the system to be measured and satisfying:

$$\sum_i M_i^\dagger M_i = I \tag{2.4}$$

Notice that $M_i^\dagger M_i$ is a hermitian, positive operator. The label i indicates a potential measurement outcome. Given a state $|\psi\rangle$ to be measured, the probability of obtaining outcome i is:

$$p(i) = \langle\psi| M_i^\dagger M_i |\psi\rangle \tag{2.5}$$

and the state of the system after the measurement will be:

$$M_i |\psi\rangle / \sqrt{p(i)} \tag{2.6}$$

If we are only interested in the probabilities of the different outcomes and not in the post-measurement state then we can write $E_i = M_i M_i^\dagger$ and we will refer to the set $\{E_i\}_i$ as a *positive-operator valued measure* (POVM). When performing a measurement in a 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 $M_i^2 = M_i$ and $M_i^\dagger = M_i$.

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, on $\mathcal{H}$, we know that there exists a basis $\mathcal{B} = \{|i\rangle\}_i$ of $\mathcal{H}$, such that:

$$O = \sum_i \lambda_i |i\rangle \tag{2.7}$$

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 $\mathcal{B}$[1]. 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 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[2]. 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 observations 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 semi-definite, trace one, hermitian operators. For a pure state, $|\psi\rangle$, the corresponding density matrix is:

$$\rho = |\psi\rangle \langle\psi| \tag{2.8}$$

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:

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| \tag{2.9}$$

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 thesis, 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 a quantum system. What this means is that for any basis

[1] 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.

[2] 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 [48] for more details.

$\{|v_i\rangle\}_i$ of the Hilbert space of dimension d, the maximally mixed state is:

$$\frac{I}{d} = \frac{1}{d}\sum_{i=1}^{d} |v_i\rangle \langle v_i| \tag{2.10}$$

Equivalently, any projective measurement, specified by a complete 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:

$$p(i) = Tr(M_i M_i^\dagger \rho) \tag{2.11}$$

and the post-measurement state will be:

$$M_i \rho M_i^\dagger / p(i) \tag{2.12}$$

Partial trace and 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. Taking the partial trace is a linear operation, 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:

$$\rho_1 = Tr_2(\rho_1 \otimes \rho_2) \qquad \rho_2 = Tr_1(\rho_1 \otimes \rho_2) \tag{2.13}$$

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 define the partial trace. For if we take any general density matrix, ρ, on $\mathcal{H}_1 \otimes \mathcal{H}_2$, expressed as:

$$\rho = \sum_{i,i',j,j'} a_{ii'jj'} |i\rangle_1 \langle i'|_1 \otimes |j\rangle_2 \langle j'|_2 \tag{2.14}$$

where $\{|i\rangle\}$, $\{|i'\rangle\}$ are orthonormal bases for $\mathcal{H}_1$ and $\{|j\rangle\}$, $\{|j'\rangle\}$ are orthonormal bases for $\mathcal{H}_2$, tracing our subsystem 2 yields:

$$Tr_2(\rho) = Tr_2\left(\sum_{i,i',j,j'} a_{ii'jj'} |i\rangle_1 \langle i'|_1 \otimes |j\rangle_2 \langle j'|_2\right) = \sum_{i,i',j} a_{ii'jj} |i\rangle_1 \langle i'|_1 \tag{2.15}$$

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 $dim(\mathcal{H}_1) = dim(\mathcal{H}_2)$[3] and:

$$Tr_2(|\psi\rangle \langle\psi|) = \rho \tag{2.16}$$

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

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:

$$|\phi\rangle = (I \otimes U) |\psi\rangle \tag{2.17}$$

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, $\mathcal{O} : \mathcal{D}(\mathcal{H}_{in}) \rightarrow \mathcal{D}(\mathcal{H}_{out})$, 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 $\mathcal{O} \otimes I$, where I acts on $\mathcal{H}_E$, takes positive states to positive states, for all extensions $\mathcal{H}_E$.

3. **Trace preserving**: $Tr(\mathcal{O}(\rho)) = Tr(\rho)$.

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:

$$\mathcal{O}(\rho) = \sum_i K_i \rho K_i^\dagger \tag{2.18}$$

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

$$\sum_i K_i^\dagger K_i = I \tag{2.19}$$

CPTP maps are also referred to as *quantum channels.* Additionally, we also mention *isometries* which are CPTP maps $\mathcal{O}$ for which $\mathcal{O}^\dagger \circ \mathcal{O} = I$.

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

$$\langle \Phi(\psi) | \phi \rangle = \langle \psi | \Phi^\dagger(\phi) \rangle \tag{2.20}$$

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

$$\Phi^\dagger \circ \Phi = id \tag{2.21}$$

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

The tensor product of isometries (or linear maps in general) is defined as follows. Let $\Phi_1 : \mathcal{H}^1_{in} \rightarrow \mathcal{H}^1_{out}$ and $\Phi_2 : \mathcal{H}^2_{in} \rightarrow \mathcal{H}^2_{out}$ be two isometries. The *product* isometry $\Phi : \mathcal{H}^1_{in} \otimes \mathcal{H}^2_{in} \rightarrow \mathcal{H}^1_{out} \otimes \mathcal{H}^2_{out}$, which we write as $\Phi = \Phi_1 \otimes \Phi_2$, is defined (linearly) by its action on a basis of $\mathcal{H}^1_{in} \otimes \mathcal{H}^2_{in}$. Specifically, let $\{|i\rangle\}_i$

be an orthonormal basis of $\mathcal{H}_{in}^1$ and $\{|j\rangle\}_j$ be an orthonormal basis of $\mathcal{H}_{in}^2$. We then have that:

$$\Phi(|i\rangle \otimes |j\rangle) \equiv \Phi_1(|i\rangle) \otimes \Phi_2(|j\rangle) \tag{2.22}$$

Consider now two Hilbert spaces $\mathcal{H}_{in} = \bigotimes_{i=1}^n \mathcal{H}_{in}^i$ and $\mathcal{H}_{out} = \bigotimes_{i=1}^n \mathcal{H}_{out}^i$. We say that an isometry $\Phi : \mathcal{H}_{in} \to \mathcal{H}_{out}$ is a *local isometry* (with respect to the n-partitioning of spaces $\mathcal{H}_{in}$ and $\mathcal{H}_{out}$) if there exist isometries $\Phi_i : \mathcal{H}_{in}^i \to \mathcal{H}_{out}^i$ such that $\Phi = \Phi_1 \otimes \Phi_2 ... \otimes \Phi_n$.

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 generalises *variation distance* for probability distributions. Recall that if one has two probability distributions $p(x)$ and $q(x)$, over a finite sample space denoted Ω, the variation distance between them is defined as:

$$D(p, q) = \frac{1}{2} \sum_{x \in \Omega} |p(x) - q(x)| \tag{2.23}$$

Informally, this represents the largest possible difference between the probabilities that the two distributions can assign to some event E, where $E \subseteq \Omega$. The quantum analogue of this, for density matrices, is:

$$TD(\rho_1, \rho_2) = \frac{1}{2} Tr\left(|\rho_1 - \rho_2|\right) \tag{2.24}$$

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 for a measurement basis. So, in fact, the trace distance is the *supremum* over all possible measurements of the variation distance between the corresponding probability distributions[4].

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. Universal quantum computation is achieved by considering a fixed set of quantum gates which can approximate any unitary operation up to a chosen precision[5].

[4] For this reason, an alternative definition of trace distance is: $TD(\rho_1, \rho_2) = \frac{1}{2} \sup_U Tr\left(\left|U\rho_1 U^\dagger - U\rho_2 U^\dagger\right|\right)$.

[5] To be more precise, we say that a given set S is universal if the following is true: for any n-qubit unitary U (i.e. U is a matrix of size $2^n \times 2^n$) and any $\epsilon > 0$, there exists a finite number of matrix products and tensor products of elements of S, yielding a matrix $\tilde{U}$, such that: $max_{|\psi\rangle} ||(U - \tilde{U})\,|\psi\rangle\,|| \leq \epsilon$.

The most commonly used universal set of gates is the following:

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \; \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \; \mathsf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \; \mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \; \mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.25}$$

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 $|0\rangle|\psi\rangle \rightarrow |0\rangle|\psi\rangle$, $|1\rangle|\psi\rangle \rightarrow |1\rangle U|\psi\rangle$. 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 states. While they are not part of the standard universal set, we will also encounter the following additional gates:

$$\mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad \mathsf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{2.26}$$

which are known as the Pauli Y gate and the S or phase gate, respectively. Note that $\mathsf{S} = \mathsf{T}^2$.

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

$$\mathbb{P}_n = \{\alpha\, \sigma_1 \otimes ... \otimes \sigma_n | \alpha \in \{+1, -1, +i, -i\}, \sigma_i \in \{I, \mathsf{X}, \mathsf{Y}, \mathsf{Z}\}\} \tag{2.27}$$

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:

$$\mathfrak{C}_n = \{U \in U(2^n) | \sigma \in P_n \implies U\sigma U^\dagger \in P_n\} \tag{2.28}$$

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. Operationally they can be obtained through combinations of the Pauli gates together with H, CNOT and S, in which case they are referred to as *Clifford circuits.* We note that the T is not a Clifford operation. However, Clifford circuits combined with the T give us a universal set of unitaries.

Having a universal gate set, an n-qubit quantum computation can then be represented as a circuit comprising of gates from that set, acting on an n-qubit input state. As an example, a 2-qubit circuit is shown in Figure 2.1.

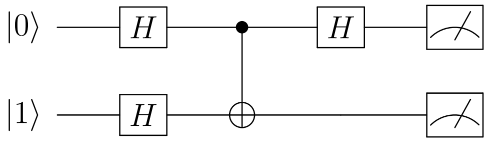


Figure 2.1: Simple quantum circuit

The circuit involves the application of two Hadamard gates, followed by a CNOT (top qubit is the control, bottom qubit is the target), followed by a Hadamard on the top qubit. The boxes at the end of the circuit represent measurements in the computational basis. Classical outputs from quantum circuits are always obtained by performing a measurement. In the circuit model, all measurements can be postponed to the end of the circuit if one is allowed to add extra qubits, known as *ancilla*.

Bloch sphere

Another important concept that we will utilise is the *Bloch sphere*, which offers a useful geometric picture for visualising single qubit states. Any such state is represented as a point on the surface of the sphere. In Figure 2.2, one can see a visualisation of the Bloch sphere together with the states $|0\rangle, |1\rangle$, the eigenstates of Z, as well as $|+\rangle, |-\rangle$, the eigenstates of X and $|+_{\pi/2}\rangle, |-_{\pi/2}\rangle$, the eigenstates of Y. All of the previously mentioned single-qubit operations can be viewed as rotations of this sphere. Specifically, 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. The T gate, corresponds to a rotation by $\pi/4$ radians around the Z axis and the S corresponds to a rotation by $\pi/2$ around the Z axis.

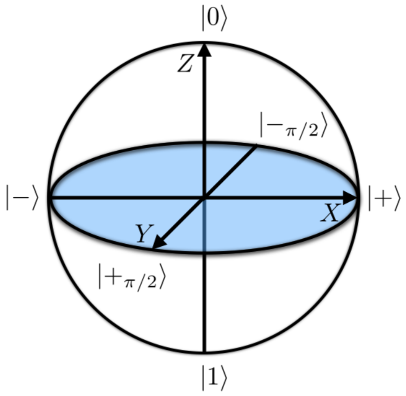


Figure 2.2: Bloch sphere

Notice that the states $|+_\phi\rangle, |-_\phi\rangle$, for all $\phi \in [0, 2\pi]$, lie on the boundary of 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 $|+_{\pi/2}\rangle, |-_{\pi/2}\rangle$ 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 $(|+_\phi\rangle, |-_\phi\rangle)$ and wish to measure the state $|+_\theta\rangle$. It can be shown that the probability of the state being projected to $|+_\phi\rangle$ is $cos^2((\phi - \theta)/2)$, whereas the probability of it being projected to $|-_\phi\rangle$ 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 SWAP test

A test which is widely used in quantum information is the *quantum* SWAP *test*. This is a simple procedure for determining whether two pure quantum states $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ are close to each other or far apart. We express closeness in terms of the absolute value of their inner product $|\langle\psi|\phi\rangle|$. The test involves preparing a qubit in the state $(|0\rangle + |1\rangle)/\sqrt{2}$ and performing a controlled-SWAP operation between that qubit and the state $|\psi\rangle|\phi\rangle$. SWAP is defined by the mapping $\mathsf{SWAP}|\psi\rangle|\phi\rangle = |\phi\rangle|\psi\rangle$, for any pair of states $|\psi\rangle$ and $|\phi\rangle$. The controlled-SWAP procedure leads to the following state:

$$\frac{|0\rangle|\psi\rangle|\phi\rangle + |1\rangle|\phi\rangle|\psi\rangle}{\sqrt{2}}$$

If one then applies a Hadamard operation to the first qubit and measures it in the computational basis it can be shown that the probability of obtaining outcome $|0\rangle$ is $(1 + |\langle\psi|\phi\rangle|^2)/2$. When the states are close, i.e. $|\langle\psi|\phi\rangle| \approx 1$, the SWAP test yields $|0\rangle$ with high probability. Conversely, when the states are far apart, i.e. $|\langle\psi|\phi\rangle| \approx 0$, the SWAP test yields $|0\rangle$ with probability close to $1/2$. The "probability gap" between the two cases can be made arbitrarily close to 1, by simply repeating the procedure many times (provided that multiple copies of $|\psi\rangle$ and $|\phi\rangle$ are available).

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, of which we give a simplified description.

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:

$$\mathcal{E}(G|\psi\rangle) = \sum_j E_j \; G|\psi\rangle\langle\psi| G^\dagger \; E_j^\dagger \tag{2.29}$$

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 [49].

To detect and correct for errors from the set $\{E_j\}_j$, one first performs an encoding procedure on $|\psi\rangle$ by 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 logical 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 *bit 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 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 $g_1 = \mathsf{Z} \otimes \mathsf{Z} \otimes I$ and $g_2 = I \otimes \mathsf{Z} \otimes \mathsf{Z}$[6]. 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 $|\psi\rangle \to |\phi\rangle = U\,|\psi\rangle$. Thus, a logical operation U_L would map $|\psi\rangle_L \to |\phi\rangle_L$. 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 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.

2.1.2 Measurement-based quantum computation (MBQC)

Measurement-based quantum computation (MBQC), defined in [50,51], is a model of quantum computation that is unlike the circuit model. In MBQC, a quantum computation is performed by doing successive measurements on qubits from a large entangled state. Typically, this state consists of qubits that have all been

[6] These are known as *stabilizer* operators for the states in the code spaces. We also encounter these operators in Subsection 2.1.2. The operators form a group under multiplication and so, when specifying the code space, it is sufficient to provide the generators of the group.

initialised as $|+\rangle$ and then entangled using the CZ (controlled-Z) operation, where:

$$\mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{2.30}$$

The qubits are then measured in the basis $(|+_\phi\rangle, |-_\phi\rangle)$. These measurements are denoted by $M(\phi)$, and depending on the values of ϕ for each qubit, one can perform universal quantum computation. For this to work, the entangled qubits need to form a so-called *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[7] 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.

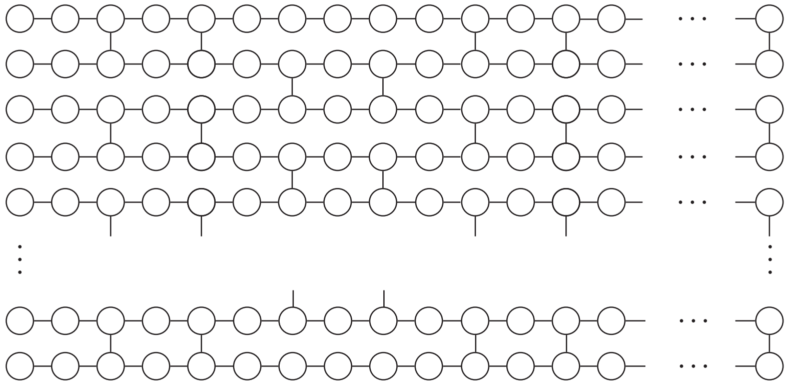


Figure 2.3: Brickwork state, reproduced from [52]

An example of a universal graph state is the *brickwork state*, defined in [52], which we illustrate in Figure 2.3 from that work. 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. **Initialisation**. 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.
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*.

[7] 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.

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 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 $O_1^{-1}|\psi\rangle = |\psi\rangle$ and $O_1 O_2 |\psi\rangle = |\psi\rangle$. 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:

$$K_v = \mathsf{X}_v \prod_{w \in N_G(v)} \mathsf{Z}_w \tag{2.31}$$

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 [52] for more details.

2.1.3 Self-testing

A concept which is of particular importance to entanglement-based verification protocols is that of *self-testing*. The idea of self-testing was introduced by Mayers and Yao in [53], 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[8]. 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 players' systems. This means that if the referee observes a 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 local isometries. More formally, we give the definition of self-testing, adapted from [54] and using notation similar to that of [36]:

Definition 1 (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. 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 responses of each player P_i are implemented through 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. In other words, when player i receives question j, his answer will be the outcome of measuring observable O_i^j on the state $|\psi\rangle$. We say that the game G self-tests the strategy S, with robustness $\epsilon = \epsilon(\delta)$, for some $\delta > 0$, if, for any strategy $\tilde{S} = (|\tilde{\psi}\rangle, \{\tilde{O}_i^j\}_{ij})$ 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:

$$TD(\Phi(|\tilde{\psi}\rangle), |junk\rangle |\psi\rangle) \leq \delta \tag{2.32}$$

and for all j:

$$TD\left(\Phi\left(\bigotimes_{i=1}^{n} \tilde{O}_i^j |\tilde{\psi}\rangle\right), |junk\rangle \bigotimes_{i=1}^{n} O_i^j |\psi\rangle\right) \leq \delta \tag{2.33}$$

To give some intuition, let us consider an example. Suppose we have two players, Alice and Bob. The game they will be playing is the *CHSH game* [55]. Alice and Bob will receive one of two possible questions. We denote Alice's input question as x and Bob's input as y, and we have that $x, y \in \{0, 1\}$. They will each provide a one bit output and we label Alice's output as a and Bob's output as b, with $a, b \in \{0, 1\}$. Alice and Bob win the game if an only if:

$$a \oplus b = x \cdot y \tag{2.34}$$

In other words, when the input bits are $x = y = 1$, Alice and Bob need to provide

[8] 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.

different responses ($a \neq b$). However, in all other cases, their responses must be identical ($a = b$). Assuming the referee chooses the inputs uniformly at random, it is not difficult to show that the optimal classical strategy for Alice and Bob achieves a success probability of 3/4 or 75%. In contrast to this, the optimal quantum strategy has a probability of success of $\omega^*(CHSH) = cos^2(\pi/8)$, or approximately 85.4% [56].

The optimal quantum strategy works as follows. First, Alice and Bob 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 $\mathsf{V} = (\mathsf{X} + \mathsf{Z})/\sqrt{2}$, on his half of the Bell pair, and on input $b = 1$, he measures $\mathsf{W} = (\mathsf{X} - \mathsf{Z})/\sqrt{2}$. Thus, in keeping with the notation of Definition 1, we have a strategy $S = (|\Phi_+\rangle_{AB}, \{\{\mathsf{X}, \mathsf{Z}\}_A, \{\mathsf{V}, \mathsf{W}\}_B\})$.

It was shown by McKague, Yang and Scarani that the CHSH game robustly self-tests the above strategy [32]. Specifically, they proved the following:

Theorem 1 (CHSH self-testing [32])**.** *Suppose Alice and Bob play the CHSH game with a referee that chooses the inputs x, y uniformly at random. Then, for any strategy $\tilde{S} = (|\tilde{\psi}\rangle_{AB}, \{\{\tilde{A}_1, \tilde{A}_2\}_A, \{\tilde{B}_1, \tilde{B}_2\}_B\})$ achieving winning probability $cos^2(\pi/8) - \epsilon$ there exists a local isometry $\Phi = \Phi_A \otimes \Phi_B$ and a state $|junk\rangle_{AB}$ such that:*

$$TD(\Phi(|\tilde{\psi}\rangle_{AB}), |junk\rangle_{AB} |\Phi_+\rangle_{AB}) \leq \delta \tag{2.35}$$

and for all $i, j \in \{1, 2\}$:

$$TD\left(\Phi\left(\tilde{A}_i \tilde{B}_j |\tilde{\psi}\rangle_{AB}\right), |junk\rangle_{AB} A_i B_j |\Phi_+\rangle_{AB}\right) \leq \delta \tag{2.36}$$

where $A_1 = \mathsf{X}$, $A_2 = \mathsf{Z}$, $B_1 = \mathsf{V}$, $B_2 = \mathsf{W}$ and $\delta = O(\epsilon^{1/4})$.

It should be noted that self-testing of Bell states, though not explicitly called that, had been considered in a number of works prior to that of McKague, Yang and Scarani [57–59].

It should also be noted that the 1/4 exponent in δ is not optimal. A tighter bound of $\delta = O(\sqrt{\epsilon})$ can be achieved, as is shown for instance in [31].

2.1.4 Quantum one-time pad

Suppose Alice wishes to send a classical message, M, to Bob, through a public channel, without anyone learning what the message is. Let us also suppose that Alice and Bob have a pre-shared key K, of the same length as M, denoted $n = |K| = |M|$. We also assume that this key is a uniformly random bit string, i.e. for any K, $Pr(K) = 2^{-n}$. What can Alice do? The answer is given by a construction known as the *one-time pad.* Alice will perform a bit-wise xor between the key and the message, obtaining a ciphertext C:

$$C = K \oplus M \tag{2.37}$$

She will send C, over the public channel, to Bob. Bob then xors his copy of K with C yielding:

$$M = C \oplus K \tag{2.38}$$

The reason he gets back M is because the xor operation is commutative and because $K \oplus K = 0$, for any K.

Suppose an eavesdropper, Eve, makes a copy of C in an attempt to learn M. What can she learn about M from C? To answer that, let us consider Eve's *prior* information about the message, before observing C, which is encoded in the probability distribution $Pr(M)$. We want to compare this distribution to the *posterior* distribution of Eve guessing M, after seeing C, i.e. $Pr(M|C)$. Using Bayes' theorem we have:

$$Pr(M|C) = \frac{Pr(C|M)Pr(M)}{Pr(C)} \tag{2.39}$$

But we know that $C = K \oplus M$ and that xor-ing with a given value is an injective operation, which means that $Pr(C|M) = Pr(K) = 2^{-n}$. Additionally, we have that:

$$Pr(C) = \sum_{M} Pr(C|M)Pr(M) = 2^{-n} \sum_{M} Pr(M) = 2^{-n} \tag{2.40}$$

This leads to:

$$Pr(M|C) = Pr(M) \tag{2.41}$$

In other words, seeing the string C has not changed Eve's information about the message M. The encrypted ciphertext reveals no information to Eve. Note that this argument did not rely on any sort of assumption about Eve's computational capabilities. For this reason, the one-time pad is said to be *information-theoretic secure* (or unconditionally secure).

We now consider the quantum case, where we have a quantum one-time pad for encrypting quantum information. Once again, we have two parties, Alice and Bob, but this time 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:

$$\frac{1}{4} \sum_{b_1,b_2 \in \{0,1\}} \mathsf{X}^{b_1}\mathsf{Z}^{b_2}\rho\mathsf{Z}^{b_2}\mathsf{X}^{b_1} \tag{2.42}$$

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

$$\frac{1}{4} \sum_{b_1,b_2 \in \{0,1\}} \mathsf{X}^{b_1}\mathsf{Z}^{b_2}\rho\mathsf{Z}^{b_2}\mathsf{X}^{b_1} = I/2 \tag{2.43}$$

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 Equation 2.43 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 2.43 generalises for the multi-qubit case so that for an n-qubit state ρ we have:

$$\frac{1}{4^n} \sum_{\mathbf{b_1},\mathbf{b_2} \in \{0,1\}^n} \mathsf{X}(\mathbf{b_1})\mathsf{Z}(\mathbf{b_2})\rho\mathsf{Z}(\mathbf{b_2})\mathsf{X}(\mathbf{b_1}) = I/2^n \tag{2.44}$$

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.

2.2 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 main complexity classes used in the thesis. We let $\{0,1\}^*$ denote the set of all finite binary strings 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 to the subject we refer the reader to [11, 60].

2.2.1 Decision classes

We start by describing decision classes. These are sets of languages, where a language is a subset of $\{0,1\}^*$. We assume familiarity with the classes P, of languages that can be decided in deterministic polynomial time and NP, of languages for which the "yes" instances can be verified in deterministic polynomial time. Let us now provide definitions for some of the main classes that we encounter throughout this thesis.

Definition 2. *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 $x \in L$, $M(x)$*[9] *accepts with probability at least c,*
- *when $x \notin L$, $M(x)$ accepts with probability at most s,*

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

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

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.

Before defining BQP we first define the concept of *polynomial-time uniform circuit family.*

Definition 3. *A family of (quantum) circuits $\{C_n\}_n$ is said to be polynomial-time uniform if there exists a deterministic Turing machine M and a polynomial p such that the running time of M on inputs of length n is bounded by $p(n)$ and $M(1^n)$ outputs a description of C_n.*

From now on, whenever we refer to "uniform circuits" we are implicitly speaking about a polynomial-time uniform circuit family.

Definition 4. *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 yielding 1 when measured in the computational basis. As mentioned in the introduction, it is known that BPP $\subseteq$ BQP, though it has not been proven that the containment is strict.

Definition 5. *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:*

- *when $x \in L$, there exists a string $w \in \{0,1\}^{p(n)}$, such that $V(x,w)$ accepts with probability at least c,*
- *when $x \notin L$, for all strings $w \in \{0,1\}^{p(n)}$, $V(x,w)$ 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 is tasked with checking a purported proof that $x \in L$, provided by the prover. A related class, which is briefly mentioned in Chapter 6 is AM. In AM the verifier flips a polynomial number of coins and sends the outcomes to the prover. The prover will then send a witness to the verifier, and based on this witness and the random bits the verifier will determine whether to accept or reject x. Assuming certain derandomization assumptions hold, it is the case that AM = MA = NP [11] (see also [61,62]).

We now define a quantum version of MA:

Definition 6. *A language $L \subseteq \{0,1\}^*$ belongs to* QMA *if there exists a polynomial p and a uniform quantum circuit family $\{V_n\}_n$, where each circuit has at most $p(n)$ gates, 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[10] since this will be referenced in some of the protocols. The specific problem we state was defined by Kitaev et al. and is known as the *k-local Hamiltonian problem* [63]. 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 reproduce the definition of the k-local Hamiltonian problem from [64]:

Definition 7 (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 $L \in \mathsf{QMA}$, 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 $|\psi\rangle$, 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

[10] A problem, P, is complete for the complexity class QMA if $P \in \mathsf{QMA}$ and all problems in QMA can be classically reduced in polynomial time to P.

least 5 for the problem to be QMA-complete. Subsequent work has shown that even with $k = 2$ the problem remains QMA-complete [65].

It should be noted that there is another quantum analogue of MA, called QCMA, for which we also provide the definition:

Definition 8. *A language $L \subseteq \{0,1\}^*$ belongs to* QCMA *if there exists a polynomial p and a uniform quantum circuit family $\{V_n\}_n$, where each circuit has at most $p(n)$ gates, taking x and a string w as inputs, such that for any $x \in \{0,1\}^n$ the following are true:*

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

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

Essentially, this is the same as QMA, except that the witness is a classical string, instead of a quantum state. Clearly MA $\subseteq$ QCMA $\subseteq$ QMA and it is believed that the containments are strict, though this is unproven.

We now move on to classes that characterise languages decided by interactive protocols. Let us start by defining an interactive protocol with classical messages. This definition is adapted from [60]:

Definition 9. *Let V be a probabilistic Turing machine whose running time on inputs of size n is bounded by $p(n)$, where p is a polynomial, and let $P : \{0,1\}^* \to \{0,1\}^*$ be a function from binary strings to binary strings. We will refer to the function P as a prover. A k-round (for even $k \geq 0$) interactive protocol between V and P, denoted $\langle V, P \rangle_k (x)$ is the sequence of strings m_1, m_2, ... $m_k \in \{0,1\}^{p(|x|)}$ defined as follows:*

$$\begin{gathered} m_1 = V(x) \\ m_2 = P(x, m_1) \\ m_3 = V(x, m_1, m_2) \\ \cdots \\ m_k = P(x, m_1, m_2, ...m_{k-1}) \end{gathered}$$

We say that the interactive protocol accepts x if $V(x, m_1, m_2, ...m_k)$ accepts and say that it rejects x otherwise.

One can analogously define an interactive protocol with quantum messages in which each message is a quantum state on $p(|x|)$ qubits and the verifier has the ability to receive and process quantum states.

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 $P : \{0,1\}^* \to \{0,1\}^*$ such that $\langle V, P \rangle_{p(n)} (x)$ accepts with probability at least c,*

- *when $x \notin L$, for any prover $P : \{0,1\}^* \to \{0,1\}^*$, $\langle V, P \rangle_{p(n)} (x)$ accepts with probability at most s,*

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

All previously defined complexity classes are contained in IP. In fact, a celebrated result of Shamir shows that IP is equivalent to the set of languages that can be decided using polynomial space on a deterministic Turing machine [66]. In other words, IP = PSPACE.

While the previous are fairly standard complexity classes, we now state the definition of a more non-standard class, which first appeared in [23]:

Definition 11. *A language $L \subseteq \{0,1\}^*$ belongs to* QPIP *if there exist polynomials 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 where the prover can be any function from binary strings to binary strings (and may not even be a computable function), in QPIP the prover is restricted to BQP computations. This constraint on the prover has the direct implication that QPIP $\subseteq$ BQP. As we will see, the Fitzsimons-Kashefi protocol and the Morimae-Fitzsimons protocol, which we detail in Subsection 2.4, allow a BPP verifier with the ability to prepare or measure single qubits to delegate and verify arbitrary BQP computations to a quantum prover. This, in effect, shows that BQP $\subseteq$ QPIP. Hence QPIP = BQP.

We now proceed to the multi-prover setting and define the multi-prover generalisation 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 exchange at most $p(n)$ messages (of length at most $p(n)$) with V, are not allowed to communicate, 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 exchange at most $p(n)$ messages (of length at most $p(n)$) with V and are not allowed to communicate, can make V accept with probability at most s,*

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

Note that $\mathsf{MIP[1]} = \mathsf{IP}$ and it was shown that for all $k > 2$, $\mathsf{MIP[k]} = \mathsf{MIP[2]}$ [67]. The latter class is simply denoted MIP. If the provers are allowed to share entanglement then we obtain the class $\mathsf{MIP^*[k]}$:

Definition 13. *A language $L \subseteq \{0, 1\}^*$ belongs to $\mathsf{MIP^*[k]}$ 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, 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 $\mathsf{MIP^*[k]} = \mathsf{MIP^*[2]}$, for $k \geq 2$, and this class is denoted as $\mathsf{MIP^*}$ [68]. It is not known whether $\mathsf{MIP} = \mathsf{MIP^*}$, however, it is known that both classes contain BQP. If the provers are restricted to BQP computations, then the resulting class is, in fact, equal to BQP [31]. This is demonstrated by the Reichardt-Unger-Vazirani protocol, which we describe in Subsection 2.4.

Note that while the protocols we will be presenting 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 protocol of the QPIP type, from the complexity theoretic perspective, we know that we have a BPP verifier with a quantum device for preparing or measuring a constant number of qubits and that this verifier can delegate arbitrary BQP decision problems to the prover by interacting with it for a polynomial number of rounds. However, we will also be interested in other characteristics of such a protocol, for instance:

- 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.

An important category of complexity classes, that we will use in Chapter 6, is that of *advice classes*. Let us provide a definition of this concept:

Definition 14. *Let $\mathcal{C}$ be a complexity class and $\mathcal{F}$ a family of functions $f : \mathbb{N} \to \{0,1\}^*$. The complexity class $\mathcal{C}/\mathcal{F}$, known as $\mathcal{C}$ with $\mathcal{F}$ advice, is the set of all languages L, for which there exists an $L' \in \mathcal{C}$ and a function $f \in \mathcal{F}$ such that for all $x \in \{0,1\}^*$, $x \in L$ iff. $\langle x, f(|x|)\rangle \in L'$.*

As an example, consider the class $\mathsf{P/poly}$. This consists of all languages that can be decided by a polynomial-time deterministic Turing machine, that receives polynomially-many bits of advice for all inputs of the same length. In other words, for all inputs $x \in \{0,1\}^n$, the Turing machine also receives some string $a \in \{0,1\}^{poly(n)}$, aiding it in deciding whether to accept x or not. Analogously $\mathsf{NP/poly}$ consists of languages that can be decided by a polynomial-time *non-deterministic* Turing machine, that receives polynomially-many bits of advice, for inputs of the same length. In Chapter 6 we will also encounter the class $\mathsf{NP/O(n^d)}$ in which the size of the advice string is $O(n^d)$, for some fixed constant d.

We will also use complexity classes with advice that are not covered by Definition 14. For instance, the class $\mathsf{BPP/rpoly}$ denotes the set of languages that can be decided by a BPP machine that receives *randomized* polynomial-size advice. In other words, for all inputs $x \in \{0,1\}^n$, the probabilistic Turing machine (or algorithm) also receives some string $a \in \{0,1\}^{poly(n)}$, that is drawn from a distribution $\mathcal{D}_n$. We can see that this does not satisfy Definition 14 since the advice string is not the result of some deterministic function, but is a sample from a probability distribution. It is therefore to be understood that rpoly will correspond to polynomial-size advice drawn from a probability distribution that only depends on the size of the input.

Another possibility, which we will encounter in Chapter 6 is that of quantum advice. As an example, the class $\mathsf{BQP/qpoly}$ denotes the set of languages that can be decided by a BQP machine that receives as advice a quantum state of polynomially-many qubits. In other words, for all inputs $x \in \{0,1\}^n$, the quantum Turing machine (or quantum circuit) also receives a quantum state $|\psi_n\rangle \in \mathcal{H}_n$, such that $dim(\mathcal{H}_n) = 2^{poly(n)}$. It is therefore to be understood that qpoly will correspond to polynomial-size quantum advice and represents a quantum state of polynomially-many qubits that only depends on the size of the input.

The concept of oracles will be encountered in Chapter 6, and so we give a brief description of the subject. An oracle is a black box function that can be invoked by a Turing machine in order to obtain the solution to some problem in one time step. For example, the class of problems which can be solved by a deterministic polynomial-time Turing machine with access to some oracle function $O : \{0,1\}^* \to \{0,1\}$ is denoted P^O. If O is an oracle for some NP-complete problem, then the corresponding class is P^{NP}. For quantum classes, oracles are viewed as unitaries performing mappings such as $U_O |x\rangle |y\rangle = |x\rangle |y \oplus O(x)\rangle$, where O is the oracle function. Additionally, whenever a result involving complexity classes remains true when those classes are given access to an oracle, O, we say that the result *relativises.*

Another important concept is that of the complement of a complexity class:

Definition 15. *Let $\mathcal{C}$ be a complexity class. The class $\mathsf{co}\mathcal{C}$ is the complement of $\mathcal{C}$ and consists of all languages $L^c = \{0,1\}^* \setminus L$, where $L \in \mathcal{C}$.*

Note that L^c is referred to as the complement of the language L. It should also be noted that P, BPP and BQP are closed under complement, i.e. $\mathsf{P} = \mathsf{coP}$, $\mathsf{BPP} = \mathsf{coBPP}$, $\mathsf{BQP} = \mathsf{coBQP}$. The same is not known to be true for NP or the Merlin-Arthur classes (MA, QCMA, QMA). Having the notion of the complement, we can now define the *polynomial hierarchy*:

Definition 16. *Let $\mathsf{\Sigma}_0^{\mathsf{P}} = \mathsf{\Pi}_0^{\mathsf{P}} = \mathsf{P}$, denote the zeroth level of the polynomial hierarchy. Additionally, for some $k > 0$, let $\mathsf{\Sigma}_k^{\mathsf{P}} = \mathsf{NP}^{\mathsf{\Sigma}_{k-1}^{\mathsf{P}}}$, $\mathsf{\Pi}_k^{\mathsf{P}} = \mathsf{coNP}^{\mathsf{\Sigma}_{k-1}^{\mathsf{P}}}$ denote the k'th level of the polynomial hierarchy. Finally, the polynomial hierarchy is denoted PH and is defined as $\mathsf{PH} = \mathsf{\Sigma}_0^{\mathsf{P}} \cup \mathsf{\Sigma}_1^{\mathsf{P}} \cup \mathsf{\Sigma}_2^{\mathsf{P}} \cup ....$*

It is clear that each level of the polynomial hierarchy is contained in the one above it and it is conjectured that all of these containments are strict. This condition is typically expressed by saying "the polynomial hierarchy is infinite" or "the polynomial hierarchy does not collapse". A collapse of the polynomial hierarchy at level k means that for all $j > k$, $\mathsf{\Sigma}_k^{\mathsf{P}} = \mathsf{\Sigma}_j^{\mathsf{P}}$ (and $\mathsf{\Pi}_k^{\mathsf{P}} = \mathsf{\Pi}_j^{\mathsf{P}}$).

The final complexity class we define is $\#\mathsf{P}$, which is a *counting class* (a class of functions that output natural numbers) and not a decision class.

Definition 17. *A function $f : \{0,1\}^* \to \mathbb{N}$ belongs to the class $\#\mathsf{P}$ if there exists a polynomial p, and a language $L \in \mathsf{P}$ such that for every $x \in \{0,1\}^*$:*

$$f(x) = \left| \{ w \in \{0,1\}^{p(|x|)} | \langle x, w \rangle \in L \} \right| \tag{2.45}$$

While $\#\mathsf{P}$ itself is not a decision class, we can easily create a decision class that captures the power of $\#\mathsf{P}$ by considering $\mathsf{P}^{\#\mathsf{P}}$. A celebrated result by Toda shows that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$ [69].

2.2.2 Sampling classes and BosonSampling

In contrast to decision classes, where one is only interested in providing a yes/no answer to a problem, sampling classes involve problems requiring one to obtain a sample from a particular probability distribution. We provide the definition, from [70], for the classes of sampling problems that can be solved efficiently by polynomial-time classical and quantum algorithms, respectively:

Definition 18. *A sampling problem S is a collection of probability distributions $(\mathcal{D}_x)_{x\in\{0,1\}^*}$, one for each input string $x \in \{0,1\}^n$, where $\mathcal{D}_x$ is a distribution over $\{0,1\}^{p(n)}$, for some fixed polynomial p. Then* SampBPP *is the class of sampling problems $S = (\mathcal{D}_x)_{x\in\{0,1\}^*}$ for which there exists a probabilistic polynomial-time algorithm B that, given $\langle x, 0^{\lceil 1/\varepsilon \rceil} \rangle$ as input, samples from a probability distribution $\mathcal{C}_x$ such that $D(\mathcal{C}_x, \mathcal{D}_x) \leq \varepsilon$, where D denotes total variation distance.* SampBQP *is defined the same way, except that B is a quantum algorithm rather than a classical one.*

Similar definitions can be given for SampMA, SampQMA and so on. In other words, for any decision class involving randomness, one can define the corresponding sampling class. For this reason, we will occasionally abuse the notation and say that a particular sampling problem is contained in, say, BPP, rather than SampBPP.

A specific sampling problem, contained in SampBQP, that we will encounter in Chapter 6 is BosonSampling. This problem was defined by Aaronson and Arkhipov in [71] and used to show that if the problem were also contained in SampBPP, and provided certain assumptions about the permanents[11] of Gaussian matrices are true, then the polynomial hierarchy collapses at the third level. We refer the reader to [71] for a more detailed description of the problem as well as the result of Aaronson and Arkhipov. It should be noted that Definition 18 considers the case of *approximate sampling*, i.e. the distribution that is sampled from should be close in variation distance to the target distribution. If one is interested in *exact sampling* (sampling from the target distribution), for BosonSampling, then it is relatively easy to show that an exact BPP sampler would lead to a collapse of the polynomial hierarchy. Let us give a description of this argument.

First of all, we start by describing BosonSampling. In BosonSampling, identical photons (bosons) are sent through a linear optics network and non-adaptive measurements are performed to count the number of photons in each mode. For a system with m modes and n photons, the basis states of the system are of the form $S = (s_1, ...s_m)$, where s_i denotes the number of photons in mode i (so $s_1 + ... + s_m = n$). A general state, is then a state of the form:

$$|\psi\rangle = \sum_S \alpha_S |S\rangle , \quad \sum_S |\alpha_S|^2 = 1 \tag{2.46}$$

Note that the number of basis states is $M = \binom{m+n-1}{n}$. The action of the linear optics network can be expressed as a matrix $A \in \mathcal{U}_{m,n}$, where $\mathcal{U}_{m,n}$ is the set of all $m \times n$ column-orthonormal matrices. Let A_S be the matrix obtained by taking s_i copies of the i'th row of A, for all $i \leq m$. If the initial state of the system consists of one photon in each of the first n modes (it is assumed that $m \geq n$), then it can be shown that the probability of observing the state S, upon passing the photons through the network described by A and measuring the number of

[11] The permanent of a matrix $M = (m_{ij})_{i,j\leq n}$, is defined as $Per(M) = \sum_{\sigma\in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$, with S_n the symmetric group of all permutations of the elements 1 up to n.

photons in each mode, will be:

$$Pr(S) = \frac{|Per(A_S)|^2}{s_1!s_2!...s_m!} \tag{2.47}$$

BosonSampling is then the problem of sampling from the distribution defined by Equation 2.47.

To explain why BosonSampling is believed to be hard for classical computers, we first need to state a result known as *Stockmeyer's approximate counting method* [72]. This says that given an efficiently computable[12] function $f : \{0,1\}^n \to \{0,1\}$, there is a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm for giving a multiplicative estimate[13] of:

$$p = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) \tag{2.48}$$

Now, suppose there existed a BPP algorithm that, given A as input, could sample from the distribution of Equation 2.47. This algorithm can be viewed as a deterministic polynomial-time computable function F that, given A and a string $r \in \{0,1\}^{p(n)}$, for some polynomial p, produces a vector $S = (s_1, ...s_m)$ (of the form described above). The fact that this algorithm can sample from the BosonSampling distribution can be expressed mathematically as:

$$\Pr_{r \leftarrow_R \{0,1\}^{p(n)}} (F(A,r) = S) = \frac{|Per(A_S)|^2}{s_1!s_2!...s_m!} \tag{2.49}$$

where $r \leftarrow_R \{0,1\}^{p(n)}$ denotes the fact that r was drawn uniformly at random from the set $\{0,1\}^{p(n)}$. Consider now a vector, which we denote as $S_{|1\rangle}$ in which all s_i are either 0 or 1 (of course, it should still be the case that $s_1 + ... + s_m = n$). Note that $Pr(S_{|1\rangle}) = |Per(A_{S_{|1\rangle}})|^2$. We will define a function f as follows:

$$f(A,r) = \begin{cases} 0, & \text{if } F(A,r) \neq S_{|1\rangle} \\ 1, & \text{if } F(A,r) = S_{|1\rangle} \end{cases} \tag{2.50}$$

Note that f is computable in polynomial time (since it simply involves evaluating F and testing whether the output is $S_{|1\rangle}$). The probability that the BPP algorithm produces the output $S_{|1\rangle}$ can then be expressed as:

$$\Pr_{r \leftarrow_R \{0,1\}^{p(n)}} \left(F(A,r) = S_{|1\rangle}\right) = \frac{1}{2^{p(n)}} \sum_{r \in \{0,1\}^{p(n)}} f(A,r) \tag{2.51}$$

But this sum can be estimated, up to multiplicative error, in $\mathsf{BPP}^{\mathsf{NP}}$ using Stockmeyer's method. In other words, there is a $\mathsf{BPP}^{\mathsf{NP}}$ algorithm for estimating $|Per(A_{S_{|1\rangle}})|^2$. It is shown in [71] that one can consider any matrix having entries from the set $\{-1,0,1\}$ and embed it in A (with only an added polynomial over-

[12] In other words, there is a polynomial-time classical algorithm for computing f.

[13] A multiplicative estimate of some value p is a number $\tilde{p}$ such that there exists a $g \geq 1$ and $p/g \leq \tilde{p} \leq gp$. For the p of Equation 2.48, the $\mathsf{BPP}^{\mathsf{NP}}$ algorithm produces an estimate with $g = 1 + 1/poly(n)$ [71,72].

head) so that the probability of sampling the $S_{|1\rangle}$ vector is the squared permanent of this matrix. By the above argument, this means that computing a multiplicative estimate for the squared permanent of a matrix over $\{-1, 0, 1\}$ is in $\mathsf{BPP}^{\mathsf{NP}}$. However, computing such an estimate is #P-hard [71]. Since it is known that $\mathsf{BPP}^{\mathsf{NP}}$ is contained in the third level of the polynomial hierarchy [73], by Toda's theorem, the existence of such an algorithm would lead to a collapse of PH at that level. Such a collapse is regarded as unlikely and therefore the existence of an efficient classical algorithm for BosonSampling is also considered unlikely.

2.3 Universal blind quantum computation (UBQC)

The concept of blind computing is highly relevant to quantum verification. Here, we simply give a succinct description of a particular protocol, known as universal blind quantum computation (UBQC). For more details, see this review of blind quantum computing protocols by Fitzsimons [74] as well as [52, 75–78].

Blindness is related to the idea of *computing on encrypted data* [79]. 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 $y = Enc(x)$. 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, $f(x) = Dec(z)$, where Dec is a decryption procedure that can be performed efficiently by the client[14]. The encryption procedure can, roughly speaking, 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, as we have seen with the (quantum) one-time pad, guarantees that the protocol is secure even against a server of unbounded computational power. See [84] 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 [75]. His pro-

[14] 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 [80–83]. 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.

tocol 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 the UBQC protocol of Broadbent, Fitzsimons and Kashefi [52], that we present here. UBQC maintained information-theoretic security but reduced the quantum communication to a single message from the client to the server. The protocol 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 [52]:

Definition 19 (Blindness)**.** *Let P be a quantum delegated computation protocol, between a client and server, on input X and let $L(X)$ be any function of the input. We say that P is blind while leaking at most $L(X)$ if, on the client's input X, for any fixed $Y = L(X)$, the following two hold when given Y:*

1. *The distribution of the classical information obtained by the server in P is independent of X.*
2. *Given the classical information described in 1 (i.e. given a sample from the distribution of classical information), the state of the quantum system obtained by the server in P is fixed and independent of X.*

For $L(X) = |X|$, UBQC satisfies this definition of blindness [52]. What this means is 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. We now give the description of UBQC.

We will refer to the client as Alice and the server as Bob. Alice is restricted to BPP computations but has the ability to prepare single-qubit states. Bob, on the other hand is a full fledged quantum computer, with the ability to perform BQP computations. Alice wishes to delegate to Bob the application of some quantum circuit $\mathcal{C}$ on a particular input, $|x\rangle$, where x is a classical bit string[15]. UBQC will allow Alice to do this while keeping both x and $\mathcal{C}$ hidden from Bob, in the sense of Definition 19.

We will view $\mathcal{C}$ as an MBQC computation. By considering some universal graph state, $|G\rangle$, such as the brickwork state (see Figure 2.3), Alice can convert $\mathcal{C} |x\rangle$ 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 states, the graph G would be the same for all circuits $\mathcal{C}'$ having the same number of gates as $\mathcal{C}$ (and acting on inputs of the same size as x). Hence, if she were to send this description to Bob, it would not reveal to him the circuit $\mathcal{C}$, nor

[15] In UBQC, Alice can also provide a quantum state as input, however, for our purposes we are only interested in a classical input.

the input x, but merely an upper bound on their size. It is, in fact, the measurement angles and the ordering of the measurements (known as *flow*) that uniquely characterise $\mathcal{C}\,|x\rangle$ [85]. However, 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 Subsection 2.1, 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[16]. 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 random, irrespective of the values of ϕ_i. He would, however, learn the measurement outcomes of the MBQC computation. 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 the 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 $\{0, 1\}$.

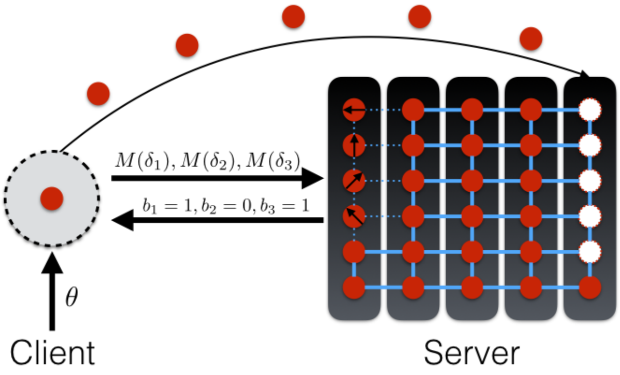

$M(\delta_1), M(\delta_2), M(\delta_3)$
$b_1 = 1, b_2 = 0, b_3 = 1$
θ
Client
Server

Figure 2.4: Universal Blind Quantum Computation

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 $\mathcal{C}\,|x\rangle$ into a pair $(G, \{\phi_i\}_i)$ and sends G to Bob, 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 $\{0, 1\}$.

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

(4) She then prepares the states $|+_{\theta_i}\rangle$ and sends them to Bob, who is instructed to entangle them, using CZ operations, according to the graph structure of 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 Subsection 2.1.2.

(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. This is done by xor-ing r_i with the corrected measurement outcome of qubit i.

The protocol is illustrated schematically in Figure 2.4, reproduced from [45] (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 by Dunjko and Kashefi shows that Alice can utilise any two non-overlapping, non-orthogonal states in order to perform UBQC, as opposed to using $|+_\theta\rangle$ states [86]. Of course, this is merely a sketch argument for how blindness works and for a complete proof we refer the reader to the following [52, 86, 87].

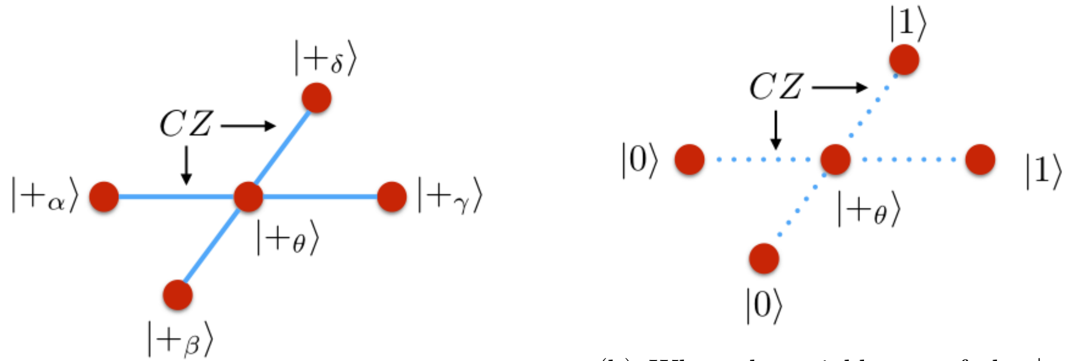
2.4 Verification of quantum computation

In this section, we will outline three protocols for verifying quantum computations. We will be referencing these protocols particularly in Chapters 3, 4 and 5. Each of these protocols is representative for a family of verification techniques. The first protocol we describe is the Fitzsimons and Kashefi protocol, which is representative of prepare-and-send protocols. The second protocol is the one of Reichardt, Unger and Vazirani, which is representative of entanglement-based protocols. Lastly, we outline the post hoc verification protocol of Morimae and Fitzsimons, which is representative of receive-and-measure protocols.

2.4.1 Fitzsimons-Kashefi (FK) protocol

In this subsection we discuss the prepare-and-send protocol developed by Fitzsimons and Kashefi in [24], which we shall refer to as the FK protocol. 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 Subsection 2.3. 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.

Let us 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 $|G\rangle$. An illustration is provided in Figure 2.5.



(a) When the neighbours of the $|+_\theta\rangle$ state are XY-plane states, the state becomes entangled with them upon the application of the CZ operations.

(b) When the neighbours of the $|+_\theta\rangle$ state are computational basis states, the state does not become entangled with them upon the application of the CZ operations.

Figure 2.5

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[17] 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.

Traps serve two purposes: one is to ensure that the server is performing the correct measurements; the second is to ensure that the server entangles the states according to the graph structure G. The second property might not seem

[17] 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 Subsection 2.1.2 allows for only one trap qubit to be embedded, whereas other graph states allow for multiple traps. See [24,88] for more details.

obvious at first sight. The reason the traps check the entanglement structure is because, while traps themselves are not entangled to the rest of the qubits in the graph, the CZ operation will make them classically correlated with their dummy neighbours. Specifically, a CZ between a $|1\rangle$ state and a $|+_\theta\rangle$ qubit has the effect of mapping this latter qubit to $|+_{\theta+\pi}\rangle = |-_\theta\rangle$. The π rotation, as before, flips the measurement outcome for this qubit. Therefore, the verifier can check whether the outcome for a particular trap has been flipped, in accordance with its local neighbours. Having many such local checks, allows the verifier to effectively check the global entanglement structure. Thus, by using the traps (and the dummies), the verifier can check both the prover's measurements and his entangling operations and hence verify the MBQC computation.

It should be noted, that checking for the correlation between the traps and the neighbouring dummies can be done in two ways. Either the verifier adjusts the measurement angles for the trap qubits accordingly, or the θ angles for the traps (as well as any other qubit that is connected to a dummy) are chosen as a function of the neighbouring dummies. FK uses this latter approach. Specifically, if for qubit i, one chooses the preparation angle θ_i, the actual angle that will be used is $\theta_i + \sum_{j \in N^D_G(i)} d_j \pi$, where $N^D_G(i)$ denotes the set of dummy neighbours of qubit i in $|G\rangle$, and d_j is the value of dummy qubit j.

We are now ready to present the steps of the FK protocol:

(1) The verifier chooses an input x and a quantum computation $\mathcal{C}$ that she would like the prover to perform on $|x\rangle$[18].

(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[19].

(3) Alice picks, uniformly at random, values θ_i, with i ranging 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 $|+_{\theta_i}\rangle$, 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. Note that here we are slightly abusing notation since, as mentioned, the actual preparation angle for qubit i will be $\theta_i + \sum_{j \in N^D_G(i)} d_j \pi$.

(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

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

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

that incorporates corrections resulting from previous measurements; trap qubits will 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.

(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.

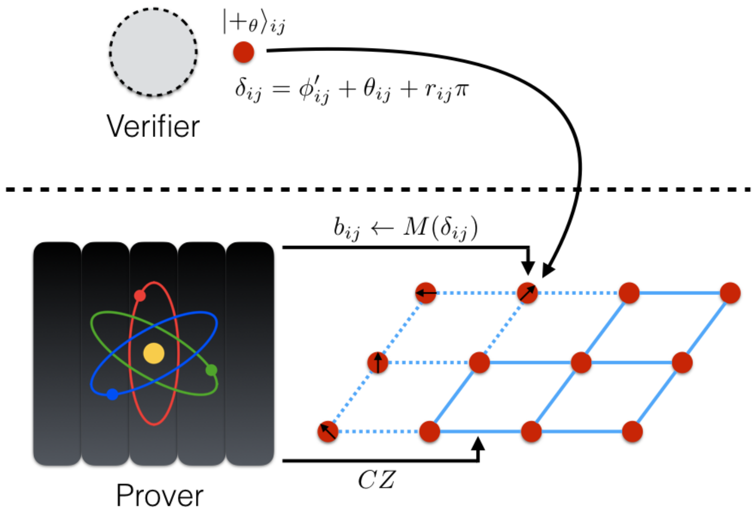

$|+_\theta\rangle_{ij}$
$\delta_{ij} = \phi'_{ij} + \theta_{ij} + r_{ij}\pi$
Verifier
$b_{ij} \leftarrow M(\delta_{ij})$
Prover
CZ

Figure 2.6: The FK protocol for verifiable universal blind quantum computing. The ij subscripts index the positions of the qubits in a rectangular graph state.

The protocol is illustrated schematically in Figure 2.6.

Notice that when Bob is being honest, i.e. following the instructions of the protocol (and of Alice), the verifier will accept the correct output. Or, more precisely, she will be sampling from the correct probability distribution, obtained by measuring $\mathcal{C}|x\rangle$ in the computational basis.

What about the case when Bob is dishonest and trying to deceive Alice? One can upper bound the probability that Alice accepts *and* the outcome is incorrect. We will sketch the proof of this, from [24], for a simplified version of the protocol in which Alice introduces a single trap qubit, ($T = 1$) at a uniformly random position in $|G\rangle$, denoted $|+_{\theta_t}\rangle$. First of all, what does it mean for the outcome to be incorrect? Fitzsimons and Kashefi define it as follows. If the prover is honest and follows the instructions of the protocol, then, the result of the MBQC computation should be $\mathcal{C}|x\rangle$ (prior to measuring this result). An incorrect output is then defined as a state in the complementary subspace, defined by the projector $I - \mathcal{C}|x\rangle\langle x|\mathcal{C}^\dagger$. As an extension of this, they also define:

$$P^{\nu}_{incorrect} = (I - \mathcal{C}|x\rangle\langle x|\mathcal{C}^\dagger) \otimes |+^{\nu}_{\theta_t}\rangle\langle +^{\nu}_{\theta_t}| \tag{2.52}$$

as the projection of the output to the incorrect subspace, but with the trap projected on the state that leads the verifier to acceptance. The superscript 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 $\{r_i\}_i$ (in this case, there is only one trap qubit and the only dummies are the neighbours of that trap). The dependence on ν, for the trap, encodes the position of the trap in the graph state as well as the fact that the state could have been flipped depending on the states of its neighbouring dummies.

Continuing with the notation from [24], we let:

$$\mathcal{B}_j(\nu) = \sum_{\mathbf{s}} p_{\nu,j}(\mathbf{s}) \left|\mathbf{s}\right\rangle \left\langle\mathbf{s}\right| \otimes \rho^{\mathbf{s}}_{\nu,j} \tag{2.53}$$

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, performed by the prover (which corresponds to measuring the output of the computation). The subscript j ranges over the possible strategies of the prover[20] with $j = 0$ corresponding to the 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). Note that $\mathbf{s}$ and $\rho^{\mathbf{s}}_{\nu,j}$ are classically correlated, since $\mathbf{s}$ is a result of the classical transcript of interaction between Alice and Bob.

For the probability of Alice accepting an incorrect outcome to be upper bounded, one needs to find an $\epsilon > 0$ such that:

$$Tr\left(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} \mathcal{B}_j(\nu)\right) \leq \epsilon \tag{2.54}$$

The proof entails 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 Bob 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:

$$\mathcal{B}_j(\nu) = \mathcal{E}_j(\mathcal{B}_0(\nu)) \tag{2.55}$$

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

[20] 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.

decomposed into Pauli operators leading to:

$$\mathcal{B}_j(\nu) = \sum_{k,l,m} \alpha_{kl}(j)\alpha^*_{km}(j) \; P_l \mathcal{B}_0(\nu) P_m \tag{2.56}$$

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

$$\mathcal{B}_j(\nu) = \beta \mathcal{B}_0(\nu) + (1-\beta) \sum_{k,l,m} \alpha_{kl}(j)\alpha^*_{km}(j) \; P_l \mathcal{B}_0(\nu) P_m \tag{2.57}$$

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

$$\begin{aligned} Tr\left(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} \mathcal{B}_j(\nu) \right) &= \\ (1-\beta) Tr &\left(\sum_{\nu} \sum_{k,l,m} p(\nu) P^{\nu}_{incorrect} (\alpha_{kl}(j)\alpha^*_{km}(j) \; P_l \mathcal{B}_0(\nu) P_m) \right) \end{aligned} \tag{2.58}$$

We now use the fact that $P^{\nu}_{incorrect} = (I - \mathcal{C} \left|x\right\rangle \left\langle x\right| \mathcal{C}^{\dagger}) \otimes \left|+^{\nu}_{\theta_t}\right\rangle \left\langle +^{\nu}_{\theta_t}\right|$ 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:

$$\begin{aligned} Tr\left(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} \mathcal{B}_j(\nu) \right) &\leq \\ Tr &\left(\sum_{\nu} p(\nu) \left|+^{\nu}_{\theta_t}\right\rangle \left\langle +^{\nu}_{\theta_t}\right| \sum_{k,l,m} \alpha_{kl}(j)\alpha^*_{km}(j) \; P_l \mathcal{B}_0(\nu) P_m \right) \end{aligned} \tag{2.59}$$

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 Bob (a fact which is essentially ensured by UBQC). The above expression then becomes:

$$Tr\left(\sum_{\nu^t} p(\nu^t) \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \sum_{k,l,m} \alpha_{kl}(j)\alpha^*_{km}(j) \; P_l (\left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \otimes (I/Tr(I))) P_m \right) \tag{2.60}$$

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 Paulis (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. What about the trap system?

For that we will have:

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

$$\sum_{\nu^t} p(\nu^t) \left\langle +^{\nu^t}_{\theta_t}\right| P_l \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| P_m \left|+^{\nu^t}_{\theta_t}\right\rangle \quad (2.61)$$

But now recall that ν^t consists of three things: θ_t, r_t and the position of the trap. Additionally, we're taking $p(\nu^t)$ to be the uniform distribution over these parameters. By summing over the first two parameters, the above expression becomes zero, whenever $l \neq m$. This is known as Pauli twirling [15, 25]. Thus, only terms in which $l = m$ will remain. Substituting this back into expression 2.60 leads to:

$$Tr\left(\sum_{\nu^t} p(\nu^t) \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \sum_{k,l} |\alpha_{kl}(j)|^2 \; P_l(\left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \otimes (I/Tr(I)))P_l\right) \quad (2.62)$$

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:

$$Tr\left(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} \mathcal{B}_j(\nu)\right) \leq \frac{N-1}{N} = 1 - \frac{1}{N} \quad (2.63)$$

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 $\epsilon = (8/9)^d$ can be achieved [88]. Alternatively, if the input state and computation are encoded in an error correcting code of Pauli-weight[21] 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 [24, 89, 90] for more details. An important observation, however, is that the fault tolerant encoding, is used *only to boost security* and not for correcting deviations arising from faulty devices.

[21] The Pauli-weight of an error correcting code represents the maximum number of non-identity Pauli operations that the code can correct for.

For a more detailed derivation of the above bound, see [24, 91].

2.4.2 Reichardt-Unger-Vazirani (RUV) protocol

In this subsection we describe the entanglement-based protocol of Reichardt, Unger and Vazirani, which we shall refer to as the RUV protocol. As we saw in Subsection 2.1.3, the CHSH game is an example of a two-player non-local game in which a quantum strategy for playing the game outperforms any classical strategy. The quantum strategy achieves a success probability of $cos^2(\pi/8) \approx 85.4\%$. It was shown by Tsirelson that this strategy is optimal for quantum mechanics [56], which led to the development of robust self-tests based on the CHSH game [32,57]. More specifically, these results show that if one observes two players winning the CHSH game with a near $cos^2(\pi/8)$ probability, it can be concluded that the players' shared state is close to a Bell pair and that 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).

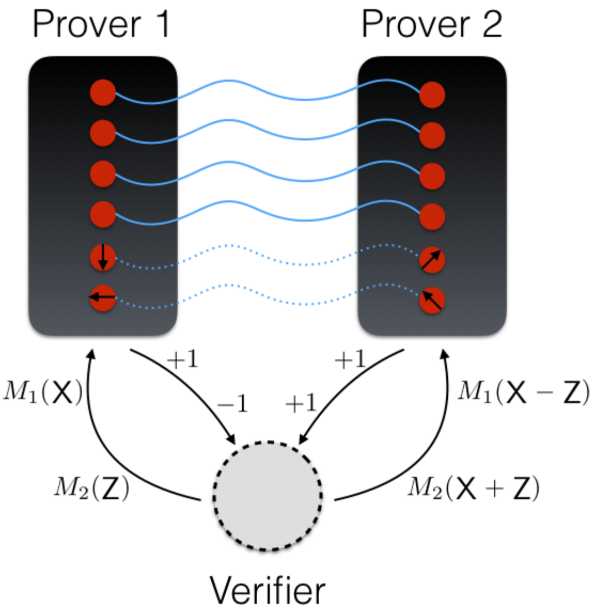


Figure 2.7: Ideal CHSH game strategy

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 [31][22]. It is this latter result that is relevant for the RUV protocol so we give a more formal statement:

[22] Note that the Summers and Werner and McKague, Yang and Scarani results 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.

Theorem 2. *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 $S = (|\tilde{\psi}\rangle, \tilde{A}(\mathbf{a}), \tilde{B}(\mathbf{b}))$ be the strategy employed by Alice and Bob in playing the n CHSH games, where $|\tilde{\psi}\rangle$ 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 exists a local isometry $\Phi = \Phi_A \otimes \Phi_B$ and a state $|junk\rangle$ such that:

$$TD(\Phi(|\tilde{\psi}\rangle), |junk\rangle |\Phi_+\rangle^{\otimes n}) \leq \delta \tag{2.64}$$

and:

$$TD\left(\Phi\left(\tilde{A}(\mathbf{a}) \otimes \tilde{B}(\mathbf{b}) |\tilde{\psi}\rangle\right), |junk\rangle A(\mathbf{a}) \otimes B(\mathbf{b}) |\Phi_+\rangle^{\otimes n}\right) \leq \delta \tag{2.65}$$

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.

Before giving the description of the protocol, we first give a succinct introduction to the concept of *gate teleportation*, first defined in [92]. 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 $(I \otimes U)|\Phi_+\rangle$. 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:

$$|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad |\Phi_-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$|\Psi_+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \qquad |\Psi_-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

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. While this example involved a one-qubit gate, it is simple to extend the primitive to two-qubit gates, by simply considering two Bell pairs. If one were to repeat this procedure with gates from a universal gate set, then one would be able to perform universal quantum computations. In that sense, as is explained in [92], gate teleportation is a universal computational primitive.

We now describe the RUV protocol. It uses a classical verifier and two quantum provers, but can be generalised to any number of provers greater than two. Suppose that Alice and Bob are the two provers[23]. They are allowed to share an

[23] Note that here we are breaking from the convention of having Alice as the verifier and Bob

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 $\mathcal{C}$, acting on some input $|x\rangle$. The protocol consists of 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 Figure 2.7.

- **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 index the qubits upon which 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

as the prover. The Alice and Bob convention is used here in a similar fashion to how it is used for the CHSH game. The two parties that are sharing entanglement (in this case, the provers) are the ones traditionally known as Alice and Bob.

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, possess that property as well. This follows from a more involved argument regarding the way in which, as a by-product of the teleportation, the state is kept one time padded.

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.

2.4.3 Morimae-Fitzsimons (MF) protocol

In this subsection, we present the post hoc verification protocol of Morimae and Fitzsimons, which we shall refer to as the MF protocol [28, 93]. We note that the same protocol was independently proposed in [94] by Hangleiter et al.

The starting point is the complexity class QMA, for which we have stated the definition in Subsection 2.2. 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 7. We mentioned that for $k = 2$ the problem is QMA-complete. 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 $w_X(S_i) + w_Z(S_i) \leq 2$. Essentially, as the name suggests, an XZ-Hamiltonian is one in which each local term consists of two terms, each of which can be either X or Z.

The MF protocol starts with the observation that $\mathsf{BQP} \subseteq \mathsf{QMA}$. This means that any problem in BQP can be viewed as an instance of the 2-local Hamiltonian problem. Therefore, for any language $L \in \mathsf{BQP}$ 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 and the prover is restricted to BQP computations. The MF 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 [95] that if one chooses the local term S_i according to a probability distribution given by the normalised 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 a constant number of measurements each of which can be either an X or a Z measurement. Thus, 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 [63,65], 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:

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t U_{t-1} ... U_0 |x\rangle |1^t 0^{T-t}\rangle \tag{2.66}$$

This is essentially a superposition over all time steps of the time evolved state in the circuit $\mathcal{C}$. Hence, the state can be prepared by a BQP machine. The XZ-Hamiltonian, proposed by Kempe, Kitaev and Regev [65], is then a series of 2-local constraints that are all simultaneously satisfied by this state.
We can now present the steps of the MF protocol:

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

(2) The verifier determines the XZ-Hamiltonian H, associated to $\mathcal{C}|x\rangle$. In other words, she computes the terms a_i of $H = \sum_i a_i S_i$, which has as a ground state the Feynman-Kitaev state associated with $\mathcal{C}x$. Denote that state as $|\psi\rangle$.

(3) 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 normalised 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 [28] that the precision of her estimate is also inverse polynomial in $|x|$.

The protocol, as described, suggests that it is sufficient for the verifier to measure only two qubits. However, since the energy gap $b - a$, that distinguishes acceptance from rejection, decreases with the size of the input, in practice one would perform a sequential repetition of this protocol in order to boost the probability of distinguishing between the two cases.

Chapter 3

Robust and device-independent verification

> **Dr. Chase:** You can trust me.
> **Dr. House:** Problem is, if I can't trust you, I can't trust your statement that I can trust you.
>
> — House MD, Season 1, Episode 15

As mentioned, while the implementation of a large-scale universal quantum computer is still distant, there is a need to develop protocols for the verification of quantum computations. The approaches that have proven to be the most promising are those based on interactive proof systems, where a *trusted*, computationally limited verifier exchanges messages with an *untrusted*, powerful quantum prover, or multiple provers. As mentioned, it is still an open problem as to whether a fully classical verifier can validate the computations performed by a single quantum prover. Existing approaches avoid this problem by either having the verifier prepare and send quantum states to the prover, receive and measure quantum states from the prover or simply have a classical verifier interact with multiple provers that share entanglement.

One protocol, in particular, that is of interest to us, is the Fitzsimons-Kashefi, or FK, protocol. We saw that this is a prepare-and-send protocol in which the verifier is required to prepare single qubit states and send them to the prover. FK has a number of advantages. Firstly, quantum communication between the verifier and the prover is only one-way and happens at the beginning of the protocol, whereas the rest of the communication is entirely classical. Secondly, the protocol can be made to have linear *communication complexity* [88,96]. This refers to the fact that, if the verifier wishes to delegate the computation of some quantum circuit $\mathcal{C}$, then the total communication between verifier and prover scales as $O(|\mathcal{C}|)$. Thirdly, the protocol is *blind*, meaning that the description of $\mathcal{C}$ (as well as the input $|x\rangle$ on which it acts) is hidden from the prover, in an information theoretic sense. Finally, the protocol achieves inverse exponential security in a security parameter, d. In other words, for some value $d > 0$, that is chosen by the verifier, the probability of accepting an incorrect outcome is upper bounded by $(8/9)^d$.

In this chapter we wish to address two important questions about the FK protocol:

1. Is the protocol robust with respect to deviations in the quantum state that is sent by the verifier to the prover? What if the deviations are known to the prover (i.e. its system is correlated with the state prepared by the verifier)[1]?

2. Can the protocol be made *device-independent*? In other words, can one have an FK-like protocol in which all quantum devices, including the verifier's device, are untrusted?

We answer both questions in the affirmative. We will first prove the robustness of the FK protocol and then use that result in order to construct a device-independent version of the protocol. This construction will involve composing FK with the entanglement-based RUV protocol. The end result will be a protocol that takes the best of both worlds.

We note that our construction does not utilise the *universal composability (UC) framework* (that allows for secure composition of cryptographic protocols and primitives) [98], which has been successfully extended to the quantum regime [99–101]. There are two reasons for this. The first is that we are not literally composing the RUV protocol with the FK protocol. Instead, we are taking elements from the RUV protocol in order to design a new protocol that makes use of those elements and FK in order to achieve device-independent verification. The second reason is that UC requires the protocols to be proven secure in the abstract cryptographic framework [102]. While this has been done for the FK protocol, in [103], no such proof exists for the RUV protocol. Of course, this does not stop us from proving the *stand-alone* security of our protocol.

Our construction essentially works as follows: the verifier will interact classically with an untrusted measurement device and a quantum server, that are non-communicating but sharing entanglement. We will first use a sub-protocol of RUV, known as a *state tomography* protocol. This will involve the verifier directing the two untrusted devices so as to remotely prepare the quantum state used by the FK protocol (i.e. the $|+_\theta\rangle$ states and the $|0\rangle$, $|1\rangle$ dummies) on the server's system. The verifier will then run the FK protocol with the server, as if she had sent the ideal quantum states to that server. A schematic illustration of the protocol is given in Figure 3.1. At first glance, this construction might be problematic. This is because the "output" of the state tomography protocol, the state prepared on the server's side, might not necessarily be an acceptable state for the FK protocol. In particular, since the protocols are probabilistic, the state can be deviated from its intended value. Thus, it is necessary for the FK protocol to be robust to such deviations. Moreover, we have to make sure that the two quantum devices cannot exploit their shared correlations to compromise the security of the protocol.

The two main results of this chapter can be summarised as follows:

[1] Presumably, the verifier's device was acquired from some vendor. This vendor could have added (or been instructed to add) known deviations to the prepared quantum states as a type of "backdoor" into the protocol. There is, of course, precedent for this with classical devices [97].

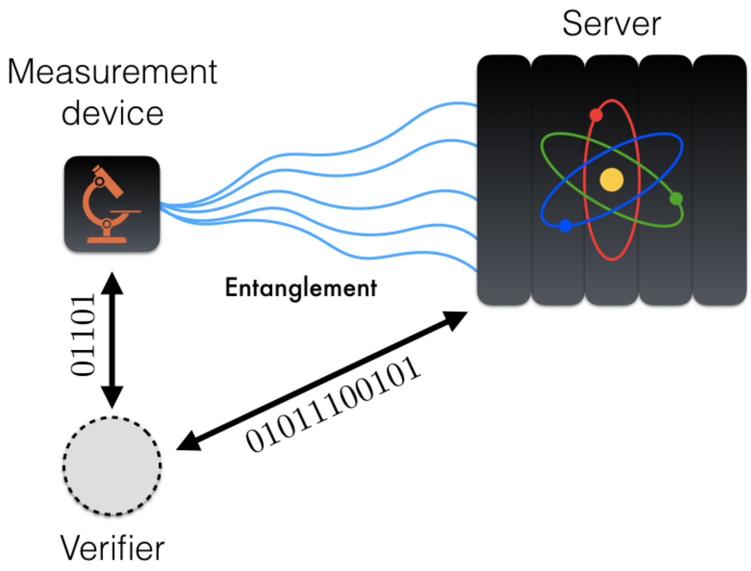


Figure 3.1: Device-independent single-server quantum verification protocol.

1. We prove that the FK protocol is strongly robust, see Theorem 3. First, we show that FK can tolerate quantum states which deviate from their ideal values by a small amount. This is shown in Lemma 1. The result assumes that the prover does not have a purification of the deviated state (in fact we are assuming that the prover's system is completely uncorrelated with the deviated state). But this is a requirement for our composite protocol. We therefore proceed to show that the FK protocol is robust even when the deviated state is correlated with an external system possessed by an adversary, like the prover. This is shown in Lemma 2.

2. As an immediate consequence of the robustness theorem, we construct our composite protocol combining RUV with FK. The required quantum states for the FK protocol, that would normally be sent by the verifier's preparation device, are prepared via the state tomography sub-protocol of RUV. We will require the verifier to have a measurement device, instead of a preparation device, which will be untrusted. This device will act as the second prover of the RUV protocol. Our composite protocol then inherits the device independence property of RUV, see Theorem 5. Additionally, since we do not require the full RUV protocol, the composite protocol also has an improved communication complexity, when compared to RUV.

We should emphasize that the composite protocol we propose does not inherit the linear communication complexity of the FK protocol. This is due to the fact that the state tomography sub-protocol adds an additional overhead in communication complexity. One can therefore wonder why we emphasized the efficiency of the FK protocol at the beginning of this chapter. The reason is that our composite protocol can be divided into a state preparation part and a verification part, this latter part corresponding to the FK protocol. The communication complexity of the whole protocol is therefore the sum of the communication complexities of these two parts. This means that any improvement in how state preparation is performed will directly improve the complexity of our approach and, in particular, if state preparation can be achieved with a linear overhead this will result in a linear communication complexity for the composite as well. Thus, the complexity of the verification part sets a lower bound for the complexity of the composite protocol.

3.1 Main Results

3.1.1 Robustness

The first result we prove is that the FK protocol is robust with respect to small variations in the quantum input. Throughout this chapter, by "quantum input" (or sometimes just "input") we will be referring to the quantum states that the verifier sends to the prover and not the input x for the quantum computation associated with the delegated circuit $\mathcal{C}$. Without loss of generality we can assume that the desired computation, that will be delegated to the server, has the fixed classical input $x = 00\ldots0$. Hence, for the rest of this chapter we define the *input state* of the FK protocol to be the tensor product of the individual qubits prepared by the verifier (including traps and dummies). These qubits will comprise the graph state that the prover should prepare by entangling the states using CZ operations. We will also refer to "ideal input" to mean the quantum input state that has no deviations. In other words this is the state that the verifier intends to prepare in a run of the FK the protocol.

Robustness of the FK protocol means that the protocol's input state can be deviated from its ideal value by some small amount and the protocol will continue to function. In particular, this input state could be the output of some other protocol, provided that this state was close to its ideal value. As we will see in the next subsection, the RUV protocol, or more specifically its state tomography sub-protocol, is capable of such a preparation. We start by formally defining robustness in this context. Before doing so, we first provide definitions for the notions of *completeness* and *soundness* that we will be using in relation to the FK protocol, though these apply for other verification protocols as well:

Definition 20 (δ-completeness). *For some $0 \leq \delta \leq 1$, we say that a verification protocol has δ-completeness, if the probability that the verifier accepts* and *the prover(s) performs the honest strategy is lower bounded by δ.*

Definition 21 (η-soundness). *For some $0 \leq \eta \leq 1$, we say that a verification protocol has η-soundness, if the probability that the verifier accepts* and *the prover(s) does not perform the honest strategy is upper bounded by η.*

Note that the FK protocol has completeness $\delta = 1$ and soundness $\eta = (8/9)^d$, for a fixed constant $d > 0$, chosen by the verifier [88]. We now define robustness as follows:

Definition 22 (Weak and Strong Robustness). *Let $V = V(\rho, \delta, \eta)$ be a verification protocol, with quantum input ρ, having completeness δ and soundness η, such that $0 \leq \delta, \eta \leq 1$. For some $\epsilon \geq 0$, we denote $V_\epsilon = V(\rho_\epsilon, \delta_\epsilon, \eta_\epsilon)$ as the* *protocol V with quantum input state ρ_ϵ, completeness δ_ϵ and soundness η_ϵ such that $0 \leq \delta_\epsilon, \eta_\epsilon \leq 1$ and:*

$$TD(\rho, \rho_\epsilon) \leq \epsilon \tag{3.1}$$

We say that protocol V is weakly robust in its quantum input (with respect to the parameters δ, η) iff:

(1) *The prover's quantum system in V_ϵ is uncorrelated with ρ_ϵ.*

(2) *It is the case that:*

$$\delta_\epsilon \to \delta \quad \eta_\epsilon \to \eta \tag{3.2}$$

as $\epsilon \to 0$.

*We say the protocol is strongly robust in its quantum input if only condition **(2)** is true*[2].

As a point of clarification, note that if a protocol has completeness $\delta > 0$ then it also has completeness δ' for some $0 \leq \delta' < \delta$. For instance, if a protocol has completeness $2/3$ then it also has completeness $1/2$. A similar fact is true for soundness. For this reason it is important to specify robustness with respect to the given completeness and soundness parameters of a protocol. In regards to the FK protocol, we will always implicitly consider robustness with respect to aforementioned parameters $\delta = 1$ and $\eta = (8/9)^d$.

For the FK protocol, in the case of a computation $\mathcal{C}$, the $N = O(|\mathcal{C}|)$-qubit input state is comprised of single qubit states π_i, where i ranges from 1 to N. We are considering running the FK protocol with an N-qubit input state ρ_ϵ such that:

$$TD\left(\rho_\epsilon, \bigotimes_{i=1}^{N} \pi_i\right) \leq \epsilon \tag{3.3}$$

Depending on whether we are interested in strong robustness or weak robustness, the prover may or may not be correlated with the state ρ_ϵ. In the case of strong robustness, one can even assume that the prover holds a purification of ρ_ϵ. Given this, we prove the following:

Theorem 3. *The FK protocol is strongly robust, in the sense of Definition 22, and given an input which is ϵ-close to its ideal value, for $\epsilon \geq 0$, the completeness becomes $\delta_\epsilon = 1 - 2\epsilon$ and the soundness becomes $\eta_\epsilon = (8/9)^d + O(\sqrt{\epsilon})$, for some constant $d > 0$, chosen by the verifier.*

Let us give a sketch of the proof, while the full proof can be found in Section 3.2

Proof sketch. We first examine soundness which considers the case of a dishonest prover. Intuitively, when the prover is malevolent, he will try to convince the verifier to accept an incorrect outcome. We know that, in the ideal case, this probability is upper bounded. One could expect that if the input state (which

[2] It might seem paradoxical that a strongly robust protocol needs to satisfy fewer constraints than a weakly robust one. However, strong robustness implies weak robustness (i.e. if a protocol is strongly robust then it is also weakly robust, whereas the converse is not necessarily true) hence the reason for naming them this way.

contains the trap qubits) is deviated from ideal, the soundness bound remains unchanged. Why? Because the effect of a deviated input could be incorporated in the deviated actions of the prover, leading to a corruption of the trap states and determining the verifier to reject. However, this is not necessarily the case, even if the input state is uncorrelated with the prover's system. The reason is that we are only assuming that the state is close to the ideal one, however the way in which it is deviated from ideal could, in principle, depend on the secret parameters. Specifically, one could have a state in which all the trap and dummy qubits are in the ideal state, whereas the state of the computation qubits is ϵ-close in trace distance to the ideal state (for some $\epsilon > 0$). In this case, even if the prover behaves honestly, the probability that the verifier accepts an incorrect outcome increases by $O(\epsilon)$. We prove that this is essentially optimal and the soundness bound changes by at most $O(\epsilon)$. We do this by using the fact that when the input state is uncorrelated with the prover's system, the action of the protocol can be viewed as a CPTP map acting on this state. Since CPTP maps are trace non-increasing, it follows that soundness bound changes by at most $O(\epsilon)$. This is shown in Lemma 1.

In the general case, however, the deviated input could be correlated with the malicious prover's system. In fact, one can even assume that the deviation was orchestrated by the prover in such a way so as to improve his cheating probability. Mathematically this is made manifest by the fact that the prover's deviation, in the presence of initial correlations, *is not*, in general, a trace preserving map. Instead, it can be expressed as a linear combination between a CPTP map and an inhomogeneous term which could be either positive or negative as shown in [104]. In this case, we use the ϵ-closeness of the input state to derive a bound of order $O(\sqrt{\epsilon})$ for the norm of the inhomogeneous term. From linearity, and using the previous argument, it follows that in the general case (of strong robustness) the soundness bound changes by at most $O(\sqrt{\epsilon})$ (see Lemma 2 for the detailed proof).

In the case of completeness, the prover is assumed to be honest. If we start with an ϵ-close input state, because of the linearity of the operators involved, we will end up with an output state that is ϵ-close to the ideal output. This is true, regardless of whether the state is correlated with the prover's system, since he is not being malicious (see Lemma 3). □

A similar approach to Lemma 1 (the proof of weak robustness) was used in [105] for defining *ϵ-blindness*. This simply means performing the UBQC protocol with a quantum state that is ϵ-close in trace distance to the ideal state. The concept was then used in [103] to prove universal composability for blind quantum computing protocols. However, to our knowledge, these results are not strong enough to cover the requirements for the composition with the RUV protocol. In [105] only the blindness property was examined while verifying the quantum computation was not considered. In [103] the authors considered a concept known as *local-verifiability* which does not take into account the possibility of correlated attacks such as those that are possible when the quantum input is correlated with the prover's system.

3.1.2 Device independence

As mentioned, we intend to use the robustness result in order to construct a device-independent version of the FK protocol. Strong robustness guarantees that if we have an input state that is only approximately the ideal one, the protocol continues to work (the completeness and soundness bounds do not change by much). We can now break the task of achieving device-independent FK into two parts, which we will compose sequentially:

1. **Verified State Preparation** - use a device-independent protocol to prepare, on the server's side, a state which is ϵ-close to the intended FK input.

2. **Verified Delegated Computation** - run the FK protocol with the server that has the ϵ-close input state (since robustness allows this).

The advantage of this modular approach is that we are free to use any protocol for state preparation as long as we have the guarantee of ϵ-closeness. This is due to our strong robustness result, which shows that FK will work even if the deviation in the prepared state is correlated with the prover's cheating strategy in the delegated computation stage. In our case, we achieve state preparation using the device-independent state tomography sub-protocol of RUV. This sub-protocol has the ϵ-closeness property that we require, as explained in [31]. The resulting protocol, will consist of a verifier with an untrusted, single-qubit, measurement device and a quantum server. While both of these devices can be viewed as provers, as in the RUV setting, the difference is that in the RUV protocol both provers are quantum computers. In our case, only one prover is required to perform universal quantum computations, whereas the other prover is only required to perform single-qubit measurements. The protocol will also have a smaller communication complexity than the RUV protocol. The complexity could be improved further if a more efficient state preparation protocol is used, such as the one from [106].

We first clarify some details of the RUV protocol, which are essential in understanding how our composite protocol will work. As we have seen, RUV uses the rigidity property of CHSH games to determine if the provers share multiple copies of the Bell state $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$, which is XZ-determined. A state is XZ-determined if it is uniquely characterised by its traces against tensor products of the I and Pauli X and Z operators. RUV uses state tomography, in the X and Z bases, to verify the preparation of XZ-determined states. In particular, they use it to tomographically verify the preparation of a set of states which can be used to perform universal computation (the resource states mentioned in Subsection 2.4.2).

This XZ state tomography is possible, because RUV leverages the rigidity result to characterise the provers' measurement operations and determine that they are indeed, up to an isometry, tensor products of X and Z observables. They also describe how it is possible to extend the protocol to include the Y observable as well. However, there is a catch. It is not possible to find an isometry that allows one to determine all three Pauli observables at the same time. Instead, one can determine that the provers are using X and Z and either Y or $-\mathsf{Y}$. In other words, the Y operator is fixed up to a sign change. That is, the provers can always choose to measure in either the Y or $-\mathsf{Y}$ bases without being detected (this corresponds to complex conjugating the states with respect to their representations in the computational basis). This problem has been mentioned by others as well [32, 107]. The reason for this potential flip of the Y operator is quite technical and exceeds the scope of this chapter. However, for our purposes, we need to mention that it is the Y operator that cannot be fixed because the Bell states that need to be determined are $|\Phi^+\rangle$ Bell states, which are XZ-determined and invariant under complex conjugation. In spite of this limitation, it is possible to force the provers to consistently choose either Y or $-\mathsf{Y}$ for their measurements, as explained in [31]. By "force" we simply mean that if the provers decide to change the Y observable to $-\mathsf{Y}$ (or vice-versa) during the protocol, this will be detected by the verifier, with high probability. For state tomography, this means that the state, that will be remotely prepared, will be close to either the ideal state or the complex conjugate of the ideal state.

At first glance it would seem that this could be problematic for the FK protocol. However, note that the complex conjugation attack affects only states in the XY-plane, while the computational basis states (the dummy qubits) are unaffected. We can simply turn this around, so that only dummies are affected, while traps and computation qubits are not. There are two ways of doing this. One is to perform MBQC (and the FK protocol) in the XZ-plane rather than the XY-plane. This is known as real MBQC [108, 109] and is used, for instance, by McKague in [32]. If one were to adapt the FK protocol to real MQBC, then the computation and trap qubits would be states from the XZ-plane, whereas the dummy qubits would be the eigenstates of the Y operator. Alternatively, we could do the following. Ultimately, the complex conjugation problem stems from the fact that we are attempting to self-test the XZ-determined $|\Phi^+\rangle$ state. Let us instead consider the Bell state $|\Psi^+\rangle = (|01\rangle + |10\rangle)/\sqrt{2}$. This state is stabilized by the operators generated by the set $\{\mathsf{X} \otimes \mathsf{X}, \mathsf{Y} \otimes \mathsf{Y}\}$. It is shown in [31] that:

Theorem 4. *A stabilizer state is determined by any of its sets of stabilizer generators.*

In other words, the $|\Psi^+\rangle$ state is XY-determined. In principle it is possible to run a form of the RUV protocol in which we choose the CHSH games such that we rigidly determine that the provers share multiple copies of the Bell state $|\Psi^+\rangle$ instead of $|\Phi^+\rangle$. Analogous to the previous case, the extended form of the protocol would then fix the Z operator up to a sign change (instead of the Y operator). This means that the provers can always perform a reflection with respect to the XY-plane with no noticeable changes (but recall that the provers are forced to do this consistently). However, the XY-plane states are invariant under such a reflection. Instead, the dummy qubits will be flipped ($|0\rangle$ is mapped to $|1\rangle$ and vice-versa). As described in Subsection 2.4.1, these dummy qubits are used to isolate the trap qubits from the computation qubits. In the honest scenario for the FK protocol, dummy qubits in the state $|1\rangle$ introduce an additional Z correction on their neighbouring qubits, effectively flipping the measurement outcomes (this is because we are using the controlled-Z operation for entangling qubits). In the dishonest case, where the prover chooses to perform the flip, all trap qubits having an unequal number of $|0\rangle$ and $|1\rangle$ dummy neighbours, will have their measurement outcome flipped, with respect to the honest case. Since the states of the dummies are chosen at random, there will be a high probability that at least one trap qubit will be of this type. The verifier can detect this and abort. We, therefore, give a modified version of the state tomography protocol of RUV, which uses this extended self-test for the $|\Psi^+\rangle$ state as well as the Pauli X, Y and Z operators (see Protocol 1). In this protocol, prover 1 is the quantum server and prover 2 is an untrusted measurement device.

The state tomography protocol is then sequentially combined with FK, resulting in our composite protocol, given as Protocol 2. Note, that since prover 1 (acting as the quantum server) is involved in both state tomography as well as the FK protocol, the strong version of the robustness property is required. This is to address the effect of any potential correlated attacks where provers 1 and 2 have agreed in advance on a strategy. Overall, the resulting protocol can be viewed as a device-independent single-server verification protocol. We therefore have the following:

Theorem 5. *Assuming the verifier wants to delegate the computation of a quantum circuit $\mathcal{C}$, Protocol 2 is a device-independent verification protocol satisfying the following completeness and soundness properties:*

Completeness: *Provided that the provers behave honestly and respect the verifier's instructions, the verifier will accept with probability at least $1 - O(|\mathcal{C}|^{-32})$.*

Soundness: *Provided that the verifier accepts with probability at least $1 - O(|\mathcal{C}|^{-64/3})$ in the verified preparation stage (step 1 of the protocol), the probability that she accepts an incorrect outcome of the computation is upper bounded by $O(|\mathcal{C}|^{-10/3})$.*

Additionally, the protocol has communication complexity $O(|\mathcal{C}|^c)$, with some constant c, $c > 2048$.

The detailed proofs are given in Section 3.3. While the obtained communication complexity is an improvement over RUV (which has a communication complexity of $O(|\mathcal{C}|^c)$, with $c > 8192$ [30, 31]), it is still far from practical. However, we believe our approach serves as a proof of principle, that this type of composition can be beneficial. Indeed, a number of protocols have been developed which have the same division between verified preparation and verified computation [34, 96, 106]. The approach also highlights where improvements could be made. It is the state tomography subprotocol that has the large communication overhead, while the FK protocol has only linear communication complexity [24, 88]. A recent approach of Coladangelo et al., from [106], inspired from [36], introduces a rigidity result with drastically smaller communication overhead. This allows them to develop entanglement-based verification protocols, with two provers, in which the total communication complexity scales as $O(|\mathcal{C}|log(|\mathcal{C}|))$. However, in their protocols both provers are universal quantum computers. It would be interesting to see whether their result can be adapted to our setting, in which one prover is a single-qubit measurement device. We leave this as an open problem, to be explored in future work.

Protocol 1 Modified State Tomography Protocol

Assumptions:

$$C = C_1 \cup C_2 \cup C_3$$

$$C_1 = \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$C_2 = \{\frac{1}{\sqrt{2}}(1,1,0), \frac{1}{\sqrt{2}}(1,-1,0), \frac{1}{\sqrt{2}}(1,0,1)\}$$

$$C_3 = \{\frac{1}{\sqrt{2}}(1,0,-1), \frac{1}{\sqrt{2}}(0,1,1), \frac{1}{\sqrt{2}}(0,1,-1)\}$$

Let M_v be a 2 outcome projective measurement defined by the projectors: $\frac{1}{2}(I + \vec{v} \cdot (\mathsf{X}, \mathsf{Y}, \mathsf{Z}))$ and $\frac{1}{2}(I - \vec{v} \cdot (\mathsf{X}, \mathsf{Y}, \mathsf{Z}))$. Let the tuple $(\vec{a}, \vec{b}) \in C \times C$ denote the measurements M_a for prover 1 and M_b for prover 2 that they need to perform on their halves on an entangled state when instructed by the verifier. Sets of such tuples define CHSH games. For example, as explained in [31], the set $\{(1,0,0), (0,0,1)\} \times \{\frac{1}{\sqrt{2}}(1,0,1), \frac{1}{\sqrt{2}}(1,0,-1)\}$ defines an XZ CHSH game as does the set $\{\frac{1}{\sqrt{2}}(1,0,1), \frac{1}{\sqrt{2}}(1,0,-1)\} \times \{(1,0,0), (0,0,1)\}$. Given C, there are six such sets of CHSH games (two XZ, two XY and two YZ) [31]. For a suitable numbering of these games, we let $CHSH_i$ be the ith type of CHSH game, $i \in \{1, ...6\}$.

Protocol: Suppose the verifier wishes to certify the preparation of $n > 0$ resource states in prover 1's system. These resource states will be the states used for the FK protocol (they are $|+_\theta\rangle$ states and $|0\rangle$, $|1\rangle$ states). Let $n_g = O(n^{64})$ and $N = O(n_g^{\alpha-1})$, with $\alpha > 32$. With probability 1/2, the verifier chooses to perform one of the following:

(1) CHSH games. The verifier referees 6 groups of Nn_g sequential CHSH games between prover 1 and prover 2. Group i contains only CHSH games of type $CHSH_i$. The order of the games is chosen at random by the verifier. In other words, from the provers' perspectives, game j is equally likely to be any of the 6 possible games. For each group of CHSH games the verifier rejects if the two provers win less than:

$$\cos^2(\pi/8)Nn_g - \frac{1}{2\sqrt{2}}\sqrt{Nn_g \log(Nn_g)}$$

of the games.

(2) State tomography. Let R_6 be a random permutation of the elements $\{1, ...6\}$. For j going from 1 to 6, the verifier does the following. She chooses $K \in [N]$ uniformly at random and also chooses $CHSH_{R_6(j)}$ as one of the 6 possible CHSH games. She referees $(K-1)n_g$ $CHSH_{R_6(j)}$ games, which can be viewed as $(K-1)$ rounds, each consisting of n_g CHSH games. In the Kth round, she will send n_g questions corresponding to $CHSH_{R_6(j)}$ to prover 1, while running a state tomography protocol with prover 2. In this protocol prover 2 is asked to prepare resource states by measuring his halves of the shared Bell states. This will collapse prover 1's states to the same resource states up to corrections. The verifier uses the measurement outcomes of prover 1 to tomographically check this preparation. She rejects if the tomography statistics are inconsistent. For the remaining $N - K$ rounds, she again referees n_g $CHSH_{R_6(j)}$ games.

Protocol 2 Composite Verification Protocol

1. Run the modified state tomography protocol (Protocol 1).
2. From the states prepared by this protocol on prover 1's side, select the input for FK and run the FK protocol with prover 1.

3.2 Proof of Robustness

In this section we prove the robustness of the FK protocol. We will first look at how the soundness of the protocol is affected by having a deviated input state and then examine the simpler case of completeness. To do this, we start by considering weak robustness, as per Definition 22. Recall that this means that the input state is ϵ-deviated but uncorrelated with the prover's system. We will then remove this assumption and consider strong robustness.

Lemma 1. *If the initial input state of the FK protocol is ϵ-close to the ideal input state, for some $\epsilon \geq 0$, and uncorrelated with the prover's system, the soundness bound changes by at most $O(\epsilon)$.*

Proof. To prove this, we will incorporate the deviated state into the outcome density operator from Subsection 2.4.1. Recall that the outcome density operator of the FK protocol is denoted $B_j(\nu)$, where ν denotes the verifier's secret parameters and j ranges over the prover's choices of possible actions (with $j = 0$ being the correct/honest action). For a given computation $\mathcal{C}$, the projector associated with the output state being incorrect *and* the verifier accepting was the following:

$$P^{\nu}_{incorrect} = (\mathbb{I} - \mathcal{C} \left|0\right\rangle \left\langle 0\right| \mathcal{C}^{\dagger}) \otimes \bigotimes_{t \in T} \left|\eta^{\nu_T}_t\right\rangle \left\langle\eta^{\nu_T}_t\right| \tag{3.4}$$

Where $\bigotimes_{t \in T} \left|\eta^{\nu_T}_t\right\rangle \left\langle\eta^{\nu_T}_t\right|$ is the state associated with the trap qubits. Note that, unlike in Subsection 2.4.1, we are considering the more general projector which has multiple trap qubits, as opposed to just one. Also note that, as we mentioned in Section 3.1, we are taking the input to the computation to be $\left|0\right\rangle$ and incorporating the preparation of an arbitrary input in $\mathcal{C}$. However, what we have been referring to as the quantum input state of the protocol will be denoted $\left|\psi^{\nu}_{in}\right\rangle$. This state has a dependency on the secret parameters, since it consists of the $\left|+_{\theta_i}\right\rangle$ states that act as computation qubits and traps, as well as the $\left|0\right\rangle$, $\left|1\right\rangle$ dummy qubits. The choices for the angles $\{\theta_i\}_i$ as well as for the values of the dummies are encoded in ν. The associated probability for accepting an incorrect outcome is $p_{incorrect}$ and can be expressed as:

$$p_{incorrect} = \sum_{\nu} p(\nu) Tr(P^{\nu}_{incorrect} B_j(\nu)) \tag{3.5}$$

which is a weighted average of the incorrect outcome probabilities (expressed by the trace operator) over all choices of the secret parameters. We now need to consider the exact form of the outcome density operator:

$$B_j(\nu) = Tr_P \left(\sum_b |b + c_r\rangle \langle b| C_{\nu_C,b} \Omega P \; \sigma^{\nu,b} \; P^\dagger \Omega^\dagger C^\dagger_{\nu_C,b} |b\rangle \langle b + c_r| \right) \tag{3.6}$$

where:

$$\sigma^{\nu,b} = \bigotimes |0\rangle \langle 0|_P \otimes |\psi^\nu_{in}\rangle \langle \psi^\nu_{in}| \otimes \bigotimes_i |\delta^b_i\rangle \langle \delta^b_i| \tag{3.7}$$

Let us explain the two equations:

- We are tracing over the prover's qubits.

- b indicates the possible branches of computation parameterised by the measurement results sent by the prover to the verifier. Thus, b denotes classical information. It should be noted that b denotes measurements performed up to but *excluding* the last layer, which is the output of the computation (on which we perform the projection with $P^\nu_{incorrect}$).

- c_r indicates corrections that need to be performed on the classical output (the b's) due to the MBQC computation together with the random phases introduced by the verifier. In other words $b + c_r$ represents the corrected outcomes (as if we had performed the MBQC computation of $\mathcal{C}$ and there was no need for corrections).

- $C_{\nu_C,b}$ are the corrections the prover should apply to the output state (i.e. the final layer in the MBQC computation), depending on the previous measurement outcomes (as well as the verifier's secret parameters).

- Ω is a unitary map which represents the prover's deviation from the desired computation. Technically, Ω should have a dependency on j, but this was suppressed to simplify the already cumbersome notation. When $j = 0$, $\Omega = I$.

- P is the ideal action of the protocol, that the prover should perform in the honest setting. Mathematically this is expressed as $P = (\bigotimes_i \mathsf{H}_i \mathsf{Z}_i(\delta_i)) \, E_G$, where $\mathsf{Z}_i(\delta_i)$ is a unitary rotation around the Z axis of the Bloch sphere by δ_i radians, and E_G is the entanglement operator. This operator corresponds to performing CZ gates on the provided qubits, according to the graph structure G. Note that the $\mathsf{Z}_i(\delta_i)$ operation is symbolic, since P should be independent of the δ_i angles. The actual operation is a controlled rotation around the Z axis, where the control system will be a state of the form $|\delta^b_i\rangle$, contained in $\sigma^{\nu,b}$.

- The joint state, comprised of the input state, the prover's qubits, as well as the measurement angles δ^b_i, is denoted $\sigma^{\nu,b}$. Note that the measurement angles depend on b, since future angles are adapted based on past measurement outcomes.

We now need to incorporate the approximate input state into this operator. For some $\epsilon \geq 0$, let ρ^{ν} be a state, such that:

$$TD(\rho^{\nu}, |\psi^{\nu}_{in}\rangle \langle\psi^{\nu}_{in}|) \leq \epsilon \tag{3.8}$$

Instead of using $\sigma^{\nu,b}$ in the expression for $B_j(\nu)$, we will use:

$$\tau^{\nu,b} = \bigotimes |0\rangle \langle 0|_P \otimes \rho^{\nu} \otimes \bigotimes_i |\delta^b_i\rangle \langle\delta^b_i| \tag{3.9}$$

Note that:

$$TD(\tau^{\nu,b}, \sigma^{\nu,b}) \leq \epsilon \tag{3.10}$$

and that ρ^{ν} and the prover's system are in product form (i.e. they are not correlated). We then have:

$$B'_j(\nu) = Tr_P\left(\sum_b |b + c_r\rangle \langle b| C_{\nu_C,b}\Omega' P \; \tau^{\nu,b} \; P^{\dagger}\Omega'^{\dagger}C^{\dagger}_{\nu_C,b} |b\rangle \langle b + c_r|\right) \tag{3.11}$$

Because the input state and the prover's system are uncorrelated, the state $B'_j(\nu)$ can be written as:

$$B'_j(\nu) = \mathcal{E}(\rho^{\nu}) \tag{3.12}$$

where $\mathcal{E}$ is a CPTP map defined by acting on the state with the honest run of the protocol, followed by the prover's deviation, followed by tracing out the prover's system. Since CPTP maps cannot increase the trace distance, we have:

$$TD(B_j(\nu), B'_j(\nu)) \leq TD(\rho^{\nu}, |\psi^{\nu}_{in}\rangle \langle\psi^{\nu}_{in}|) \leq \epsilon \tag{3.13}$$

Projection operators are also trace non-increasing, and if we consider $P^{\nu}_{incorrect}$ we have that:

$$TD(P^{\nu}_{incorrect}B_j(\nu), P^{\nu}_{incorrect}B'_j(\nu)) \leq \epsilon \tag{3.14}$$

Next we use the reverse triangle inequality, which gives us:

$$\begin{aligned}\big|\; \|P^{\nu}_{incorrect}B_j(\nu)\|_{Tr} - \|P^{\nu}_{incorrect}B'_j(\nu)\|_{Tr} \;\big| \leq \\ TD(P^{\nu}_{incorrect}B_j(\nu), P^{\nu}_{incorrect}B'_j(\nu)) \leq \epsilon\end{aligned} \tag{3.15}$$

And since we are dealing with positive definite operators, we know that:

$$\|P^{\nu}_{incorrect}B_j(\nu)\|_{Tr} = \frac{1}{2}Tr(P^{\nu}_{incorrect}B_j(\nu)) \tag{3.16}$$

$$\|P^{\nu}_{incorrect}B'_j(\nu)\|_{Tr} = \frac{1}{2}Tr(P^{\nu}_{incorrect}B'_j(\nu)) \tag{3.17}$$

We therefore find that:

$$|Tr(P^{\nu}_{incorrect}B'_j(\nu)) - Tr(P^{\nu}_{incorrect}B_j(\nu))| \leq 2\epsilon \tag{3.18}$$

It immediately follows that the probability of accepting an incorrect outcome in the case where the input state is deviated, denoted $p'_{incorrect}$ will satisfy:

$$|p'_{incorrect} - p_{incorrect}| \leq 2\epsilon \tag{3.19}$$

This follows from the fact that:

$$p'_{incorrect} = \sum_{\nu} p(\nu) Tr(P^{\nu}_{incorrect} B'_j(\nu)) \tag{3.20}$$

which is a convex sum of terms each of which is 2ϵ-close to a term from the corresponding expression for $p_{incorrect}$. Thus, the soundness bound of the protocol changes by at most 2ϵ. □

We now examine the more general case, in which we make no assumption about how the input state has been ϵ-deviated. In this case, the state can be correlated with the prover's system. To address this issue, we make use of the Gentle Measurement Lemma [110, 111], which states the following:

Theorem 6 (Gentle Measurement Lemma). *Let ρ be a state, and Π a projector. Then:*

$$TD(\rho, \sqrt{\Pi}\rho\sqrt{\Pi}) \leq 2\sqrt{1 - Tr(\Pi\rho)} \tag{3.21}$$

More specifically, we will use a corollary of this, that was introduced in [31]:

Corollary 1. *Let ρ_{AB} be a state in $\mathcal{D}(\mathcal{H}_A \otimes \mathcal{H}_B)$, and let π be a pure state on $\mathcal{H}_A$. If for some $\delta \geq 0$, $\mathrm{Tr}(\pi \, \mathrm{Tr}_B(\rho_{AB})) \geq 1 - \delta$, then*

$$TD(\rho_{AB}, \pi \otimes \mathrm{Tr}_A(\rho_{AB})) \leq 2\sqrt{\delta} + \delta \tag{3.22}$$

This will allow us to show the following:

Lemma 2. *If the initial input state of the FK protocol is ϵ-close to the ideal input state, for some $\epsilon \geq 0$, the soundness bound changes by at most $O(\sqrt{\epsilon})$.*

Proof. Consider a composite state $\rho_{VP} \in \mathcal{D}(\mathcal{H}_V \otimes \mathcal{H}_P)$, where we identify system V with the verifier and system P with the prover. Additionally, let:

$$\rho_V = Tr_P(\rho_{VP}) \quad \rho_P = Tr_V(\rho_{VP}) \tag{3.23}$$

The state ρ_V is the deviated input state of the verifier, which can be correlated with the prover's system, ρ_P. Thus, if the state ρ_V is used as input for the FK protocol, the existence of initial correlations can be exploited by an adversarial prover. This means that the action of the protocol (which includes the prover's deviation) can no longer be expressed as a CPTP map over this subsystem. To characterise the more general deviation, let us first consider:

$$\rho_{corr} = \rho_{VP} - \rho_V \otimes \rho_P \tag{3.24}$$

Now, assume that the state ρ_{VP} evolves under the action of some unitary U. We will later identify this unitary with the honest action of the protocol and prover's deviation. As it is shown in [104], in the presence of correlations, the evolution of subsystem ρ_V is given by:

$$\rho'_V = Tr_P(U\rho_{VP}U^\dagger) = Tr_P(U\rho_V \otimes \rho_P U^\dagger) + Tr_P(U\rho_{corr}U^\dagger) = \mathcal{E}(\rho_V) + \delta\rho_V \quad (3.25)$$

where $\mathcal{E}$ is a CPTP map and $\delta\rho_V$ is known as an *inhomogeneous term* which is added to the CPTP evolution due to the presence of initial correlations. This inhomogeneous term need not be positive, since it is defined as:

$$\delta\rho_V = Tr_P(U\rho_{corr}U^\dagger) \quad (3.26)$$

We can now see that if we substitute ρ_V in the outcome density operator of the FK protocol, and use the above relations (identifying $U = P\Omega C_{\nu_C,b}$), we would have a term that is identical to the one obtained in Lemma 1, corresponding to $\mathcal{E}(\rho_V)$, and a term arising from the inhomogeneous term $\delta\rho_V$.

To determine how much this extra term affects the soundness bound, we make use of the ϵ-closeness of ρ_V to the ideal input state $|\psi^\nu_{in}\rangle$. We first find a bound for the norm of ρ_{corr}, and since $\delta\rho_V$ is obtained by tracing out the prover's system from ρ_{corr}, the same bound will hold for the norm of $\delta\rho_V$ as well. The soundness bound can change by at most the norm of this term. This is because averaging over the different computational paths (the b's) and over the verifier's secret parameters (the ν's) cannot increase the norm.

We start by noting that:

$$TD(\rho_V, |\psi^\nu_{in}\rangle\langle\psi^\nu_{in}|) \leq \epsilon \quad (3.27)$$

for some $\epsilon \geq 0$. It is also known, from the relationship between fidelity and trace distance, that:

$$1 - \langle\psi^\nu_{in}|\rho_V|\psi^\nu_{in}\rangle \leq TD(\rho_V, |\psi^\nu_{in}\rangle\langle\psi^\nu_{in}|) \quad (3.28)$$

Combining these two yields:

$$\langle\psi^\nu_{in}|\rho_V|\psi^\nu_{in}\rangle \geq 1 - \epsilon \quad (3.29)$$

But since $Tr(|\psi^\nu_{in}\rangle\langle\psi^\nu_{in}|\rho_V) = \langle\psi^\nu_{in}|\rho_V|\psi^\nu_{in}\rangle$, using Equation 3.29 and Corollary 1 we have:

$$TD(\rho_{VP}, |\psi^\nu_{in}\rangle\langle\psi^\nu_{in}| \otimes \rho_P) \leq 2\sqrt{\epsilon} + \epsilon \quad (3.30)$$

Now, the trace norm of ρ_{corr} is simply the trace distance between ρ_{VP} and $\rho_V \otimes \rho_P$, as can be seen from Equation 3.24. Using the triangle inequality, we have:

$$TD(\rho_{VP}, \rho_V \otimes \rho_P) \leq TD(\rho_{VP}, |\psi^\nu_{in}\rangle\langle\psi^\nu_{in}| \otimes \rho_P) + TD(|\psi^\nu_{in}\rangle\langle\psi^\nu_{in}| \otimes \rho_P, \rho_V \otimes \rho_P) \quad (3.31)$$

The second term is simply ϵ and the first term is upper bounded from Equation 3.30 so:

$$TD(\rho_{VP}, \rho_V \otimes \rho_P) \leq 2\sqrt{\epsilon} + 2\epsilon \quad (3.32)$$

Hence, for the trace norm of ρ_{corr} (and $\delta\rho_V$) we have:

$$||\rho_{corr}||_{Tr} \leq 2\sqrt{\epsilon} + 2\epsilon \quad (3.33)$$

which is a bound of order $O(\sqrt{\epsilon})$. Therefore, the inhomogeneous term can change the soundness bound of the FK protocol by at most $O(\sqrt{\epsilon})$. This concludes the proof. □

We now examine the completeness of the protocol.

Lemma 3. *If the initial input state of the FK protocol is ϵ-close to the ideal input state, for some $\epsilon \geq 0$, the protocol's completeness will be lower bounded by $1 - 2\epsilon$.*

Proof. As we have seen, the FK protocol can be abstractly thought of as a CPTP map $\mathcal{P}$, that takes some input state to an output state. Since we are assuming the prover is honest, the output state, for an ideal (non-deviated) input, will be $B_0(\nu)$. In the deviated input case, we will again denote the output state as $B'_0(\nu)$. Writing these out explicitly we have:

$$B_0(\nu) = \mathcal{P}(|\psi^{\nu}_{in}\rangle \langle\psi^{\nu}_{in}|) \tag{3.34}$$

$$B'_0(\nu) = \mathcal{P}(\rho^{\nu}) \tag{3.35}$$

Where, ρ^{ν} is the deviated input:

$$TD(\rho^{\nu}, |\psi^{\nu}_{in}\rangle \langle\psi^{\nu}_{in}|) \leq \epsilon \tag{3.36}$$

Following the same argument as in Lemma 1, we find that:

$$||P_{correct}B_0(\nu)||_{Tr} = \frac{1}{2}Tr(P_{correct}B_0(\nu)) \tag{3.37}$$

$$||P_{correct}B'_0(\nu)||_{Tr} = \frac{1}{2}Tr(P_{correct}B'_0(\nu)) \tag{3.38}$$

But $Tr(P_{correct}B_0(\nu)) = 1$, since the FK protocol (with ideal input state) has completeness 1. Thus:

$$|1 - Tr(P_{correct}B'_0(\nu))| \leq 2\epsilon \tag{3.39}$$

Lastly, because $Tr(P_{correct}B'_0(\nu)) \leq 1$, we get:

$$1 - 2\epsilon \leq Tr(P_{correct}B'_0(\nu)) \tag{3.40}$$

In other words, the probability of accepting a correct outcome, under the assumption that the input state is ϵ-close to the ideal input, is at least $1 - 2\epsilon$. □

It is now easy to see that the proof of Theorem 3 follows directly from Lemmas 2 and 3. Having the robustness property, the FK protocol can receive an input, which is ϵ-close to its ideal value, from another protocol. As we have shown, even if this input is correlated with an external and adversarial system, we can still perform the verification as long as ϵ is sufficiently small.

3.3 Proof of Compositionality

To prove the security of the composite protocol, we first need to prove that the FK protocol rejects with high probability a state close to a reflection with respect to the XY-plane (Lemma 4). Then we prove that, excluding the XY-reflected case, the modified state tomography protocol (Protocol 1), satisfies the ϵ-closeness property required by FK. This is shown in Lemma 5.

Lemma 4. *The soundness bound of the FK protocol, with ϵ-deviated input state, does not change, if the initial input state is ϵ-close to a reflection along the XY-plane, of the ideal input state, for some $\epsilon \geq 0$.*

Proof. Let us denote the reflection operation as $\mathcal{R}_{\mathsf{XY}}$. We will first consider the case in which we have an exact reflection of the ideal input state and then consider the ϵ-close case.

Recall that the input to the FK protocol consists of XY-plane states and dummy qubits which are either $|0\rangle$ or $|1\rangle$. The XY-plane states are invariant under the reflection, while the dummy states will be flipped. More formally:

$$\forall\theta \in [0, 2\pi], \mathcal{R}_{\mathsf{XY}}(|+_\theta\rangle) = |+_\theta\rangle \tag{3.41}$$

and:

$$\mathcal{R}_{\mathsf{XY}}(|0\rangle) = |1\rangle \quad \mathcal{R}_{\mathsf{XY}}(|1\rangle) = |0\rangle \tag{3.42}$$

As in the proof outlined in Subsection 2.4.1, the outcome density operator of the FK protocol will be:

$$B_j(\nu) = Tr_P\left(\sum_b |b + c_r\rangle \langle b| \, C_{\nu_C,b}\Omega P \; \sigma^{\nu,b} \; P^\dagger\Omega^\dagger C^\dagger_{\nu_C,b} \, |b\rangle \langle b + c_r|\right) \tag{3.43}$$

except this time $\sigma^{\nu,b}$ is:

$$\sigma^{\nu,b} = \bigotimes |0\rangle \langle 0|_P \otimes \mathcal{R}_{\mathsf{XY}}\left(|\psi^\nu_{in}\rangle \langle \psi^\nu_{in}|\right) \otimes \bigotimes_i |\delta^b_i\rangle \langle \delta^b_i| \tag{3.44}$$

We can now go through the proof of the FK protocol, as outlined in Subsection 2.4.1, noting the changes resulting from having the reflected state. First of all, $B_j(\nu)$ can still be split into a term containing the honest action of the prover $B_0(\nu)$ and the complement of this. Of course, in this case, $B_0(\nu)$ represents the state obtained when acting with the ideal operators of the protocol on the reflected state. Since the computation state remains unaffected, this term will still be cancelled by $P^\nu_{incorrect}$. Following the proof, we will have that the probability of accepting an incorrect outcome is upper bounded by:

$$Tr\left(\sum_{\nu^t} p(\nu^t) \, |+^{\nu^t}_{\theta_t}\rangle \langle +^{\nu^t}_{\theta_t}| \sum_{k,l,m} \alpha_{kl}(j)\alpha^*_{km}(j) \; P_l(|+^{\nu^t}_{\theta'_t}\rangle \langle +^{\nu^t}_{\theta'_t}| \otimes (I/Tr(I)))P_m\right) \tag{3.45}$$

This is similar to the state from expression 2.60, except the trap state has angle

θ'_t instead of θ_t. The difference is the following:

$$\theta_t = \theta_t + \sum_{j \in N_G^D(t)} d_j \pi \qquad \theta'_t = \theta_t + \sum_{j \in N_G^D(t)} (d_j \oplus 1)\pi \tag{3.46}$$

where $N_G^D(t)$ denotes the set of dummy neighbours of trap t in $|G\rangle$, and d_j is the value of dummy qubit j. Note that for θ'_t the difference stems from the reflection operator essentially flipping the values of the dummies (d_j has become $d_j \oplus 1$). Also note that the system containing the computation and dummy qubits is still the maximally mixed state (just like in expression 2.60). The reason for this is, on the one hand, because the XY-plane computation states are invariant under $\mathcal{R}_{\mathsf{XY}}$, and on the other hand, because it flips the dummy qubits. But flipping qubits that were chosen uniformly at random to be either $|0\rangle$ or $|1\rangle$ does not change the density matrix associated to this state, which is still the maximally mixed state.

For the identity system, we will, as before, have all terms for which $l \neq m$ be equal to zero. But, for the trap system, the situation will be slightly different, from that of the original proof. Specifically, we have:

$$Tr\left(\sum_{\nu^t} p(\nu^t) \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \ P_l \left|+^{\nu^t}_{\theta'_t}\right\rangle \left\langle +^{\nu^t}_{\theta'_t}\right| P_m \right) =$$
$$\sum_{\nu^t} p(\nu^t) \left\langle +^{\nu^t}_{\theta_t}\right| P_l \left|+^{\nu^t}_{\theta'_t}\right\rangle \left\langle +^{\nu^t}_{\theta'_t}\right| P_m \left|+^{\nu^t}_{\theta_t}\right\rangle \tag{3.47}$$

If $\theta_t = \theta'_t$ then the rest of the proof proceeds as in the normal FK case, since the above expression will be zero in all but the cases in which $l = m$, thanks to the Pauli twirl. So when does $\theta_t = \theta'_t$? As we can see from 3.46, making the values equal means:

$$\sum_{j \in N_G^D(t)} d_j = \sum_{j \in N_G^D(t)} (d_j \oplus 1) \tag{3.48}$$

Now note that if $|N_G^D(t)|$ is even, the two expressions are equal, whereas when $|N_G^D(t)|$ is odd, the expressions differ by a one (since we're working over $\mathbb{Z}_2$). For this latter case, this means that expression 3.47 becomes:

$$\sum_{\nu^t} p(\nu^t) \left\langle +^{\nu^t}_{\theta_t}\right| P_l \mathsf{Z} \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \mathsf{Z} P_m \left|+^{\nu^t}_{\theta_t}\right\rangle \tag{3.49}$$

We can commute Z and P_l as $P_l \mathsf{Z} = (-1)^{f(l)} \mathsf{Z} P_l$, where $f(l) = 0$ if P_l and Z commute and $f(l) = 1$ otherwise. This leads to:

$$(-1)^{f(l)} \sum_{\nu^t} p(\nu^t) \left\langle +^{\nu^t}_{\theta_t}\right| \mathsf{Z} P_l \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| \mathsf{Z} P_m \left|+^{\nu^t}_{\theta_t}\right\rangle \tag{3.50}$$

By making the substitution $Q_l = \mathsf{Z} P_l$ and $Q_m = \mathsf{Z} P_m$ we have:

$$(-1)^{f(l)} \sum_{\nu^t} p(\nu^t) \left\langle +^{\nu^t}_{\theta_t}\right| Q_l \left|+^{\nu^t}_{\theta_t}\right\rangle \left\langle +^{\nu^t}_{\theta_t}\right| Q_m \left|+^{\nu^t}_{\theta_t}\right\rangle \tag{3.51}$$

But Q_l and Q_m are arbitrary Pauli operators, which means we can now use the Pauli twirl relation so that the above expression is zero when $l \neq m$. This implies that the upper bound on soundness becomes:

$$Tr\left(\sum_{\nu^t} p(\nu^t)\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right| \sum_{k,l}(-1)^{f(l)}|\alpha_{kl}(j)|^2\ P_l(\left|+^{\nu^t}_{\theta_t}\right\rangle\left\langle+^{\nu^t}_{\theta_t}\right| \otimes (I/Tr(I)))P_l\right) \tag{3.52}$$

While this is not a convex combination of Pauli deviations, due to the negative terms, those terms cannot increase the value of the bound. The important requirement for the proof was to eliminate the so-called "off-diagonal" terms, i.e. the terms for which $l \neq m$. Having done so, the above expression, as well as the one for the case when $|N^D_G(t)|$ is even, will be upper bounded by $1 - 1/N$ (or 8/9 when considering multiple traps and using the special graph state of [88]) as in the proof given in Subsection 2.4.1.

We therefore find that the XY-plane reflected input state does not change the soundness bound of the FK protocol. For the case when the input state is ϵ-close to the reflected state, using the same argument as in the robustness proof (Lemmas 1, 2), the soundness bound changes by at most $O(\sqrt{\epsilon})$. This concludes the proof. □

Let us now turn our attention to the composite protocol (Protocol 2). As we mentioned in Subsection 3.1.2, this protocol works by using a modified state tomography sub-protocol (Protocol 1) to prepare a state (or rather, certify the preparation of a state) that is ϵ-close to the ideal state of the FK protocol, or to its XY-reflected version, and then running the FK protocol. We would like to show that the modified state tomography sub-protocol does indeed certify the preparation of the desired state. Before doing so, we first state the definition of a *state tomography protocol*[3] from [31] and also a theorem concerning such protocols[4].

Definition 23. *An* XZ *(*XY*,* YZ*) state tomography protocol is parameterised by natural numbers q, n and m, with $qn \leq m$, a q-qubit POVM $\mathcal{Q}$ with at most 2^q outcomes, and a list σ of qn distinct indices from $[m]$. The protocol involves a verifier, Eve, and two provers, Alice and Bob. Alice and Bob share a state in $\mathcal{H}_A \otimes \mathcal{H}_B$. The protocol proceeds as follows:*

- *Eve's interaction with Alice has m rounds. In round j, Eve sends Alice an independent, uniformly random bit, A_j. Alice applies a two-outcome projective measurement on $\mathcal{H}_A$ to determine her reply $X_j \in \{0, 1\}$.*
- *Eve has one round of interaction with Bob. First, Eve sends Bob the list σ. Bob returns to Eve a string $O_1, \ldots, O_n$, with the $O_j \in [2^q]$ determined by successive 2^q-outcome projective measurements on $\mathcal{H}_B$.*

[3] Specifically, this is Definition 6.11 from [31]. The original definition was specific to XZ tomography and we changed it slightly to emphasise this fact and how it can be altered for XY and YZ tomography, respectively.

[4] Specifically, this is Theorem 6.17 from [31].

No other communication is allowed.

Alice's strategy is ideal*, with respect to an isometry* $U^A : \mathcal{H}_A \hookrightarrow (\mathsf{C}^2)^{\otimes m} \otimes \mathcal{H}'_A$*, if in round* j *of her interaction with Eve, Alice returns the result of measuring the* j*th qubit with the* Z *(*X*,* Y*) observable, if* $A_j = 0$*, or the* X *(*Y*,* Z*) observable, if* $A_j = 1$*.*

Alice and Bob's joint strategy is ideal*, with respect to the isometries* $U^D : \mathcal{H}_D \hookrightarrow (\mathsf{C}^2)^{\otimes m} \otimes \mathcal{H}'_D$*,* $D \in \{A, B\}$*, if Alice's strategy is ideal with respect to* U^A *and if*

1. *The initial state consists of* m *EPR states in tensor product with a state in* $\mathcal{H}'_A \otimes \mathcal{H}'_B$*, and*

2. *Bob returns the results of measuring with* $\mathcal{Q}$ *each successive block of* q *qubits specified in* σ*.*

Theorem 7. *Fix* $\mathcal{Q} = \{\pi^1, \ldots, \pi^{2^q}\}$ *a complete, orthonormal set of* q*-qubit* XZ*-determined pure states. For a sufficiently large constant* α *and for sufficiently large* n*, let* $m = m(n) \geq qn$ *and* $N \geq m^{\alpha-1}$*. Let* $\sigma \in [m]^{qn}$ *be a list of distinct indices. Consider a combination of the following two protocols between the verifier, Eve, and the provers, Alice and Bob:*

1. *CHSH games: In the first protocol, Eve referees* Nm *sequential CHSH games. She accepts if*

$$\left|\{j \in [Nm] : A_j B_j = X_j \oplus Y_j\}\right| \geq \cos^2(\pi/8) Nm - \tfrac{1}{2\sqrt{2}}\sqrt{Nm \log(Nm)} \tag{3.53}$$

2. *State tomography: In the second protocol, Eve chooses* $K \in [N]$ *uniformly at random. She referees* $(K-1)m$ *CHSH games. For the* K*th set, she referees a state tomography protocol with parameters* q*,* n*,* m*,* $\mathcal{Q}$ *and* σ*. She accepts if the following criteria are satisfied:*

$$\max_{o \in [2^q]} \left|\#\{j : O_j = o\} - n/2^q\right| \leq 4^q \sqrt{n \log n} \tag{3.54a}$$

$$\max_{o \in [2^q], P \in \{I,X,Z\}^{\otimes q}} |\tau^{o,P} - \mathrm{Tr}(\pi^o P)| \leq 4^q \sqrt{(\log n)/n} \tag{3.54b}$$

The combined protocol satisfies the following completeness and soundness conditions:

Completeness: *If Alice and Bob use* Nm *shared EPR states to play the CHSH games according to an ideal strategy, and if Bob uses an ideal strategy with respect to the projections* $\mathcal{Q}$ *on the* K*th set of* m *EPR states in the state tomography protocol, then in both protocols,*

$$\Pr[\textit{Eve accepts}] \geq 1 - O(n^{-1/2}) \tag{3.55}$$

Soundness: *Assume that for both protocols,* $\Pr[\textit{Eve accepts}] \geq 1 - n^{-1/3}$*. Let* ρ *be Alice's state in the second protocol after* $(K-1)m$ *games and conditioned*

on Bob's messages
$O_1, \ldots, O_n$. *Then there exists an isometry* $\mathcal{X}^A : \mathcal{H}_A \hookrightarrow (\mathsf{C}^2)^{\otimes m} \otimes \mathcal{H}'_A$ *such that letting* $\rho_{\sigma,j}$ *be* $\mathcal{X}^A \rho \mathcal{X}^{A\dagger}$ *reduced to Alice's qubits* $\{\sigma(j,i) : i \in [q]\}$,

$$\Pr\Big[\big|\{j \in [n] : \mathrm{Tr}(\rho_{\sigma,j}\pi^{O_j}) \geq 1 - O(n^{-1/16})\}\big| \geq \big(1 - O(n^{-1/16})\big)n\Big] \\ \geq 1 - 4n^{-1/12} \quad (3.56)$$

Here, the probability is over K*, the first* $(K-1)m$ *games and* $O_1, \ldots, O_n$.

Consider now the following corollary to this theorem:

Corollary 2. *There is a state tomography protocol for q-qubit* XY*-determined (*YZ*-determined) states, which achieves the same completeness and soundness bound as the one from Theorem 7.*

Proof. The characterisation of the XZ-determined pure states is done up to an isometry. Indeed, the CHSH game (the self-testing using that game) allows for the characterisation of Alice and Bob's measurement operators and shared state up to an isometry. However, if such an isometry is shown to exist, then, of course, there are also isometries for any other pair of anti-commuting 2-outcome observables, such as X and Y or Y and Z. In other words, the result of Theorem 7 holds for XY-determined pure states, as well as YZ-determined pure states. □

We can now give the main lemma proving that Protocol 1 certifies the resource states required for the FK protocol.

Lemma 5. *Protocol 1 achieves the following completeness and soundness conditions:*

Completeness: *If the provers use* $6Nn_g$ *shared EPR states to play the CHSH games according to an ideal strategy, and if prover* 2 *performs the instructed measurements in the state tomography part of the protocol, then:*

$$\Pr[\textit{verifier accepts}] \geq 1 - O(n_g^{-1/2}) \qquad (3.57)$$

Soundness: *Provided that the verifier accepts with probability at least* $1 - O(n_g^{-1/3})$ *then, with probability at least* $1 - O(n_g^{-1/48})$*, it is the case that for each of the* 6 *state tomography subprotocols:*

$$TD\left(\rho_S(O_{1,n_g}), \bigotimes_{j \in S} \pi^{O_j}\right) \leq O(n_g^{-1/64}) \qquad (3.58)$$

where $\rho_S(O_{1,n_g})$ *is the reduced state on prover* 1*'s system, conditioned on outcomes* O_{1,n_g} *of prover* 2*, for a subset* $S \subseteq [n_g]$, $|S| = n$*, and* $\bigotimes_{j \in S} \pi^{O_j}$ *is the intended resource state. The trace distance is up to an isometry on the provers' systems that determines the observables* X *and* Y*, whereas* Z *is determined up to a reflection.*

Proof. As mentioned, the set C of measurement directions, from Protocol 1, defines 6 CHSH games between the two provers, Alice (prover 1) and Bob (prover 2). In fact, the 6 games together define an extended CHSH game, as explained in [31]. The main difference, with respect to [31], is that in that paper, the authors take the probability of Alice and Bob's outcomes to be x and y, when performing the ideal strategy (i.e. measuring $|\Phi^+\rangle$ EPR states along directions $\vec{a}$, $\vec{b}$) to be:

$$p_{xy|\vec{a}\vec{b}} = \left\langle \frac{1}{2}\big(I + (-1)^x\, \vec{a} \cdot (\mathsf{X}, \mathsf{Y}, \mathsf{Z})\big) \otimes \frac{1}{2}\big(I + (-1)^y\, \vec{b} \cdot (\mathsf{X}, \mathsf{Y}, \mathsf{Z})\big) \right\rangle_{\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)} \tag{3.59}$$

In our case, we will consider the ideal strategy to be that which arises when measuring $|\Psi^+\rangle$ EPR states:

$$p_{xy|\vec{a}\vec{b}} = \left\langle \frac{1}{2}\big(I + (-1)^x\, \vec{a} \cdot (\mathsf{X}, \mathsf{Y}, \mathsf{Z})\big) \otimes \frac{1}{2}\big(I + (-1)^y\, \vec{b} \cdot (\mathsf{X}, \mathsf{Y}, \mathsf{Z})\big) \right\rangle_{\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)} \tag{3.60}$$

Recall that the extended CHSH game does not allow the verifier to simultaneously characterise all three Pauli observables, for the two provers. By this we mean that there isn't one isometry, such that Alice and Bob's observables are isomorphic to X, Y and Z. Instead, one can find an isometry such that one of the observables is only characterised up to a reflection with respect to the plane defined by the other two observables. The choice of $|\Phi^+\rangle$, allows one to characterise X and Z, while Y is determined up to a reflection. In our case, we choose $|\Psi^+\rangle$ so that we can characterise X and Y, while Z is determined up to a reflection. Of course, this choice is arbitrary, since we could just as well use the XZ version, as long as we consider an XZ FK protocol (i.e. the $|+_\theta\rangle$ states are substituted with analogous states from the XZ plane, the dummies are substituted with the eigenstates of the Y operator and the CZ operations are substituted with CNOTs)[5].

In Protocol 1, the verifier will perform a state tomography sub-protocol for each of the 6 CHSH games. We are primarily interested in the XY ones, where the verifier is attempting to certify the correct preparation of the resource state:

$$|+\rangle \otimes \left|+_{\pi/4}\right\rangle \otimes \left|+_{2\pi/4}\right\rangle \otimes \left|+_{3\pi/4}\right\rangle \otimes \left|+_{4\pi/4}\right\rangle \otimes \left|+_{5\pi/4}\right\rangle \otimes \left|+_{6\pi/4}\right\rangle \otimes \left|+_{7\pi/4}\right\rangle \tag{3.61}$$

To keep things symmetric, we will assume that she also certifies the analogous states for the other games. In particular, for the XZ games, this will include:

$$|0\rangle \otimes |1\rangle \tag{3.62}$$

For each CHSH game and state tomography sub-protocol, we can apply Theorem 7. It was shown in [31] (see Theorem 6.21), that doing so leads to the following result. Provided that the verifier accepts with probability at least $1 - O(n_g^{-1/3})$

[5] Another way to view things is to note that labels like X, Y, Z are arbitrary. Ultimately, we are interested in the group theoretic properties of these operations.

then, with probability at least $1 - O({n_g}^{-1/48})$, it is the case that:

$$TD\left(\rho_S(O_{1,n_g}), \bigotimes_{j \in S} \pi^{O_j}\right) \leq O(n_g^{-1/64}) \tag{3.63}$$

Where S is a uniformly random subset from the set $[n]$, of size $O(n^{1/64})$ and $\rho_S(O_{1,n})$ is prover 1's state (up to an isometry), reduced to the subset S and conditioned on outcomes $O_{1,n}$. This is the soundness condition of our lemma. Completeness also follows from Theorem 7.

Finally, note that in the state tomography sub-protocol, the number of CHSH games that the verifier runs with the two provers is Nn_g, where $n_g = O(|\mathcal{C}|^{64})$ and $N = O(|\mathcal{C}|^{64(\alpha-1)})$, for some $\alpha > 32$. Overall this means that the round complexity of the protocol will be $O(|\mathcal{C}|^c)$, for some $c > 64 \cdot 32 = 2048$. □

We are now able to prove of our main result:

Proof of Theorem 5. In the verified preparation stage of the protocol, the verifier certifies the preparation of the input state for the FK protocol. Assuming that this consists of n qubits (where $n = O(|\mathcal{C}|)$, and $\mathcal{C}$ is the quantum computation that the verifier wishes to delegate), Lemma 5 tells us that that state will be $O(n^{-1})$ close to ideal (since, recall that $n_g = O(n^{64})$). Then, in the verified computation stage, the verifier performs the FK protocol, which we have shown is robust to deviations in the input state.

The completeness of this protocol will be inherited from the modified state tomography sub-protocol (since FK has completeness 1 in the ideal case). In other words, in the ideal case, the verifier accepts with probability at least $1 - O({n_g}^{-1/2}) = 1 - O(n^{-32})$. For soundness, we will have that, as long as the verifier accepts in the preparation stage with probability at least $1 - O(n^{-64/3})$, then she accepts an incorrect outcome of the computation with probability at most $O(n^{-64/48} \cdot n^{-2}) = O(n^{-10/3})$. The n^{-2} term comes from the soundness of the FK protocol when run with an input that is $O(n^{-1})$-close to ideal. □

3.4 Chapter summary and outlook

We have shown that the single server universal verifiable blind quantum computing protocol of Fitzsimons and Kashefi is robust with respect to general deviations in the quantum input state. We did this by first proving robustness with respect to deviations that are uncorrelated with the prover's private system and then generalising to the case of arbitrary deviations. In the former case, one could express the action of the protocol as a CPTP map which preserves distances. In the latter case, this is no longer true since the presence of initial correlations makes the action of the protocol a non-CPTP map. This general map differs from a CPTP map by a so-called inhomogeneous term. However, provided that the input state is ϵ-close to the ideal FK input, we showed that this inhomogeneous term has norm bounded by $O(\sqrt{\epsilon})$. Therefore, the contribution of this term can be made arbitrarily small, by reducing ϵ. This result complements the *local*-verifiability proof of [103] which is based on the universal composability framework.

Robustness, together with the rigidity result of Reichardt Unger and Vazirani, allowed us to come up with a device-independent verification protocol involving a single quantum server and a verifier with an untrusted measurement device. The resulting protocol is essentially a composition of the entanglement-based RUV protocol and the prepare-and-send FK protocol. This composite protocol achieves a lower communication complexity than RUV. This is because it uses only the (modified) state tomography part of RUV. The communication complexity of the composite protocol is still far too high to allow for any practical implementation in the near future. However, the reason for this high round complexity is the state tomography subprotocol and therefore, any improvement on how to prepare the FK inputs will directly improve the efficiency of our composite protocol as well.

Chapter 4

Rigidity of EPR-steering correlations

Sheldon: Quantum physics makes me so happy.

— The Big Bang Theory, Season 5, Episode 20

In the previous chapter, we saw that a key component of the RUV protocol, which was also essential for our device-independent protocol, is the rigidity of non-local correlations. In this chapter, we will prove an analogous rigidity result for a type of correlations known as *EPR-steering* (or just steering) correlations. Along the way, we will also provide a self-testing result for Bell states using these correlations. We will then show that if the composite protocol, given in the previous chapter, uses EPR-steering correlations instead of non-local correlations, this leads to a protocol that is *one-sided device-independent*.

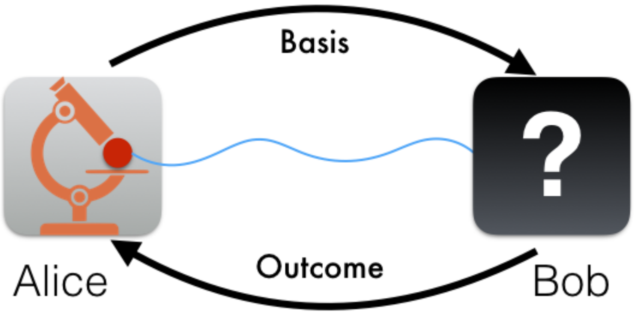


Figure 4.1: The basic steering setup.

EPR-steering correlations first appeared in the seminal paper of Einstein, Podolsky and Rosen [112] to support their argument that quantum mechanics is incomplete. In that paper, EPR never explicitly referred to these correlations as "steering" correlations. This name was given later, by Schrödinger, in [113], when he identified the defining feature of these correlations. To explain what that is, let us consider our favourite agents, Alice and Bob. Suppose that Alice has a single-qubit measurement device, which she trusts completely, and suppose that Bob sends Alice one qubit. Bob claims to be able to "steer" Alice's state.

What this means is that if Alice informs Bob that she wishes to measure her qubit in a particular basis, Bob can steer her state so as to predict her measurement outcome in that basis. He convinces Alice by telling her this outcome.

As an explicit example, suppose the qubit that Bob sent to Alice was half of an EPR state. Let us also assume that Alice wishes to test for steering in the eigenbases of X and Z. If she tells Bob to steer her state in the X basis, then Bob simply measures his half of the Bell pair using the X observable and sends the measurement outcome to Alice. Due to the properties of the Bell state, Alice, upon measuring her qubit with the X observable will obtain the same outcome as the one reported by Bob. Her state has been correctly steered in the X eigenbasis. Similarly, if she had chosen Z, then Bob would have measured the Z observable. Assuming both parties follow this protocol, Alice will always be convinced of Bob's steering abilities. Such correlations, between Bob's classical response and Alice's post-measurement quantum state, are known as steering correlations. This situations is shown schematically in Figure 4.1.

The study of quantum correlations has progressed a lot since the days of EPR and Schrödinger and this, in turn, has led to the development of numerous applications that make use of them. In particular, non-local correlations, apart from revealing counter intuitive features of nature and being tremendously important to quantum foundations, have led to the development of device-independent protocols for quantum key distribution (QKD), quantum random number generation (QRNG) and quantum verification [30–32, 34, 114–117]. As we've seen already with RUV and the device-independent protocol, non-local correlations allow one to characterise the behaviour of devices that are completely untrusted, based solely on the classical outputs of those devices. The characterisation involves determining that the devices are sharing a certain type of entangled state (such as EPR states) and are measuring them appropriately (for instance, with the Pauli observables). Such protocols are highly desired for practical implementation as they provide a higher level of security, unachievable by classical systems. However, there are certain practical issues that hinder their development, such as the need for high detection thresholds, high fidelity transmission channels, space-like separation and a high overhead [114, 118].

The practical limitations of device-independent protocols motivated the revival of research into quantum steering, due to its more relaxed trust assumptions. The existing research involves the characterisation of steering correlations both analytically and geometrically [119–121], their relationship to other types of correlations [122, 123] and their application to cryptographic tasks such as QKD and QRNG [124, 125]. Experiments testing *quantum steering inequalities* [126] (loophole-free) and testing local but steerable states [127] have also been performed. In the case of QKD, Branciard et al. showed in [124], that there is a natural correspondence between the trust assumptions of the protocol and the types of correlations between the two parties. Using this correspondence, Branciard et al. introduced *one-sided device-independent* QKD, which uses steering correlations in order to distil a shared secret key. In the cryptographic setting, such correlations allow only one device to be untrusted leading to a reduction in the overall experimental requirements of the protocol. To be precise, they showed that in typical device-independent settings, the detection efficiency of

Alice and Bob should be above 91.1%, whereas their one-sided protocol lowers that to 65.9%. A similar relation between trust assumptions and correlations is exploited for QRNG as well [125]. In this case it was shown that a detection efficiency of 50% is sufficient for random number generation in the steering setting, versus 70.7% in the device-independent setting.

For quantum verification, we will show that using steering correlations leads to a reduction in communication complexity, compared to the device-independent protocol. This comes as a direct result of having a *trusted* measurement device. As a technical observation, this one-sided device-independent protocol is a measurement-only verification protocol, according to the terminology defined in Chapter 2.

Verification takes place in a different setting than both QKD and QRNG. In the latter two, Alice and Bob are two parties that are working together towards a common objective (obtaining a shared key or certified randomness), using possibly untrusted quantum devices. But in the verification setting, Alice is a client who is delegating a difficult computation to Bob, an untrusted quantum server. Alice and Bob, in this case, are not collaborating, since the server is assumed to be malicious and attempting to deceive Alice. This is the standard cryptographic scenario when considering verification of computation, whether it is quantum or classical [23, 128]. Interestingly, the asymmetry in trust is similar to that of steering correlations. In the steering case, Alice interacts with Bob, whom she does not trust, and attempts to test his ability to steer her state. In the verification setting, Alice delegates a computation to Bob and attempts to test his ability to correctly perform that computation. One could thus argue that verification is a very natural application for steering correlations. While it is definitely possible to introduce such an asymmetry in QKD, for example, the traditional setting is to have the parties involved be identical in all respects. A broader discussion pertaining to the relationship between steering and verification can be found in [129].

This chapter is organized as follows. Section 4.1 is devoted to proving the rigidity result, which states that from observing maximal steering correlations between two parties, one can determine, up to a local isometry, that they share a tensor product of Bell pairs and that the untrusted party is performing the ideal measurements. To derive this result, we first prove the self-testing of Bell states and Pauli X and Y measurements[1] from steering correlations, in Subsection 4.1.1. Specifically, we show how from steering correlations one obtains a bound on the trace distance between the parties' shared state, and a perfect Bell pair. We also prove that this bound is optimal, up to constant factors.

It should be noted that we could also use existing self-testing results based on the CHSH game, rather than proving another self-testing result for Bell pairs. However, it is important to note that in the setting in which one party is trusted and the other is not it is not evident that a maximally entangled state can be self-tested from steering correlations as opposed to non-local correlations. Fur-

[1] The reason for choosing X and Y is to keep the same setup as in the device-independent case. Of course, this choice is arbitrary, as our result applies for any pair of anti-commuting two-outcome observables.

thermore, the steering-based self-test is simpler and highlights an important distinction with respect to non-local self tests: in the latter one needs to check that each party's observables anti-commute (thus showing that they are equivalent to the Pauli X and Z observables up to an isometry), whereas in the former this is not necessary. Knowing that one party is implementing the correct observables requires us to test only that the observables of the second party behave in the same way.

The self-testing result, on its own, is not of immediate practical use, since it involves exact probability distributions and expectation values (the so-called *infinite statistics regime*). For a realistic scenario, one should consider a finite number of observations. But this incurs two difficulties. The first is that we need to find a way to relate these finite statistics to the ideal quantum expectation values, that are used in the self-testing result. The second is that, in the most general case, we cannot assume that each observation is independent of the previous one, at least not for the untrusted party. Both of these aspects are addressed in Subsection 4.1.2. This latter result does not immediately yield the desired rigidity theorem, since there is a possibility that the characterised Bell states overlap[2]. To solve this problem we leverage the fact that the characterised states are Bell pairs as well as the fact that Alice is trusted, to derive a tensor product structure on Bob's system. This will lead to our rigidity result, which is described in Subsection 4.1.3.

Lastly, in Section 4.2 we use the rigidity result to construct a one-sided device-independent verification protocol (Subsection 4.2.1). We also show that in the verification setting, and specifically for the types of protocols we have considered, the required entangled states must be close to Bell pairs (Subsection 4.2.2).

4.1 State and strategy certification via steering

While quantum steering has been studied extensively in the context of verifying entanglement, it is important to elaborate on the subtle difference between verifying entanglement and verifying *maximal* entanglement and how this relates to the verification of quantum computations. It has already been shown that it is possible to certify, from steering correlations, that a state shared between two parties is entangled. In fact, this type of certification can be done in a fully device-independent way (under certain assumptions), and has been tested experimentally [130, 131]. However, it should be noted that these results use steering correlations as a *witness* for quantum entanglement. The purpose of a witness is to separate between entangled and non-entangled states and its existence is proven through the violation of a steering inequality, in analogy to a Bell inequality. In our setting, however, we do not simply require correlations that violate a steering inequality, rather, we require them to achieve close to their

[2] Overlapping means that the states share common degrees of freedom. As an example, suppose we have a state $|\psi\rangle$ of n qubits and a state $|\phi\rangle$ of m qubits. If the states have no overlapping qubits, then $|\psi\rangle |\phi\rangle$ is a state in a 2^{n+m}-dimensional Hilbert space. Otherwise, the joint state lives in a Hilbert space of dimension strictly less than 2^{n+m}, since there are shared degrees of freedom.

maximum possible value. This is analogous to the rigidity of non-local correlations, where one would check that CHSH correlations are close to their maximal quantum value, and similar to that case, this will enable us to certify the state and strategies used in producing those correlations.

The setting that we will consider is similar to the one used in [122, 124]. This involves two parties, Alice and Bob, where Alice has a trusted measurement device, while Bob has an untrusted measurement device. They share an unknown joint quantum state, $|\psi\rangle$, which, without loss of generality, can be assumed to be pure[3]. Alice instructs Bob to perform a measurement on their joint state. For example, if the shared state is $|\Psi^+\rangle$, Alice can instruct Bob to measure the Y observable on his qubit and report the outcome. The measurement steers Alice's qubit to a particular quantum state. She can then measure her state to confirm that her qubit was indeed steered to the expected state. This setup defines a *steering game*, in analogy to non-local games, mentioned in Chapter 2 (Subsection 2.1.3).

We know that non-local correlations are correlations that cannot be explained by a *local hidden variable model*. In keeping with the analogy to non-locality, in our case, the correlations between Alice and Bob's responses, given certain input questions, are steering correlations if they cannot be explained by a *local hidden state model*. Alternatively, in analogy to Bell inequalities, there are so-called *steering inequalities*. Whenever the expectation values of Alice and Bob's observables violate a steering inequality, this again indicates that Bob is able to steer Alice's state. Throughout this chapter, we will use the steering inequality as our "steering test" as opposed to local hidden state models. For a derivation of a similar self-testing result, using local hidden state models, see the independent result of Šupić and Hoban that appeared on the arXiv around the same time as our result [133].

Our main objective, in this section, is to show that if Alice and Bob violate a steering inequality close to maximally (i.e. up to order $O(\epsilon)$), this allows us to determine that their shared state is a tensor product of Bell pairs and that their measurements are close to the ideal Pauli measurements. The proof of this rigidity result goes as follows. First, building on the work regarding self-testing the singlet by McKague, Yang and Scarani [134], we derive a robust self-testing result of the $|\Psi^+\rangle$ Bell state from steering correlations. This will then allow us to characterise a tensor product of Bell pairs, under an i.i.d. assumption. We also show that the closeness bound, for self-testing the EPR state, is tight, up to constant factors. All of this is covered in Subsection 4.1.1.

Then, in Subsection 4.1.2, we remove the i.i.d. assumption by modelling the measurement process as a martingale and using the Azuma-Hoeffding inequality [135, 136], as is also done in [34, 116]. The way in which we remove the i.i.d. assumption is not specific to steering and can be applied to the non-local setting as well, thus complementing the work of McKague, Yang and Scarani. Our rigidity result then follows from the non-i.i.d. characterisation and ideas inspired

[3] To be more precise, if $\mathcal{H}_A$ is Alice's Hilbert space and $\mathcal{H}_B$ is Bob's Hilbert space, their shared state, ρ, could be mixed. But one can always take such a mixed state to the Church of the Larger Hilbert space [132] (i.e. a purification of the state). Hence, we view the state $|\psi\rangle$ as belonging to $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$, where $\mathcal{H}_C$ is a space associated with the purification of ρ.

from [31]. This is covered in Subsection 4.1.3.

Throughout this chapter we denote $|| \, |\psi\rangle \, || = \sqrt{\langle \psi \, | \, \psi \rangle}$ as the l^2-norm of a state $|\psi\rangle$. Additionally, for the trace distance of pure states we will often write $TD(|\psi\rangle \, , |\phi\rangle)$, to mean $TD(|\psi\rangle \, \langle\psi| \, , |\phi\rangle \, \langle\phi|)$.

4.1.1 Self-testing from steering correlations

Recall that in self-testing, we examine the correlations in the responses of players of a non-local game and from these correlations we determine two things:

- That their shared quantum state is close to some target quantum state.
- That they are performing local measurements on this state and their measurement operators are close to some target measurement operators (when acting on the target state).

What changes in the steering setting? As explained, the difference is that we will know, a priori, Alice's (the trusted party) measurement operators, as well as the local dimension of the quantum state she is measuring (or, in other words, the dimension of her Hilbert space $\mathcal{H}_A$). Thus, self-testing in the steering setting involves determining the shared state between her and Bob as well as Bob's measurement operators.

The target shared state that we wish to characterise will be the XY-determined $|\Psi^+\rangle$ Bell state, and the target observables for Bob will be the Pauli X and Y observables. The result can, of course, be generalized for any pair of anti-commuting observables and any Bell state. It should be noted, that our steering game will involve Alice and Bob measuring *only* Pauli X and Y and not observables such as $(\mathsf{X}+\mathsf{Y})/\sqrt{2}$ and $(\mathsf{X}-\mathsf{Y})/\sqrt{2}$, as one would consider for the CHSH game. This leads to an interesting observation: a Bell state has a local hidden variable model for Pauli basis measurements by both parties, but it does not have a local hidden state model. This highlights the difference between non-local and steering correlations and emphasizes the importance of trusting Alice's system in order to characterise the shared state and Bob's measurements.

We start by proving a theorem analogous to Theorem 1 of [134], allowing us to characterise the shared state of Alice and Bob, given bounds on the action of their observables on that state. Just as in [134], the primed observables denote untrusted operators. The shared state, which is also assumed to be untrusted is denoted as $|\psi\rangle$.

Theorem 8. *Suppose that from the observed correlations of measurements performed by Alice and Bob and knowing that Alice is measuring the* $\{\mathsf{X}, \mathsf{Y}\}$ *observables (denoted* X_A, Y_A*), one can deduce the existence of local observables* $\{\mathsf{X}'_B, \mathsf{Y}'_B\}$ *on Bob's side, with eigenvalues* ± 1*, which act on a bipartite state* $|\psi\rangle$ *such that:*

$$\begin{aligned} ||(\mathsf{X}_A - \mathsf{X}'_B)\, |\psi\rangle \, || &\leq \gamma_1, && (4.1)\\ ||(\mathsf{Y}_A - \mathsf{Y}'_B)\, |\psi\rangle \, || &\leq \gamma_1, && (4.2)\\ ||(\mathsf{X}'_B\mathsf{Y}'_B + \mathsf{Y}'_B\mathsf{X}'_B)\, |\psi\rangle \, || &\leq \gamma_2. && (4.3)\end{aligned}$$

for some $\gamma_1, \gamma_2 \geq 0$. Then there exists a local isometry $\Phi = I \otimes \Phi_B$ and a state $|junk\rangle_B$ such that

$$\big|\big|\Phi(M_A N'_B |\psi\rangle) - |junk\rangle_B M_A N_B |\Psi^+\rangle_{AB}\big|\big| \leq \varepsilon \tag{4.4}$$

with $M_A, N_B \in \{I, \mathsf{X}, \mathsf{Y}\}$, $N'_B \in \{I, \mathsf{X}'_B, \mathsf{Y}'_B\}$, $\varepsilon = 3\gamma_1 + \gamma_1^2/4 + 2\gamma_2$.

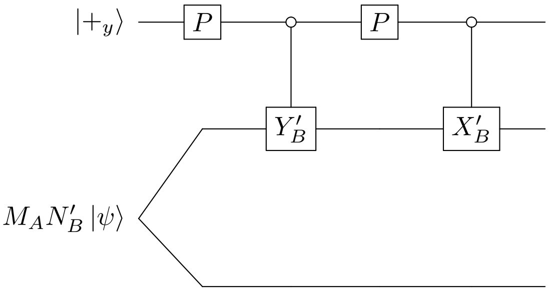


Figure 4.2: The local isometry $\Phi = I_A \otimes \Phi_B$.

Proof sketch. The proof relies on finding an isometry which, given conditions (4.1-4.3), maps $|\psi\rangle$ to an almost perfect Bell state. Similar to [134], M_A and N'_B are the physical observables of Alice and Bob which act on the shared state. The isometry we considered is illustrated in Figure 4.2, where $P = \frac{1}{\sqrt{2}}(\mathsf{X} + \mathsf{Y})$ and the control gates act on the target state when the control qubit is in the $|-_y\rangle$ state instead of the $|1\rangle$ state, and act as identity when the control is in the $|+_y\rangle$ state instead of the $|0\rangle$ state. Here, $|+_y\rangle$ and $|-_y\rangle$ are the two eigenstates for the Pauli Y operator, corresponding to the $+1$ and -1 eigenvalues, respectively. The fact that we are using these states and the P operator is a consequence of shifting everything to the XY-plane of the Bloch sphere instead of the more familiar XZ-plane. It should be noted that M_A is trusted and acts on Alice's part of the shared state, whereas N'_B, acting on Bob's part of $|\psi\rangle$, is untrusted. However, the action of N'_B is equivalent to the honest N_B acting on the ancilla introduced by Φ. Having the isometry, we write out its action on the state $M_A N'_B |\psi\rangle$ and use inequalities (4.1-4.3) together with the trace preserving properties of the operators and triangle inequalities to prove condition 4.4. The full proof of Theorem 8 can be found in Section 4.3. □

Our next result is to show that conditions (4.1-4.3) are satisfied if an almost maximal violation of a particular steering inequality occurs. As mentioned before, the requirement for maximal violation is in contrast to previous work on entanglement detection. In that case, one uses the steering inequality as an entanglement witness to separate the space of correlations into steering correlations and their complement. Violating the inequality determines that the shared state can produce steering correlations and is therefore entangled. For example, similar to the works of [137, 138], assuming Bob measures local observables X'_B and Y'_B, one could consider the inequality:

$$|\langle\psi| \mathsf{X}_A \mathsf{X}'_B + \mathsf{Y}_A \mathsf{Y}'_B |\psi\rangle| \leq \sqrt{2} \tag{4.5}$$

This inequality holds, whenever there is a local hidden state model for Bob's system. If this is not the case, then the state is steerable. In our case, we do not simply require a violation of this inequality, we require a (close to) maximal violation. The maximum value that the left-hand side of 4.5 can take is 2 and this is both the algebraic maximum as well as the maximum achievable in quantum mechanics. When this quantity is close to its maximum, we say that the steering inequality is being saturated and we write that as:

$$|\langle\psi| X_A \mathsf{X}'_B + \mathsf{Y}_A \mathsf{Y}'_B |\psi\rangle| \geq 2 - \epsilon \tag{4.6}$$

for some $\epsilon \geq 0$.

Before giving the self-testing result, let us clarify the game that we wish Alice and Bob to play. A referee will instruct Alice and Bob to either both measure the X observable on their local systems or to both measure the Y observable, and report their outcomes. They win the game if their outcomes match in both cases[4]. Assume that the referee chooses between the two situations with equal probability. In that case, the steering inequality 4.5 expresses the fact that any local strategy of Alice and Bob can succeed with probability at most $1/\sqrt{2}$. On the other hand, it is clear that if Alice and Bob share a Bell state and perform the instructed measurements, then their winning probability is 1. What our self-testing result shows is that the converse is also true, in a robust sense. In other words, if Alice and Bob win with probability close to 1, then, up to an isometry, their shared state is close to a Bell pair and Bob's observables are close to the ideal X and Y observables. Thus, our self-testing result is the following:

Theorem 9. *Suppose Alice measures the observables X_A, Y_A and that Bob measures the observables X'_B and Y'_B with eigenvalues ± 1, on the state $|\psi\rangle$, such that*

$$|\langle\psi| (\mathsf{X}_A \mathsf{X}'_B + \mathsf{Y}_A \mathsf{Y}'_B) |\psi\rangle| \geq 2 - \epsilon \tag{4.7}$$

where $0 \leq \epsilon \leq 1$. Then the conditions of Theorem 8 are satisfied with $\gamma_1 = \sqrt{2\epsilon}$ and $\gamma_2 = 4\sqrt{\epsilon}$.

Proof sketch. The proof entails expanding the left-hand side of inequality 4.7 and using the properties of the observables, to arrive at the bounds (4.1-4.3) from the previous theorem. More specifically, we see that the correlation of local observables that we consider:

$$|\langle\psi| (\mathsf{X}_A \mathsf{X}'_B + \mathsf{Y}_A \mathsf{Y}'_B) |\psi\rangle| \tag{4.8}$$

is simply a sum of two expectation values which are upper bounded by unity (because the observables have ± 1 eigenvalues). Hence, to saturate the absolute value of this quantity, it must be the case that both expectation values are saturated i.e. lower bounded by $1 - \epsilon$ or upper bounded by $-1 + \epsilon$. We will only examine the first case since the second is analogous, so we will drop the absolute value of

[4] Alternatively, they win the game if their outcomes are opposite in both cases. This is because we are taking the absolute value of the left-hand side of inequality 4.5, so the outcomes can either both be correlated or both be anti-correlated.

the expression and simply consider:

$$\langle\psi| \mathsf{X}_A\mathsf{X}'_B + \mathsf{Y}_A\mathsf{Y}'_B |\psi\rangle \geq 2 - \epsilon \tag{4.9}$$

By expressing each expectation as an l^2-norm we arrive at conditions 4.1 and 4.2. To prove condition 4.3 we use the Cauchy-Schwarz inequality and the commutators $[\mathsf{X}_A, \mathsf{Y}_A]$ and $[\mathsf{X}'_B, \mathsf{Y}'_B]$, respectively. The full proof can be found in Section 4.4. □

Our self-testing result shows that if Alice and Bob saturate the steering inequality up to order ϵ, their shared state is certain to be $O(\sqrt{\epsilon})$-close to a Bell pair and Bob's observables are $O(\sqrt{\epsilon})$-close to the ideal X, Y observables, up to the local isometry $\Phi = I_A \otimes \Phi_B$. The exact bound for closeness can be computed by simply inserting the constants γ_1 and γ_2 from Theorem 9 in the calculation for ε of Theorem 8. This yields a distance of $\varepsilon = (3\sqrt{2} + 8)\sqrt{\epsilon} + \epsilon/2$.

The same asymptotic bound of $O(\sqrt{\epsilon})$ is achieved in the case of the CHSH game. One could expect that, unlike the case of non-local correlations, where both parties are untrusted, in the steering case it should be possible to obtain a tighter bound for the shared state and Bob's observables. We prove that this is not the case and that the $O(\sqrt{\epsilon})$ bound is tight:

Theorem 10. *Suppose that Bob's observables X'_B and Y'_B with eigenvalues ± 1, acting on a state $|\psi\rangle$, are such that:*

$$|\langle\psi| (\mathsf{X}_A\mathsf{X}'_B + \mathsf{Y}_A\mathsf{Y}'_B) |\psi\rangle| \geq 2 - \epsilon \tag{4.10}$$

where $0 \leq \epsilon \leq 1$. Then, up to constant factors, the bound of Theorem 8 (i.e. inequality 4.4 with $\varepsilon = O(\sqrt{\epsilon})$) is tight.

Proof sketch. The proof relies on finding a shared state and local observables for Bob such that inequality 4.7 holds, but the state is $\Theta(\sqrt{\epsilon})$-close to an ideal Bell state. We provide such a state, which is exactly $\sqrt{\epsilon}$-close to the ideal $|\Psi^+\rangle$ Bell state. We also provide observables for Bob, that are $\sqrt{\epsilon}$-close to the ideal observables, and which, together with the shared state, achieve the desired correlation of $2 - \epsilon$. The specific state and Bob's local observables are given in Section 4.5. □

An important corollary to the three theorems we have stated, is the following:

Corollary 3. *The results of Theorems 8, 9 and 10 hold for any pair of anti-commuting observables, V and W, having eigenvalues ± 1 (instead of X and Y).*

Proof. In the proof of Theorem 8 we only made use of the anti-commutation properties of the X, Y observables on Alice's side as well as the action of the two operators on the eigenstates of Y. For general observables, V and W this translates to using their anti-commutation properties and the action of the two on the eigenstates of W, for example. Essentially the proof of Theorem 8 only changes by relabelling X_A as V_A and Y_A as W_A. On Bob's side, the situation is similar. By relabelling X'_B as V'_B and Y'_B as W'_B we again have conditions (4.1-4.3) for these observables, which are then used to construct the isometry and prove the result of Theorem 8. Of course, instead of the $|\Phi^+\rangle$ Bell state, we will simply have a different Bell state. To be more precise, there exists a unitary U, such that $U\mathsf{X}U^\dagger = V$ and $U\mathsf{Y}U^\dagger = W$, up to an isometry. The Bell state determined by V and U will be $|\eta\rangle = U \otimes U |\Phi^+\rangle$.

For Theorem 9, using the same relabelling we have the inequality:

$$|\langle\psi| V_A V_B' + W_A W_B' |\psi\rangle| \geq 2 - \epsilon \tag{4.11}$$

Since the proof of Theorem 9, like Theorem 8, relies only on the anti-commutation properties and the action of the observables on the eigenstates of one of them, the relabelling does not change the results.

Lastly, for Theorem 10, the $\sqrt{\epsilon}$-close state we consider will be the same as in Theorem 10 rotated by $U \otimes U$. This state will be $\sqrt{\epsilon}$-close to $|\eta\rangle$ and also satisfy inequality 4.11. Similarly, we will consider Bob's observables from Theorem 10 and conjugate them by U. □

The self-testing result (Theorem 9) assumes ideal expectation values for the observables of Alice and Bob. Of course, in practice, after performing a finite number of measurements we merely obtain an estimate of these expectation values. The closeness of this estimate to the true expectations values can be determined by modelling the measurement process using independent and identically distributed (i.i.d.) random variables and using a Chernoff bound. We do not give a full derivation of this here, since we will treat the more general case of non-i.i.d. random variables in the next section (for which the proof can be found in Section 4.6). Instead, we simply state the result: for a fixed $\epsilon > 0$, we require at least $(1/\epsilon^2)log(1/\epsilon)$ measurements in order to certify that the closeness of each shared state is $O(\sqrt{\epsilon})$ to a perfect Bell pair. One can also compute the number of measurements as a function of the desired trace distance for the Bell states. If we denote this distance as $D = c\sqrt{\epsilon}$, then the number of measurements must be at least $(2c^4/D^4)log(c/D)$. In our case $c \approx 12.3$, so that if we wanted the trace distance to be, for instance, $D = 0.1$, we would require at least 2.2×10^9 measurements.

4.1.2 Finite statistics in the non-i.i.d. case

In this subsection, we will consider the case of having a finite number of observations in our steering scenario, without assuming independence for the random variables that model the measurement process. The following theorem essentially states that if Alice and Bob are asked to perform several rounds of measurement, in sequence, and if we notice a close to maximal steering inequality violation from their outcomes, then we can conclude that the state shared in a typical round of measurement is close to a Bell pair. By "typical round" we mean a round that is chosen uniformly at random. A similar result is obtained in [139], for non-local rather than steering correlations, and with the additional difference that their result shows that at least one state (as opposed to a typical state) is close to an ideal Bell pair.

From now on, we will assume that Alice is the one instructing Bob on what to measure, rather than having a referee instruct both of them. This is because, since Alice is trusted, we might as well have her take on the role of referee as well. Additionally, to simplify the notation we will be using, we will be denoting Alice and Bob's observables as A_0, A_1 and B_0, B_1, respectively. The correlator we will be interested in becomes:

$$| \langle A_0 B_0 + A_1 B_1 \rangle | \tag{4.12}$$

The notation $Tr_{-i}(\cdot)$ indicates that we are tracing out everything apart from the quantum states that are measured in round i. We also use the notation $Tr_{-R}(\cdot)$, which generalizes the previous notation for a set, R, of rounds (i.e. tracing out all states except those which are used in rounds $i \in R$).

Theorem 11. *Suppose Alice and Bob are required to perform K rounds of measurement and also that:*

- *The initial shared state of Alice and Bob, prior to the K rounds of measurement, is denoted σ*[5]*.*
- *Alice chooses a random set of size $K/2$, consisting of distinct indices from 1 to K and denoted $R_0 = \{i | i \in_R \{1, ...K\}\}$, $|R_0| = K/2$. We also denote $R_1 = \{1, ...K\} \backslash R_0$, to be the complement of R_0. R_0 will consist of those rounds in which Alice should measure A_0 and Bob B_0, while R_1 will consist of those rounds in which Alice should measure A_1 and Bob B_1.*
- *We denote $\rho_i = Tr_{-i}(\mathcal{E}^{AB}_{1,i-1}(\sigma))$ to be the reduced state of Alice and Bob in round i (after the $i-1$ rounds of measurements), and:*

  $$\rho_{avg} = \frac{1}{K} \sum_{i=1}^{K} \rho_i \tag{4.13}$$

  to be the typical state that we are interested in. This is simply the average of all reduced states, for all rounds. Here, $\mathcal{E}^{AB}_{1,i-1}$ denotes the actions (projections resulting from measurements) of Alice and Bob on the state σ up to and excluding round i.
- *In round i, let $r_i = 0$ iff $i \in R_0$, otherwise $r_i = 1$. Alice measures the observable A_{r_i} on her half of ρ_i. A_0 and A_1 are anti-commuting single-qubit observables having ± 1 eigenvalues.*
- *In round i, Bob is asked to measure B_{r_i}. B_0 and B_1 have ± 1 eigenvalues*[6]*.*
- *We denote a_i and b_i, respectively, as the outcomes of their measurements in round i. We also denote $\hat{C}_i = a_i b_i$ as their correlation for round i.*

[5] In the ideal setting, in which Bob is honest and prepared the correct state, σ would be a $2K$-qubit state consisting of K Bell pairs.

[6] Note that Bob's observables B_0 and B_1 are not assumed to anti-commute. This will be derived, however, from steering correlations.

- *We denote $\hat{C}^0 = \frac{1}{K/2} \sum_{i \in R_0} \hat{C}_i$ and $\hat{C}^1 = \frac{1}{K/2} \sum_{i \in R_1} \hat{C}_i$ as the averaged correlations for the cases where both Alice and Bob are asked to measure the first observable, or both are asked to measure the second, respectively.*

If, for some given $\epsilon > 0$ and suitably chosen $K = \Omega((1/\epsilon^2) log(1/\epsilon))$, it is the case that $\hat{C}^0 + \hat{C}^1 \geq 2 - \epsilon$ (or, alternatively, $\hat{C}^0 + \hat{C}^1 \leq -2 + \epsilon$) then there exists a local isometry $\Phi = I_A \otimes \Phi_B$ and a state ρ_{junk}, on Bob's system, such that:

$$TD(\Phi(\mathcal{O}_{AB}(\rho_{avg})), \hat{\mathcal{O}}_{AB}(\left|\Psi^+\right\rangle \left\langle\Psi^+\right|)\rho_{junk}) \leq O(\sqrt{\epsilon}) \tag{4.14}$$

where $\mathcal{O}^{AB}$ denotes the action of the I, A_0, A_1, B_0, B_1 operators and $\hat{\mathcal{O}}_{AB}$ is the analogous action of the ideal operators (i.e. $I, \mathsf{X}_A, \mathsf{Y}_A, \mathsf{X}_B, \mathsf{Y}_B$), as in Theorem 8.

Proof sketch. To prove this result, we first show that the average observed correlations $\hat{C}^0$ and $\hat{C}^1$ approximate the ideal quantum correlation for the averaged state, $Tr(A_0 B_0 \rho_{avg})$ and $Tr(A_1 B_1 \rho_{avg})$. The averaged state can be thought of as the state shared by Alice and Bob in each round of measurements, such that the average correlations of outcomes from this state match those observed in the real experiment (i.e. $\hat{C}^0$ and $\hat{C}^1$). Proving this step is done along similar lines to the approaches of [34, 116]. The measurement process of Alice and Bob is treated as a stochastic process with bounded increment, i.e. a martingale. The specific martingale we consider encodes the correlations of their measurement outcomes. While the individual measurements need not be independent, we can still prove that this observed correlation is, with high probability, close to the ideal quantum correlation. To do this, we use the Azuma-Hoeffding inequality for martingales [135, 136]. To ensure that the probability is indeed high, we need to take $K = \Omega((1/\epsilon^2) log(1/\epsilon))$. Having an estimate for the ideal quantum correlations then allows us to use Theorem 9 to show the closeness of the averaged state to an ideal Bell state. The full proof is given in Section 4.6. □

As in the i.i.d. case, for a fixed $\epsilon > 0$, we will require $\Omega((1/\epsilon^2) log(1/\epsilon))$ measurements to determine the closeness of a typical state to a perfect Bell pair up to order $O(\sqrt{\epsilon})$. One could ask whether we can use this result to conclude that the initial state σ is $O(K\sqrt{\epsilon})$-close to a tensor product of Bell pairs. The answer is no. First of all, our result only guarantees that a *typical* state is close to a Bell pair, not that the state used in each round is close to a Bell pair. Secondly, even if that were so, this would not guarantee that the Bell states determined in each round do not overlap on Bob's side. In other words, the qubits that Bob measures in round i could overlap with the qubits in round j, with $i \neq j$. In reality this cannot happen, however the result we have proven does not rule out this possibility. The determination of a tensor product structure of Bell pairs is done in the next section.

It should be noted that in the proof of Theorem 11, we did not use the fact that Alice is trusted except when applying the self-testing result (Theorem 9). Thus, a similar theorem can be proven in the case where both Alice and Bob are untrusted. In that case, one could simply use the self-testing results of [32, 134, 140, 141] for the i.i.d. setting, and then obtain a statement about the closeness of a typical state to the ideal one in the non-i.i.d. setting using our techniques. For example, if we were to use Theorem 2 from [134] we could once again establish from the measurement statistics that a typical state shared by Alice and Bob is close to a Bell state. This result completes the work of [134] for the non-i.i.d. setting.

It should additionally be noted that throughout this section we not only assumed that Alice's device is trusted but that it also measures the Pauli operators *exactly*. This could seem unreasonable from an experimental perspective, however note that any (fixed) deviation on Alice's measurement operator can be incorporated into ϵ. In other words, assume Alice's ideal operator is A and the deviated one is δA, such that:

$$TD(A \otimes B \left|\psi\right\rangle, \delta A \otimes B \left|\psi\right\rangle) < \delta \tag{4.15}$$

It is thus the case that the action of Alice's operators is δ-close to the action of the ideal operators which produce ϵ saturation. Hence, δ can be added to ϵ and viewed as a contribution to the total variation from maximal correlations. However, if such a deviation exists we should consider what happens when $\delta \leq \epsilon$ and when $\delta > \epsilon$, respectively. If $\delta \leq \epsilon$, then the error on Alice's device is smaller than the precision with which we wish to estimate the saturation of the correlations. Therefore, the saturation can still be considered of order $O(\epsilon)$ and the bounds on the states follow as in the ideal case. However, if $\delta > \epsilon$ then the saturation cannot be estimated within the desired precision. This means that there will be an intrinsic limitation on the determined closeness of the shared states, given by δ.

4.1.3 Rigidity of quantum steering

We now proceed to prove rigidity of quantum steering games. As mentioned, we will essentially have the same setting as in the non-i.i.d. case of Theorem 11, except we are interested in determining a tensor product of Bell states for the state shared by Alice and Bob. We do this by first defining a K-round steering game. We will then ask Alice and Bob to play N such games, and show that a state comprising of typical states from each of the K-round games is close to N Bell pairs.

Definition 24. *We say that a game consisting of players Alice and Bob is a K-round steering game with threshold $T \leq K$ iff the following conditions are satisfied:*

- *Alice and Bob share a joint unknown quantum state $\left|\psi\right\rangle$.*
- *The game has K rounds.*
- *In round i, Alice measures observable A_{r_i}, while Bob is instructed to measure B_{r_i}, with $r_i \in_R \{0, 1\}$*[7]*.*

[7] The notation $a \in_R S$ indicates that a was chosen uniformly at random from the set S. In essence, a is a random variable whose possible values are given by the elements of S, each occurring with probability $1/|S|$. In our case, r_i is chosen at random from $\{0, 1\}$ so that it is equally likely to be either 0 or 1.

- *Alice's measurement device is fully trusted to perform the correct measurement, moreover she has a complete characterisation of the device's Hilbert space (which is assumed to be 2-dimensional).*

- *Alice and Bob win round i iff their outcomes are identical.*

- *Alice and Bob win the game iff they win at least T rounds.*

We now define the correlation value of the game.

Definition 25. *Let W be the number of rounds that Alice and Bob win in a K-round steering game. The correlation value for the game is defined as the fraction W/K.*

It is useful to make the following observation: if we assume that Alice and Bob are measuring the same state, $|\phi\rangle$, in each round, then the correlation value of the game would be:

$$\frac{1}{2} \langle\phi| A_0 B_0 + A_1 B_1 |\phi\rangle \tag{4.16}$$

In general, however, this might not be the case, since Bob is free to use any state in each round. Instead, in accordance with Theorem 11, the correlation value of the game is an estimate for the correlation of the averaged state. We can now state our rigidity theorem:

Theorem 12. *Suppose Alice and Bob play $N > 0$ K-round steering games, each having correlation value at least $1 - \epsilon$, with $\epsilon > 0$ and $K = \Omega(1/\epsilon^2 log(1/\epsilon))$. Additionally, let σ be the initial state shared between Alice and Bob, R be a set consisting of N random indices, one from each of the K-round steering games and $\rho = Tr_{-R}(\sigma)$ to be the state shared by Alice and Bob in those rounds.*

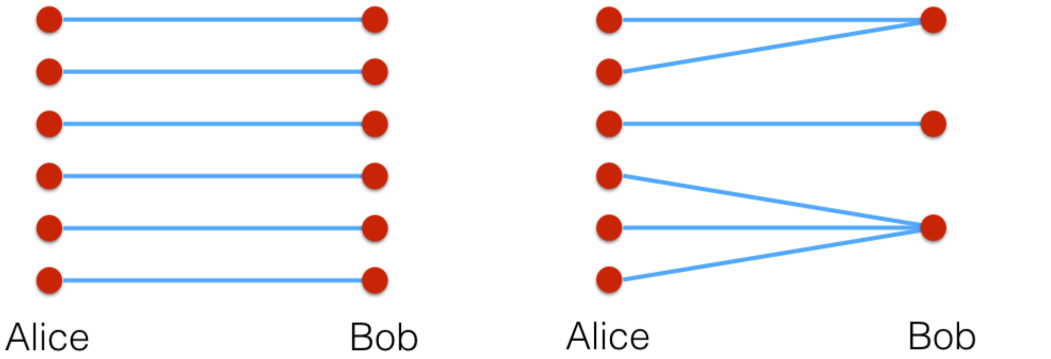
We then have that there exists a local isometry $\Phi = I_A \otimes \Phi_B$ and a state $|junk\rangle_B$ such that:

$$TD\left(\Phi(\mathcal{O}_{AB}(\rho)), \hat{\mathcal{O}}_{AB}\left(|\Psi^+\rangle_{AB}^{\otimes N}\right)|junk\rangle_B\right) \leq O(N\sqrt{\epsilon}) \tag{4.17}$$

where $\mathcal{O}^{AB}$ denotes the action of the I, A_0, A_1, B_0, B_1 operators and $\hat{\mathcal{O}}_{AB}$ is the analogous action of the ideal operators (i.e. $I, \mathsf{X}_A, \mathsf{Y}_A, \mathsf{X}_B, \mathsf{Y}_B$), as in Theorem 8.

Proof sketch. First, note that the typical state of each K-round steering game will be a state of the form of ρ_{avg} from Theorem 11, which we know is $O(\sqrt{\epsilon})$-close to a Bell state (additionally, Bob's measurement operators acting on that state are close to the ideal operators). As mentioned, while we have N such states, we cannot immediately argue that ρ is close to a tensor product of Bell states, since the qubits on Bob's side could overlap (see Figure 4.3). We want to show that this cannot happen and that his state effectively consists of N halves of EPR states, in tensor product with some junk state.

There are a number of ways to show this. In the paper which is the basis for this chapter [33], we proved this result using a technique inspired from RUV. The idea of that technique is to show that the strategy that Alice and Bob employ in the N games is close to a strategy in which Bob is performing the



(a) Ideal case in which Bob's qubits are not overlapping. Alice and Bob share a tensor product of Bell states.

(b) Some of Bob's qubits are overlapping. Alice and Bob are not sharing the intended state.

Figure 4.3

ideal measurements, which, in turn, is close to the ideal strategy in which Bob performs the ideal measurements *and* the shared state is $|\Psi^+\rangle_{AB}^{\otimes N}$. However, that technique is unnecessarily complicated for the steering setting, in which we have a full characterisation of Alice's system and her operators. We can therefore give a simpler proof that makes use of this fact.

The proof goes as follows. We know that for each typical state in a K-round steering game, the shared state of Alice and Bob is close to a Bell pair. Take the shared state of all selected rounds, ρ. We know that $Tr_{-i}(\rho) \approx |\Phi^+\rangle \otimes \rho_{junk_i}$, for each round i, and for some ρ_{junk_i} on Bob's system. Using the fact that Alice's Hilbert space is in tensor product form across the different rounds $\mathcal{H}_A = \mathcal{H}_{A_1} \otimes \mathcal{H}_{A_2} \otimes ... \otimes \mathcal{H}_{A_N}$ allows us to apply a version of the Gentle Measurement Lemma on her and Bob's systems. This allows us to show that their shared state is $O(N\sqrt{\epsilon})$-close to $|\Psi^+\rangle_{AB}^{\otimes N}$, up to a local isometry on Bob's system. The proof is provided in Section 4.7.

Let us also comment on two other alternatives for proving this result. The first is to use the results on overlapping qubits by Chao et al. [142, 143]. For a typical round, Bob's measurement operators will be close to Alice's ideal measurement operators on the shared state. We also know, since we have a characterisation of Alice's system, that her operators anti-commute when acting on the same qubit and commute with all operators acting on different qubits. Therefore, Bob's operators will behave the same way, up to order $O(\sqrt{\epsilon})$. But having a system of N pairs of such operators (X and Y), characterises a system of N qubits. One can then show that the state shared by Alice and Bob is $O(N\sqrt{\epsilon})$-close to a tensor product of Bell pairs, up to a local isometry on Bob's system.

The second approach (suggested by Matty Hoban) is the following. Let $\rho_{A_1A_2B}$, be the state of Alice and Bob for two chosen rounds, each from a separate K-round steering game. We know that ρ_{A_1B} is close to a Bell pair and we also know that ρ_{A_2B} is close to a Bell pair. Additionally, we know that $\mathcal{H}_{A_1} \otimes \mathcal{H}_{A_2} = \mathbb{C}^4$. The idea is to then consider ρ_{A_1B} to be the Choi state associated to a channel $\Theta_1 : \mathcal{D}(\mathcal{H}_B) \to \mathcal{D}(\mathcal{H}_{A_1} \otimes \mathcal{H}_{junk_1})$ and ρ_{A_2B} to be the Choi state associated to a channel $\Theta_2 : \mathcal{D}(\mathcal{H}_B) \to \mathcal{D}(\mathcal{H}_{A_2} \otimes \mathcal{H}_{junk_2})$. Since the Choi states are close to Bell pairs, together with some junk states, the channels Θ_1 and Θ_2 will be close to channels that act as identity on the A_1 and A_2 systems, while acting as preparations of fixed states on the junk systems. This would then imply that the channel associated to $\rho_{A_1A_2B}$, $\Theta_3 : \mathcal{D}(\mathcal{H}_B) \rightarrow \mathcal{D}(\mathcal{H}_{A_1} \otimes \mathcal{H}_{A_2} \otimes \mathcal{H}_{junk_3})$ would also act as identity on the A_1A_2 system. But now, if $dim(\mathcal{H}_B) < 4$, then this channel can be used to implement an approximate cloning map (from B to A_1A_2). For sufficiently small ϵ, this would be ruled out by a robust version of the no-cloning theorem [144]. This would then imply that the state $\rho_{A_1A_2B}$ must be close to two Bell pairs, together with an additional junk system. One would then apply this argument inductively to recover the tensor product of N Bell pairs. □

What can one say about the optimality of this closeness bound? In the single-state case, we were able to show that the bound obtained for our self-testing result is optimal, up to constant factors. Is this also true for the rigidity result? We conjecture that the answer is yes. To prove this, we would have to find a state ρ such that the reduced state in each round is $O(\sqrt{\epsilon})$-close to a Bell pair, while ρ itself is $O(N\sqrt{\epsilon})$-close to a tensor product of N Bell pairs. The natural choice is to take N approximate Bell states, i.e. a tensor product of N states, such that each state is $O(\sqrt{\epsilon})$-close to a Bell pair. However, this is not good enough. One can show that this state is, in fact, $O(\sqrt{N\epsilon})$-close to a tensor product of N Bell states. It would therefore seem that if the state that we are looking for exists, it would have to be entangled across the N rounds.

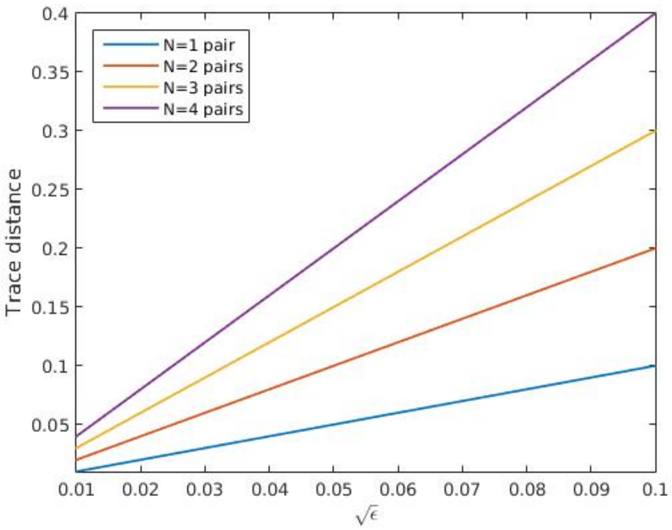


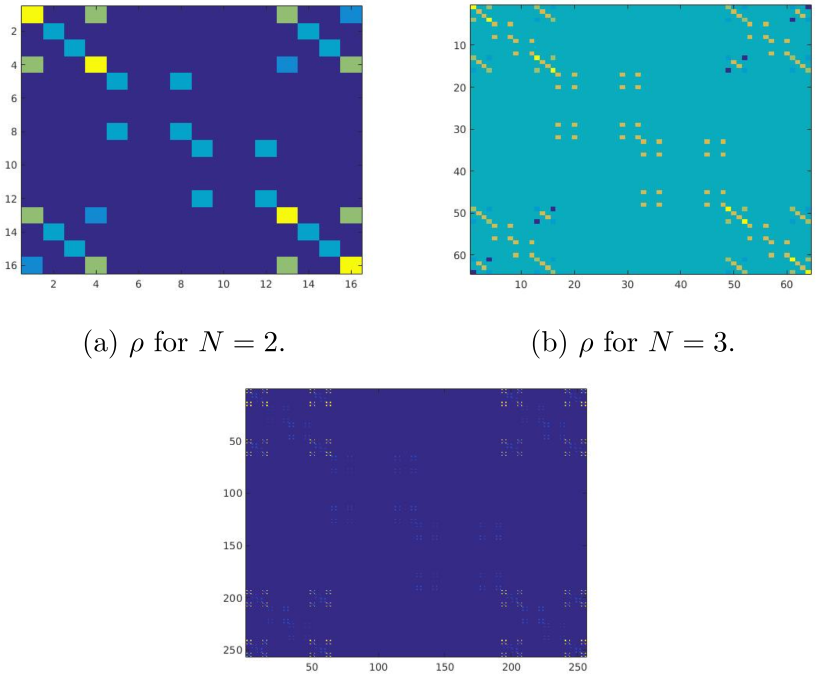
Figure 4.4: Obtained trace distance between ρ and a tensor product of N Bell pairs, for $N = 1, 2, 3, 4$ and $\sqrt{\epsilon} \in [0.01, 0.1]$.

In trying to find such a state, we considered the following semidefinite program

(SDP):

$$\begin{array}{ll} \textbf{minimize:} & Tr\left(\rho \left|\Psi^{+}\right\rangle \left\langle\Psi^{+}\right|^{\otimes N}\right) \\ \textbf{subject to:} & Tr(\Pi_i \rho) \geq 1-\epsilon \qquad \text{for all } 1 \leq i \leq N \\ & Tr(\rho) = 1 \\ & \rho = \rho^{\dagger} \\ & \rho \succeq 0 \end{array} \tag{4.18}$$

where Π_i is a projection on a $|\Psi^+\rangle$ for the state in round i. We solved this SDP for the cases of $N = 1, 2, 3, 4$ Bell pairs. Larger values of N required more memory than was available on our local machine. Nevertheless, the results were clear. In all cases, the trace distance for the obtained state ρ was *exactly* $N\sqrt{\epsilon}$, for values of $\sqrt{\epsilon}$ ranging between $[0.01, 0.1]$. This is ilustrated in Figure 4.4. Moreover, in each case the state ρ had a particular structure hinting at a general form. We plotted the density matrices (for fixed $\sqrt{\epsilon} = 0.1$) in each of the three considered cases (see Figure 4.5). Each block in the image represents an entry of the density matrix, when expressed in the computational basis. Large entries are yellow to orange in color, whereas smaller entries are blue. Note the fractal-like pattern in the density matrix as we increase the number of states. The code for our SDP solver[8] is provided in Appendix A (Listing A.1). Based on these results, we conjecture that the $O(N\sqrt{\epsilon})$ bound is optimal, up to constant factors.



(a) ρ for $N = 2$.

(b) ρ for $N = 3$.

(c) ρ for $N = 4$.

Figure 4.5

Since the closeness of the N-round state to N Bell pairs is of order $O(N\sqrt{\epsilon})$, to

[8] In our SDP solver we considered the $|\Phi^+\rangle$ Bell state, instead of $|\Psi^+\rangle$, but this does not affect the validity of the results.

obtain a decreasing error, we require $\epsilon = O(N^{-2})$. We also know from Theorem 11 that given ϵ, the number of games required is of order $K = \Omega((1/\epsilon^2)log(1/\epsilon))$. Therefore, we must have that $K = \Omega(N^4 log(N))$. Since each steering game is comprised of K rounds, we have KN rounds in total, or $\Omega(N^5 log(N))$ rounds of steering games. This will be the preparation complexity for the quantum state used in the FK protocol.

4.1.4 Comparison with other approaches

Before presenting the quantum verification application of our rigidity result, we briefly compare our approach to similar results. The literature on self-testing has become quite vast, since the concept was introduced, so we will only mention results that are very closely related to ours. As mentioned, this paper builds on the work of self-testing the singlet of McKague et al [134]. While we use similar techniques to theirs, we assume that Alice is trusted and thus arrived at an improved bound for the closeness of $|\psi\rangle$ to an ideal Bell pair ($O(\sqrt{\epsilon})$ vs $O(\epsilon^{1/4})$). Of course, there are self-testing results, in the non-local case, achieving the same bound, such as [31] or [139]. Our optimality result (Theorem 10) applies to these as well, showing that one cannot improve the asymptotic closeness of this bound.

In terms of the exact bounds, we obtained $(3\sqrt{2}+8)\sqrt{\epsilon}+\epsilon/2$, which is smaller compared to that of [31], approximately $270\sqrt{\epsilon}$. The result of [139] obtained numerically an even smaller factor of $\sqrt{2.2\epsilon}$ by using a semidefinite program. Their technique could, in principle, be used to improve our approach as well. We also mention the result of Šupić and Hoban [133], which appeared concurrently with our own. They also consider the case of self-testing from steering correlations, obtaining an analytic bound of $13\sqrt{\epsilon}$ and a numeric bound of $1.19\sqrt{\epsilon}$.

Furthermore, [139] also considers removing the i.i.d. assumption. Their approach is based on hypothesis testing, however the end result is to show that *at least* one state, out of all measured states, was close to a perfect Bell pair. As previously specified, our Theorem 11 establishes that *a typical* state, out of all measured states, was close to a perfect Bell pair. This is necessary in order to prove the rigidity result and certify the tensor product structure of Bell pairs.

For the rigidity of steering correlations, in the publication that is the basis for this chapter [33], we employed an approach similar to that of [31], that relied on showing the closeness of measurement strategies. Here, however, we used the simpler approach of utilizing the Gentle Measurement Lemma and leveraging the tensor product structure of Alice's system. Our result, as well as other results for characterising N Bell states, such as [31, 143], are self-testing results in which ϵ is polynomial in $1/N$. In the terminology introduced in Chapter 2, these are self-tests with polynomial robustness. A recent result by Natarajan and Vidick [36] provides a self-test with constant robustness. This was used to develop entanglement-based verification protocols having quasi-linear communication complexity [36, 106]. The protocol from [106], of Coladangelo et al., has a similar structure to our device-independent verification protocol as well as the one-sided device-independent one (consisting of two stages, one for certifying entanglement and one for delegating a quantum computation), which we present in the next section.

4.2 Verification

4.2.1 Verification based on steering correlations

There are two ways in which we can construct our verification protocol. The first is identical to the device-independent case. Specifically, the verifier will first run a state tomography sub-protocol in order to check for the correct preparation of Bell states and then measure a subset of these states, to remotely prepare the input of the FK protocol, for the server. She would then run that protocol with the server, thus delegating the quantum computation.

The second approach is to have the verifier randomly choose between a testing phase, which certifies the tensor product of Bell states and the verified computation. To be more precise, the verifier will ask the prover to prepare the shared Bell states and then decide to either check this preparation, or measure her halves of the shared states so as to remotely prepare the FK state and then run the FK protocol. It might seem, at first, that this latter approach is not well suited for a practical implementation, the reason being that if the server has to prepare and send the shared state, *before* the verifier makes her choice, then she would have to store her half of the state for when she chooses how to measure it. Depending on whether she decides to test the state or run the computation, she would then measure the qubits appropriately and instruct the server to do the same. But storing the states would require her to have a quantum memory of the same size as the number of Bell pairs she wishes the prover to prepare.

Fortunately, this is not the case. The verifier can decide in advance whether she wants to test the state or run the computation and, irrespective of this, ask the prover to prepare the tensor product of Bell pairs. She will then request that the prover send the Bell state halves one at a time for the verifier to measure. Due to the no-signalling principle, this will not leak any information to the prover about the verifier's measurements. Then, depending on her choice, she will instruct the server to either perform the measurements for the testing phase or for the computation phase. Thus, in terms of practical requirements, the two approaches are identical. We will implement the second approach, since it is easier to analyse.

Taking the verifier to be Alice and the prover to be Bob, Alice will test Bob's preparation of the shared state by playing multiple steering games with him and relying on the rigidity result. In the state testing phase, if Alice observes a close to maximal saturation of the steering correlations, she can conclude that a suitably chosen subset of the shared states is close to a tensor product of Bell pairs. In the verified computation stage, she measures her qubits so as to prepare the FK input and then runs the FK protocol with Bob. All of this is encapsulated in Protocol 3.

As per Theorems 11 and 12, to remotely prepare N qubits in the prover's system, we require him to create $\Theta(N^5 log(N))$ Bell states. This essentially determines the communication complexity of the protocol. Contrast this, however, to the communication complexities in the fully device-independent case, which, as we've seen, are on the order of $O(N^c)$, $c > 8192$ for RUV and $c > 2048$ for our protocol, respectively. Of course, we should stress two things:

(1) The communication overhead in the device-independent case follows from

Protocol 3 One-sided device-independent verification protocol

Assumptions: The verifier wishes to delegate a quantum computation $\mathcal{C}$ to the prover, such that $|\mathcal{C}| = M$. As in UBQC and the FK protocol, we will associate this computation with the graph state $|G\rangle$, comprising of $N = cM$ qubits, for some constant $c > 0$. The verifier's measurement device is trusted and able to perform single qubit measurements of the observables: X, Y, $(\mathsf{X}+\mathsf{Y})/\sqrt{2}$, $(\mathsf{X}-\mathsf{Y})/\sqrt{2}$, Z.

Protocol: The verifier fixes a constant δ, representing the trace distance she wishes for the FK input state, relative to the ideal state. The verifier asks the prover to prepare $kN^5 log(N)$ Bell pairs, where $k > 0$ is a constant determined by the results of Theorems 11 and 12. She views these as N blocks of $\Theta(N^4 log(N))$ states each. She instructs the prover to send her one qubit from each Bell pair. With probability 1/2 she chooses to perform one of the following:

State testing

1. For each qubit she receives, the verifier chooses, with probability 1/2, to measure it with either the X observable or the Y observable. After all qubits have been measured, she instructs the server to measure the corresponding qubit of each Bell state using the same observable and report all outcomes.
2. The verifier views each of the N blocks as a $\Theta(N^4 log(N))$-round steering game. She accepts in state testing, iff all games achieve a correlation value of at least $1 - \delta/kN^2$.

Verified computation

1. The verifier chooses uniformly at random, one qubit from each of the N blocks. She measures these qubits so as to remotely prepare the input state for the FK protocol (i.e. the qubits comprising $|G\rangle$, up to Pauli corrections).
2. She instructs the prover to discard all but the qubits that will be used for the FK protocol (determined by the previous step) and runs the FK protocol.

bounds that are not tight. It is entirely possible that if these bounds were optimized, we would obtain the same asymptotic scaling as in the steering case.

(2) As mentioned, both the device-independent case as well as the steering case are based on self-testing results having polynomial (and not constant) robustness. In fact the robustness of both self-tests are identical. Thus, the reduced overhead (at least in this non-optimized analysis) in the steering case is a direct result of trusting Alice's device. Of course, there is the recent Pauli braiding test of Natarajan and Vidick [36], which has constant robustness, and has been used to develop entanglement-based protocols having quasi-linear communication complexity [106]. Using that test in our protocols would, presumably, also lead to a quasi-linear overhead. In those cases, it would be interesting to see whether the steering case outperforms the device-independent case in constant factors.

There is another noteworthy difference between the device-independent protocol and the steering one. Recall that in the device-independent case, since the measurement device was untrusted, it was not sufficient to simply determine a tensor product structure of Bell pairs between it and the prover. We also had to ensure that the measurement device was measuring the correct observables. This was checked by performing an extended CHSH game, requiring the measurement of 9 observables. The reason for 9 observables was because we required the preparation of states both in the XY-plane as well as the XZ-plane and the extended CHSH game, which symmetrises all planes, allowed us to do this. In the steering case, however, since we trust the measurement device of the verifier it is sufficient to simply check for the correct preparation of Bell states. Having done so, Alice can simply perform the desired measurements so as to prepare the states used by FK.

We can now prove the following result for our protocol:

Theorem 13. *Protocol 3, for a computation of size N, has completeness 1, soundness $O(\sqrt{\delta})$, for some $\delta > 0$, fixed by the verifier, and communication complexity $\Theta(N^5/\delta^4 log(N/\delta))$.*

Proof. Let us first discuss the communication complexity. As we saw with Theorem 12, Alice will play N K-round steering games with Bob. To achieve closeness $\delta = kN\sqrt{\epsilon}$ for the tensor product of Bell pairs, it must be that $K = \Omega(1/\epsilon^2 log(1/\epsilon))$. Expressing ϵ as a function of N and δ we have that $\epsilon = \Theta(\delta^2/N^2)$, which leads to $K = \Theta((N/\delta)^4 log(N/\delta))$. Doing this N times, leads to our communication complexity of $\Theta(N^5/\delta^4 log(N/\delta))$.

For completeness, note that when Bob is honest and performing the instructed operations, both the state testing phase and the verified computation phase succeed with probability 1. For the state testing phase, this is because the steering game has quantum success probability 1. For the verified computation phase it is because the FK protocol has completeness 1.

In the case of soundness, we know that if the state testing phase succeeds, the FK input state will be δ-close to the ideal state. As we know from the robustness of FK, from Chapter 3, if the input state is δ-close to ideal, the soundness bound changes by at most $O(\sqrt{\delta})$[9]. □

[9] Technically, the soundness of the FK protocol is $(8/9)^d$, for some constant d, fixed by the verifier. This means that our soundness is in fact $(8/9)^d + O(\sqrt{\delta})$. However, in practice the $(8/9)^d$ factor can be made much smaller than $\sqrt{\delta}$, hence we are essentially only interested in the dependence on δ.

4.2.2 Verification from partially entangled states

We have seen that in both device-independent verification and one-sided device-independent verification we can characterise the tensor product structure of Bell pairs, between the verifier and the server, from observed correlations. More precisely, saturating an inequality involving correlations leads to a bound on the trace distance between the shared state and perfect Bell pairs, up to a local isometry. Is it, however, necessary to use Bell states? Could some other entangled state be useful? We know that for QKD and QRNG, one can use states that are not maximally entangled (or even close to being maximally entangled) [145–149] and so it is interesting to examine if this is also the case for verification. Determining this in full generality is a complicated matter. For entanglement-based protocols, we know that one can indeed use less than maximally entangled states. For instance, the verification protocol of McKague from [32] requires the provers to share a triangular cluster state, which is universal for measurement-based quantum computation. The verifier will either perform a self-test to check that the provers are indeed sharing this state or delegate a computation to them.

In the FK protocol, which is a prepare-and-send protocol, we know that the verifier needs to prepare 8 states that are evenly spread out around the XY plane plus the $|0\rangle$, $|1\rangle$ dummy states. Of course, the preparation of these states can be turned into a measurement as follows: suppose that the verifier has a trusted source that produces states of the form:

$$\rho_{CQ} = \frac{1}{10}\left(\sum_{\theta} |\theta\rangle\langle\theta| \otimes |+_\theta\rangle\langle+_\theta| + |0\rangle\langle 0| \otimes |0\rangle\langle 0| + |1\rangle\langle 1| \otimes |1\rangle\langle 1|\right) \tag{4.19}$$

This state is an equal mixture of the 10 possible states required in the FK protocol. The first register of each of these states specifies which state was prepared. Alice will therefore measure this register and send the qubit of the second register to Bob. Such states are called *classical-quantum (CQ)* states [150].

The ability to use CQ states for the FK protocol is conditioned on having a trusted source to prepare them. However, if the source is not trusted, then the verifier would need to test the preparation of these states. This is the setting we will consider. We are assuming an untrusted source (possibly the prover) that should prepare multiple copies of some bipartite state ρ_{AB}. The essential properties that we require of this state are that:

1. Alice, through local measurement on half of this state, can prepare a desired set of target states $\{\sigma_a\}_a$ on Bob's side, satisfying $\sum_a \sigma_a = Tr_A(\rho_{AB})$.

2. Alice has a single-qubit measurement device.

3. From Schmidt decomposition, condition 2 (or rather the fact that Alice's Hilbert space is two-dimensional) implies that ρ_{AB} is equivalent to a two-qubit state. We will require that Alice be able to prepare at least one pair of orthogonal states on Bob's side (such as $|0\rangle$ and $|1\rangle$), with equal probability. From condition 1 this is equivalent to $Tr_A(\rho_{AB}) = I/2$.

States that satisfy condition 1 are referred to as *completely steerable states.* Let us give their definition, from [151]:

Definition 26. *A bipartite state, ρ_{AB}, shared by Alice and Bob is completely steerable iff for any positive operators $\{\sigma_a\}_a$, satisfying $\sum_a \sigma_a = Tr_A(\rho_{AB})$, there exist a POVM $\{E_a\}$, such that $\sigma_a = Tr_A((E_a \otimes I)\rho_{AB})$.*

Condition 2 is simply a requirement stemming from practicality. Lastly, condition 3 is added since all existing prepare-and-send protocols require the verifier to prepare at least one pair of orthogonal states with equal probability [15]. We will refer to states that satisfy all three of these conditions as *totally steerable* states. For these states, we can show the following:

Theorem 14. *A bipartite state, ρ_{AB} is totally steerable iff ρ_{AB} is maximally entangled.*

Proof sketch. If ρ_{AB} is a pure state, then we are done since, given that ρ_{AB} is steerable, it must be entangled. But any pure entangled two-qubit state having $\rho_B = I/2$ is maximally entangled. We can show that ρ_{AB} is indeed a pure state. To do this, we consider a purification of this state and impose the listed constraints on the amplitudes of this purification, expressed in the computational basis. Solving the system of constraints leads to a density matrix which can easily be shown to correspond to a pure, maximally entangled state. See Section 4.8 for the proof. □

We leave as an open problem to determine whether maximally entangled states are required in more general verification settings.

4.3 Proof of Theorem 8

In this section we give a complete proof of Theorem 8 which characterises one-sided device-independent self-testing. Consider the following isometry:

$$\Phi(|\psi\rangle) = \frac{1}{2}(I + \mathsf{Y}'_B)\,|\psi\rangle\,|+_y\rangle + \frac{i}{2}\mathsf{X}'_B(I - \mathsf{Y}'_B)\,|\psi\rangle\,|-_y\rangle \tag{4.20}$$

An illustration of this isometry is given in Figure 4.2, where the upper part is Bob's system and the lower part is Alice's system. It should be noted that the control gates act on the target when the control qubit is in the $|-_y\rangle$ state, and act as identity when the control qubit is $|+_y\rangle$. This is in contrast to the standard convention in which the control is a computational basis state. Here $|+_y\rangle$ and $|-_y\rangle$ are the eigenstates of the Pauli Y operator and $P = \frac{1}{\sqrt{2}}(\mathsf{X} + \mathsf{Y})$. We can clearly see that $\Phi = I_A \otimes \Phi_B$, where Φ_B is determined by the combination of X'_B and Y'_B operators, from expression 4.20, which only act on Bob's system. We proceed to show that when conditions (4.1)-(4.3) are satisfied, we obtain condition 4.4. First, we show that:

$$||\Phi(M_A N'_B\,|\psi\rangle) - M_A N_B \Phi(|\psi\rangle)|| \leq 2\gamma_2 \tag{4.21}$$

Because M_A only acts on Alice's system, whereas the isometry is local on Bob's system, M_A trivially commutes to the left so that $\Phi(M_A N'_B |\psi\rangle) = M_A \Phi(N'_B |\psi\rangle)$. Now consider the possible choices for N'_B. If $N'_B = I$, the relation holds trivially. If $N'_B = \mathsf{Y}'_B$, since Y'_B is hermitian and unitary we have that:

$$\Phi(\mathsf{Y}'_B |\psi\rangle) = \frac{1}{2}(I + \mathsf{Y}'_B) |\psi\rangle |+_y\rangle - \frac{i}{2}\mathsf{X}'_B(I - \mathsf{Y}'_B) |\psi\rangle |-_y\rangle \tag{4.22}$$

At the same time, the ideal Pauli operator Y_B, acting on Bob's ancilla, has the following effect:

$$\mathsf{Y}_B\Phi(|\psi\rangle) = \frac{1}{2}(I + \mathsf{Y}'_B) |\psi\rangle |+_y\rangle - \frac{i}{2}\mathsf{X}'_B(I - \mathsf{Y}'_B) |\psi\rangle |-_y\rangle \tag{4.23}$$

This is because $\mathsf{Y} |+_y\rangle = |+_y\rangle$ and $\mathsf{Y} |-_y\rangle = - |-_y\rangle$ and we notice that the two expressions are identical. Lastly, when $N'_B = \mathsf{X}'_B$ we have:

$$\Phi(\mathsf{X}'_B |\psi\rangle) = \frac{1}{2}(I + \mathsf{Y}'_B)\mathsf{X}'_B |\psi\rangle |+_y\rangle + \\ \frac{i}{2}\mathsf{X}'_B(I - \mathsf{Y}'_B)\mathsf{X}'_B |\psi\rangle |-_y\rangle$$

And the action of the ideal operator yields:

$$\mathsf{X}_B\Phi(|\psi\rangle) = \frac{i}{2}(I + \mathsf{Y}'_B) |\psi\rangle |-_y\rangle + \frac{1}{2}\mathsf{X}'_B(I - \mathsf{Y}'_B) |\psi\rangle |+_y\rangle \tag{4.24}$$

This is because $\mathsf{X} |+_y\rangle = i |-_y\rangle$ and $\mathsf{X} |-_y\rangle = (-i) |+_y\rangle$. Using the approximate anti-commutation of X'_B and Y'_B, as given by condition 4.3, we notice that commuting X'_B to the left in $\Phi(\mathsf{X}'_B |\psi\rangle)$ will lead to the same expression as for $\mathsf{X}_B\Phi(|\psi\rangle)$ up to $2\gamma_2$ error. Thus:

$$||\Phi(M_A N'_B |\psi\rangle) - M_A N_B \Phi(|\psi\rangle)|| \leq 2\gamma_2 \tag{4.25}$$

We therefore, only need to examine the closeness of $\Phi(|\psi\rangle)$ to the ideal Bell state tensored with some junk state. Start by considering the state:

$$\begin{aligned} |\phi\rangle &= \frac{1}{2}(|\psi\rangle |+_y\rangle + \mathsf{Y}_A |\psi\rangle |+_y\rangle + \\ &+ i\mathsf{X}_A |\psi\rangle |-_y\rangle - i\mathsf{Y}_A\mathsf{X}_A |\psi\rangle |-_y\rangle) \end{aligned} \tag{4.26}$$

We will show that $\Phi(|\psi\rangle)$ and $|\phi\rangle$ are close in trace distance. Firstly, from conditions (4.1)-(4.2) using suitable triangle inequalities and the unitarity of operators X_A and Y_A which do not increase trace distance, it can be shown that:

$$||(\mathsf{X}'_B\mathsf{Y}'_B - \mathsf{Y}_A\mathsf{X}_A) |\psi\rangle || \leq 2\gamma_1 \tag{4.27}$$

Expanding the trace distance of $\Phi(|\psi\rangle)$ and $|\phi\rangle$ we have:

$$\begin{aligned} ||\Phi(|\psi\rangle) - |\phi\rangle\,|| = \frac{1}{2}||(\mathsf{Y}'_B - \mathsf{Y}_A)\,|\psi\rangle\,|+_y\rangle + \\ +i(\mathsf{X}'_B - \mathsf{X}_A)\,|\psi\rangle\,|-_y\rangle - i(\mathsf{X}'_B\mathsf{Y}'_B - \mathsf{Y}_A\mathsf{X}_A)\,|\psi\rangle\,|-_y\rangle\,|| \end{aligned}$$

And using the above results it follows that:

$$||\Phi(|\psi\rangle) - |\phi\rangle\,|| \leq 2\gamma_1 \tag{4.28}$$

Let us now rewrite $|\phi\rangle$. Given that we trust Alice's side of the $|\psi\rangle$ state, we can express it as follows:

$$|\psi\rangle = a\,|\alpha\rangle_B\,|+_y\rangle_A + b\,|\beta\rangle_B\,|-_y\rangle_A \tag{4.29}$$

Where $|a|^2 + |b|^2 = 1$ and the states $|\alpha\rangle$ and $|\beta\rangle$ are normalized. Here, the first part denotes Bob's system, for which we can make no assumptions, and the second part is Alice's qubit. The reason for choosing Pauli-Y eigenstates on Alice's side is to simplify the calculation. We could have expanded her system in any basis since a local unitary on her system does not change the result. Substituting this into the expression for $|\phi\rangle$ and labeling the ancillary qubit introduced by this isometry with Φ we get:

$$\begin{aligned} |\phi\rangle \;&=\; \frac{1}{2}(a\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A + b\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A) \\ &+\; \frac{1}{2}\mathsf{Y}_A(a\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A + b\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A) \\ &+\; \frac{1}{2}\mathsf{X}_A(a\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A + b\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A) \\ &-\; \frac{1}{2}\mathsf{Y}_A\mathsf{X}_A(a\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A + b\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A) \end{aligned} \tag{4.30}$$

Using the following identities:

$$\mathsf{X}\,|+_y\rangle = i\,|-_y\rangle \quad \mathsf{X}\,|-_y\rangle = -i\,|+_y\rangle \tag{4.31}$$

$$\mathsf{Y}\,|+_y\rangle = |+_y\rangle \quad \mathsf{Y}\,|-_y\rangle = -\,|-_y\rangle \tag{4.32}$$

We reduce $|\phi\rangle$ to:

$$\begin{aligned} |\phi\rangle \;&=\; \frac{1}{2}(a\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A + b\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A) \\ &+\; \frac{1}{2}(a\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A - b\,|+_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A) \\ &-\; \frac{1}{2}(a\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A - b\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A) \\ &-\; \frac{1}{2}(a\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|-_y\rangle_A + b\,|-_y\rangle_\Phi\,|\alpha\rangle_B\,|+_y\rangle_A) \end{aligned} \tag{4.33}$$

The terms with b coefficient cancel out and we are left with:

$$|\phi\rangle = a\,|\alpha\rangle_B \left(|{+_y}\rangle_\Phi |{+_y}\rangle_A - |{-_y}\rangle_\Phi |{-_y}\rangle_A\right) \tag{4.34}$$

This state is equivalent to:

$$|\phi\rangle = a\sqrt{2}\,|\alpha\rangle_B \left|\Psi^+\right\rangle_{AB} \tag{4.35}$$

We would like to equate this to $|junk\rangle_B |\Psi^+\rangle_{AB}$, however, the state we have is unnormalized unless $a = 1/\sqrt{2}$. We therefore compute a bound on $|a|$ to determine the error introduced by the unnormalized state. Condition 4.1 can be rewritten as:

$$1 - \gamma_1^2/2 \leq \langle\psi|\, \mathsf{X}_A \mathsf{X}'_B \,|\psi\rangle \tag{4.36}$$

By expanding $|\psi\rangle$, applying the operators X_A, X'_B and using the facts that $|a|^2 + |b|^2 = 1$ and that X'_B is hermitian and has ±1 eigenvalues, we obtain:

$$\sqrt{1 - \gamma_1^2/2} \leq a\sqrt{2} \leq \sqrt{1 + \gamma_1^2/2} \tag{4.37}$$

Since for small γ_1 we know that $\sqrt{1 - \gamma_1^2/2}$ approaches $1 - \gamma_1^2/4$ and $\sqrt{1 + \gamma_1^2/2}$ approaches $1 + \gamma_1^2/4$ we have that the norm of $|\phi\rangle$ can change from unity by an order of $\gamma_1^2/4$. Thus, it follows that:

$$||\Phi(|\psi\rangle) - |junk\rangle_B \left|\Psi^+\right\rangle_{AB} || \leq 3\gamma_1 + \gamma_1^2/4 \tag{4.38}$$

Lastly, together with inequality 4.25 and a triangle inequality, we get:

$$||\Phi(M_A N'_B |\psi\rangle) - |junk\rangle_B M_A N_B \left|\Psi^+\right\rangle_{AB} || \leq 3\gamma_1 + \gamma_1^2/4 + 2\gamma_2 \tag{4.39}$$

4.4 Proof of Theorem 9

Theorem 9 shows that saturating the correlation of observables on Alice and Bob's side, with Alice being trusted, leads to the necessary conditions of Theorem 8 which imply that the shared state is close, up to local isometry, to a Bell state. Similar to Theorem 8 we start the proof by denoting B_0 as X'_B and B_1 as Y'_B. Splitting equation 4.7, we have:

$$\langle\psi|\, \mathsf{X}_A \mathsf{X}'_B \,|\psi\rangle + \langle\psi|\, \mathsf{Y}_A \mathsf{Y}'_B \,|\psi\rangle \geq 2 - \epsilon \tag{4.40}$$

However, it's clear that:

$$-1 \leq \langle\psi|\, \mathsf{X}_A \mathsf{X}'_B \,|\psi\rangle \leq 1 \tag{4.41}$$

$$-1 \leq \langle\psi|\, \mathsf{Y}_A \mathsf{Y}'_B \,|\psi\rangle \leq 1 \tag{4.42}$$

So, it follows that:

$$\langle\psi|\, \mathsf{X}_A \mathsf{X}'_B \,|\psi\rangle \geq 1 - \epsilon \tag{4.43}$$

$$\langle\psi|\, \mathsf{Y}_A \mathsf{Y}'_B \,|\psi\rangle \geq 1 - \epsilon \tag{4.44}$$

This allows us to derive conditions 4.1 and 4.2, since:

$$||(\mathsf{X}_A - \mathsf{X}'_B)\,|\psi\rangle\,|| = \sqrt{2}\sqrt{1 - \langle\psi|\,\mathsf{X}_A\mathsf{X}'_B\,|\psi\rangle} \leq \sqrt{2\epsilon} \tag{4.45}$$

$$||(\mathsf{Y}_A - \mathsf{Y}'_B)\,|\psi\rangle\,|| = \sqrt{2}\sqrt{1 - \langle\psi|\,\mathsf{Y}_A\mathsf{Y}'_B\,|\psi\rangle} \leq \sqrt{2\epsilon} \tag{4.46}$$

Hence, in Theorem 8, $\gamma_1 = \sqrt{2\epsilon}$. Let us now denote:

$$S = \mathsf{X}_A\mathsf{X}'_B + \mathsf{Y}_A\mathsf{Y}'_B \tag{4.47}$$

Computing S^2 and using the fact that $\mathsf{X}_A\mathsf{Y}_A = i\mathsf{Z}_A$ we obtain:

$$S^2 = 2 + iZ_A[\mathsf{X}'_B, \mathsf{Y}'_B] \tag{4.48}$$

Since $[\mathsf{X}_A, \mathsf{Y}_A] = 2i\mathsf{Z}_A$, we can alternatively write this as:

$$S^2 = 2 + \frac{1}{2}[\mathsf{X}_A, \mathsf{Y}_A][\mathsf{X}'_B, \mathsf{Y}'_B] \tag{4.49}$$

The Cauchy-Schwarz inequality together with inequality 4.40 give us:

$$\langle\psi|\,S^2\,|\psi\rangle \geq |\,\langle\psi|\,S\,|\psi\rangle\,|^2 \geq (2-\epsilon)^2 \tag{4.50}$$

Substituting S^2:

$$\langle\psi|\,2 + \frac{1}{2}[\mathsf{X}_A, \mathsf{Y}_A][\mathsf{X}'_B, \mathsf{Y}'_B]\,|\psi\rangle \geq 4 - 4\epsilon + \epsilon^2 \geq 4 - 4\epsilon \tag{4.51}$$

Hence:

$$\langle\psi|\,[\mathsf{X}_A, \mathsf{Y}_A][\mathsf{X}'_B, \mathsf{Y}'_B]\,|\psi\rangle \geq 4 - 8\epsilon \tag{4.52}$$

Expanding the commutators yields:

$$\begin{aligned}&\langle\psi|\,\mathsf{X}_A\mathsf{Y}_A\mathsf{X}'_B\mathsf{Y}'_B\,|\psi\rangle - \langle\psi|\,\mathsf{X}_A\mathsf{Y}_A\mathsf{Y}'_B\mathsf{X}'_B\,|\psi\rangle\\ &- \langle\psi|\,\mathsf{Y}_A\mathsf{X}_A\mathsf{X}'_B\mathsf{Y}'_B\,|\psi\rangle + \langle\psi|\,\mathsf{Y}_A\mathsf{X}_A\mathsf{Y}'_B\mathsf{X}'_B\,|\psi\rangle \geq 4 - 8\epsilon\end{aligned}$$

By splitting into terms, as we did with inequality 4.40, we have that:

$$\langle\psi|\,\mathsf{X}_A\mathsf{Y}_A\mathsf{X}'_B\mathsf{Y}'_B\,|\psi\rangle \geq 1 - 8\epsilon \tag{4.53}$$

$$\langle\psi|\,\mathsf{Y}_A\mathsf{X}_A\mathsf{Y}'_B\mathsf{X}'_B\,|\psi\rangle \geq 1 - 8\epsilon \tag{4.54}$$

$$\langle\psi|\,\mathsf{X}_A\mathsf{Y}_A\mathsf{Y}'_B\mathsf{X}'_B\,|\psi\rangle \leq 8\epsilon - 1 \tag{4.55}$$

$$\langle\psi|\,\mathsf{Y}_A\mathsf{X}_A\mathsf{X}'_B\mathsf{Y}'_B\,|\psi\rangle \leq 8\epsilon - 1 \tag{4.56}$$

Now using the fact that $\mathsf{X}_A\mathsf{Y}_A + \mathsf{Y}_A\mathsf{X}_A = 0$, we have:

$$\begin{aligned}&||(\mathsf{X}'_B\mathsf{Y}'_B + \mathsf{Y}'_B\mathsf{X}'_B)\,|\psi\rangle\,|| =\\ &||(\mathsf{X}'_B\mathsf{Y}'_B + \mathsf{X}_A\mathsf{Y}_A + \mathsf{Y}_A\mathsf{X}_A + \mathsf{Y}'_B\mathsf{X}'_B)\,|\psi\rangle\,||\end{aligned}$$

And using a triangle inequality, we have:

$$||(\mathsf{X}'_B\mathsf{Y}'_B + \mathsf{Y}'_B\mathsf{X}'_B)\left|\psi\right\rangle|| \leq \\ ||(\mathsf{X}_A\mathsf{Y}_A + \mathsf{X}'_B\mathsf{Y}'_B)\left|\psi\right\rangle|| + ||(\mathsf{Y}_A\mathsf{X}_A + \mathsf{Y}'_B\mathsf{X}'_B)\left|\psi\right\rangle||$$

Additionally:

$$||(\mathsf{X}_A\mathsf{Y}_A + \mathsf{X}'_B\mathsf{Y}'_B)\left|\psi\right\rangle|| = \\ \sqrt{2 + \left\langle\psi\right|\mathsf{X}_A\mathsf{Y}_A\mathsf{Y}'_B\mathsf{X}'_B\left|\psi\right\rangle + \left\langle\psi\right|\mathsf{Y}_A\mathsf{X}_A\mathsf{X}'_B\mathsf{Y}'_B\left|\psi\right\rangle}$$

And from inequalities 4.55 and 4.56 we get that:

$$||(\mathsf{X}_A\mathsf{Y}_A + \mathsf{X}'_B\mathsf{Y}'_B)\left|\psi\right\rangle|| \leq 4\sqrt{\epsilon} \tag{4.57}$$

Similarly, using inequalities 4.53 and 4.54, we have:

$$||(\mathsf{Y}_A\mathsf{X}_A + \mathsf{Y}'_B\mathsf{X}'_B)\left|\psi\right\rangle|| \leq 4\sqrt{\epsilon} \tag{4.58}$$

Which leads to:

$$||(\mathsf{X}'_B\mathsf{Y}'_B + \mathsf{Y}'_B\mathsf{X}'_B)\left|\psi\right\rangle|| \leq 8\sqrt{\epsilon} \tag{4.59}$$

Thus satisfying condition 4.3, with $\gamma_2 = 8\sqrt{\epsilon}$, and concluding the proof.

4.5 Proof of Theorem 10

Theorems 8 and 9 show that if the correlation of local observables is saturated up to order $O(\epsilon)$, the shared state is close, up to local isometry, to a Bell state up to order $O(\sqrt{\epsilon})$. Theorem 10 shows that this bound is tight, up to constant factors. We prove this theorem by contradiction. Assume the bound of Theorem 8 is not tight and it is possible to derive an asymptotically better bound for the shared state of Alice and Bob. In particular, this means that there is no state $\left|\psi\right\rangle$ which is $O(\sqrt{\epsilon})$-close to the $\left|\Psi^+\right\rangle$ state and there are no observables B_0 and B_1 such that inequality 4.10 is satisfied. However, letting $\epsilon' = \epsilon/2$, consider the following state:

$$\left|\psi\right\rangle = \frac{1}{\sqrt{2}}\left(\sqrt{1+\sqrt{\epsilon'}}\left|01\right\rangle + \sqrt{1-\sqrt{\epsilon'}}\left|10\right\rangle\right) \tag{4.60}$$

We have that:

$$||\left|\psi\right\rangle - \left|\Psi^+\right\rangle|| = \sqrt{2 - \left\langle\psi\,\middle|\,\Psi^+\right\rangle - \left\langle\Psi^+\,\middle|\,\psi\right\rangle} \tag{4.61}$$

Notice that:

$$\left\langle\psi\,\middle|\,\Psi^+\right\rangle = \left\langle\Psi^+\,\middle|\,\psi\right\rangle = \frac{1}{2}\left(\sqrt{1+\sqrt{\epsilon'}} + \sqrt{1-\sqrt{\epsilon'}}\right) \tag{4.62}$$

Substituting this into the previous expression, we have:

$$||\left|\psi\right\rangle - \left|\Psi^+\right\rangle|| = O(\sqrt{\epsilon'}) \tag{4.63}$$

Consider also the observables:

$$\begin{aligned} B_0 &= \begin{pmatrix} -\sqrt{\epsilon'} & \sqrt{1-\epsilon'} \\ \sqrt{1-\epsilon'} & \sqrt{\epsilon'} \end{pmatrix} \\ B_1 &= \begin{pmatrix} 0 & \sqrt{\epsilon'} - i\sqrt{1-\epsilon'} \\ \sqrt{\epsilon'} + i\sqrt{1-\epsilon'} & 0 \end{pmatrix} \end{aligned}$$

One can check that $B_0 = B_0^\dagger$, $B_1 = B_1^\dagger$, $B_0 B_0^\dagger = B_1 B_1^\dagger = I$ and that the two matrices have eigenvalues ± 1. Moreover, we can see that as $\epsilon' \to 0$ we have that $B_0 \to \mathsf{X}$ and $B_1 \to \mathsf{Y}$. Importantly, we have that:

$$\langle\psi| \mathsf{X}_A B_0 |\psi\rangle = \langle\psi| \mathsf{Y}_A B_1 |\psi\rangle = 1 - \epsilon' \tag{4.64}$$

And therefore:

$$\langle\psi| (\mathsf{X}_A B_0 + \mathsf{Y}_A B_1) |\psi\rangle = 2 - 2\epsilon' = 2 - \epsilon \tag{4.65}$$

Thus, inequality 4.10 is saturated. This should not be possible under the assumption that the bound on $|\psi\rangle$'s closeness to $|\Psi^+\rangle$ is not tight. Therefore, the assumption is false and the $O(\sqrt{\epsilon})$ bound is tight for this type of steering inequality. Note that this result still holds under local isometry since the isometry is, by definition, distance preserving and so under the local isometry the state is still $O(\sqrt{\epsilon})$-close to a Bell pair.

4.6 Proof of Theorem 11

Theorem 11 shows that from the observed outcomes of Alice and Bob, after K rounds of measurement, we can conclude something about their shared quantum state in a single, randomly chosen, round, even without assuming independence. The proof consists of two steps:

1. Firstly, we show that the observed correlations of Alice and Bob, given fixed measurement settings, provide a good estimate for the true quantum correlation assuming they shared the averaged state $\rho_{avg} = \frac{1}{K}\sum_{i=1}^{K} \rho_i$.
2. Secondly, we use the previous result to estimate the correlations for the two measurement settings under consideration. We then use self-testing to show that if the correlations are close to the maximal value, the averaged state is close to a Bell state, under a suitable local isometry.

We start by proving the first step:

Lemma 6. *Assume Alice and Bob are asked to perform n rounds of measurement of the two-outcome observables with ± 1 eigenvalues, A and B, respectively. We denote the outcomes of their measurements as $\{a_i\}$ and $\{b_i\}$ and $\hat{C}_i = a_i b_i$ as their correlation for round i. Additionally, let $H_i = \{(a_j, b_j) | j < i\}$ be the history of their measurement outcomes up to, but not including round i. Finally, letting*

$C_i = E(\hat{C}_i|H_i)$ be the conditional expectation value of the correlation given the previous history of outcomes, we have that for any $\delta > 0$:

$$Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n} C_i - \frac{1}{n}\sum_{i=1}^{n}\hat{C}_i\right| \geq \delta\right) \leq exp(-\delta^2 n/8) \tag{4.66}$$

Proof. The variable C_i represents the true correlation of the outcomes in round i, as determined by the shared state of Alice and Bob. If the shared state in round i is ρ_i then $C_i = Tr(AB\ \rho_i)$. As mentioned, while we trust Alice and know that she is indeed measuring the observable A, we can still assume that Bob is measuring the observable B in each round. This is because the observable B is unrestricted (apart from being a two-outcome observable) and can in principle act on Bob's ancilla as well. Furthermore, we make no assumption about the state ρ_i, since it is prepared by Bob. Another way in which we can express C_i is using its definition, which leads us to:

$$C_i = Pr(a_i = b_i|H_i) - Pr(a_i \neq b_i|H_i) \tag{4.67}$$

We now define the random variables:

$$X_j = \sum_{i=1}^{j}(C_i - \hat{C}_i) \tag{4.68}$$

Notice that for any $j \leq n$, $|X_{j+1} - X_j| \leq 2$ (because $\hat{C}_i = \pm 1$, $-1 \leq C_i \leq 1$), $E(X_j) \leq \infty$ and:

$$E(X_{j+1} - X_j|H_{j+1}) = C_{j+1} - C_{j+1} = 0 \tag{4.69}$$

Therefore, $\{X_j\}$ forms a martingale. We can therefore apply the Azuma-Hoeffding inequality [135], in a manner analogous to [34, 116]. Setting $j = n$, we have that for any $t > 0$:

$$Pr(|X_n| > t) \leq exp(-t^2/8n) \tag{4.70}$$

Expanding, we have that:

$$Pr\left(\left|\sum_{i=1}^{n}(C_i - \hat{C}_i)\right| > t\right) \leq exp(-t^2/8n) \tag{4.71}$$

For some $\delta > 0$, let $t = n\delta$. This yields:

$$Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n} C_i - \frac{1}{n}\sum_{i=1}^{n}\hat{C}_i\right| > \delta\right) \leq exp(-\delta^2 n/8) \tag{4.72}$$

Thus concluding the proof of Lemma 6. □

We can now prove the second step, thus proving Theorem 11.

Proof of Theorem 11. We will treat the case $\hat{C}^0 + \hat{C}^1 \geq 2 - \epsilon$ since for $\hat{C}^0 + \hat{C}^1 \leq -2 + \epsilon$ the derivation is similar. Additionally, we only consider the case $\mathcal{O}_{AB} = I$, since the other cases follow from the linearity of the operators. The previous lemma essentially shows us that the observed average correlation is a good estimate for the average true correlation. Specifically, it is the case that $\hat{C}^b$, $b \in \{0, 1\}$, is close to the quantum correlation $Tr(A_b B_b \rho_{avg})$. Consider now a state $|\zeta\rangle$ which is a purification of ρ_{avg}. We can then write the quantum correlation as $\langle\zeta| A_b B_b |\zeta\rangle$. Using these results, if our estimate of the true correlation is of precision (closeness) $\delta > 0$, this leads us to conclude that:

$$\langle\zeta| A_0 B_0 |\zeta\rangle + \langle\zeta| A_1 B_1 |\zeta\rangle \geq 2 - \epsilon - \delta \tag{4.73}$$

with probability $1 - exp(-\delta^2 K/16)$. Let $\delta = \epsilon$ so that we have:

$$\langle\zeta| A_0 B_0 |\zeta\rangle + \langle\zeta| A_1 B_1 |\zeta\rangle \geq 2 - O(\epsilon) \tag{4.74}$$

Using Theorem 9, it follows that there exists a local isometry Φ and a state $|junk\rangle$ such that, with probability $1 - exp(-\epsilon^2 K/16)$, we have:

$$||\Phi(|\zeta\rangle) - |\Psi^+\rangle |junk\rangle || \leq O(\sqrt{\epsilon}) \tag{4.75}$$

This also implies:

$$TD(\Phi(|\zeta\rangle), |\Psi^+\rangle |junk\rangle) \leq O(\sqrt{\epsilon}) \tag{4.76}$$

As mentioned, we are only considering the case of I acting on the state $|\zeta\rangle$. Of course, the argument proceeds identically, when considering $M_A N'_B |\zeta\rangle$, as in Theorem 8, leading to the $\mathcal{O}_{AB} \neq I$ cases. It should be noted from the construction of Φ (in Theorem 8), in the case where the shared state is a purification of some mixed state (as is the case with $|\zeta\rangle$ and ρ_{avg}), that the isometry does not act on the quantum states used for purification. Therefore, we can trace out those states, and since this operation cannot increase trace distance we have that:

$$TD(\Phi(\rho_{avg}), |\Psi^+\rangle \langle\Psi^+| \otimes \rho_{junk}) \leq O(\sqrt{\epsilon}) \tag{4.77}$$

With probability $1 - exp(-\epsilon^2 K/16)$. We can incorporate this probability into the trace distance[10], and we have that:

$$TD(\Phi(\rho_{avg}), |\Psi^+\rangle \langle\Psi^+| \otimes \rho_{junk}) \leq O(\sqrt{\epsilon}) + exp(-\epsilon^2 K/16) \tag{4.78}$$

Setting $K = -(16/\epsilon^2)log(\sqrt{\epsilon}) = (8/\epsilon^2)log(1/\epsilon)$ we are left with:

$$TD(\Phi(\rho_{avg}), |\Psi^+\rangle \langle\Psi^+| \otimes \rho_{junk}) \leq O(\sqrt{\epsilon}) \tag{4.79}$$

[10] By this we mean that if we know that $TD(\rho, \sigma) \leq \epsilon$ with probability $1 - p$, this means that with probability p the trace distance could be anything. In particular, the states can be orthogonal, in which case their trace distance would be 1. This means that with probability 1, $TD(\rho, \sigma) \leq (1 - p)\epsilon + p$.

Analogously, we would get:

$$TD(\Phi(\mathcal{O}_{AB}(\rho_{avg})), \hat{\mathcal{O}}_{AB}(\left|\Psi^+\right\rangle\left\langle\Psi^+\right|) \otimes \rho_{junk}) \leq O(\sqrt{\epsilon}) \tag{4.80}$$

Concluding the proof of Theorem 11. □

4.7 Proof of Theorem 12

The main idea of the proof is to use the Gentle Measurement Lemma (Theorem 6), as well as its corollary from Chapter 3 (Corollary 1). Throughout the proof we will suppress the isometry acting on Bob's system, to reduce the complexity of our expressions. However, it is to be understood that all the states we write have the local isometry acting on them. We will also omit the action of $\mathcal{O}_{AB}$, since, just like the isometry, this can also be included in the proof.

Let us denote the shared state ρ of Alice and Bob, for the selected rounds, as $\rho_{A_1A_2...A_NB} \in \mathcal{D}(\mathcal{H}_{A_1} \otimes ... \otimes \mathcal{H}_{A_N} \otimes \mathcal{H}_B)$, where $\mathcal{H}_{A_i}$ is the Hilbert space for Alice's system in round i, and $\mathcal{H}_B$ is Bob's Hilbert space. Whenever we remove one system from that state, it is to be understood that that system has been traced out. For instance:

$$Tr_{A_1}(\rho_{A_1A_2...A_NB}) = Tr_1(\rho_{A_1A_2...A_NB}) = \rho_{A_2...A_NB} \tag{4.81}$$

We also denote as $\Pi_{A_1A_2...A_NB} = \bigotimes_{i=1}^N \left|\Phi^+\right\rangle\left\langle\Phi^+\right|_{A_iB}$ a projection onto N Bell pairs between Alice and Bob.

To start with, we know from the fact that the reduced state in each round saturates the steering inequality that:

$$Tr(\Pi_{A_iB}\rho_{A_iB}) \geq 1 - O(\epsilon) \tag{4.82}$$

for all i, $1 \leq i \leq N$. Note that $Tr(\Pi_{A_iB}\rho_{A_1A_2...A_NB}) = Tr(\Pi_{A_iB}\rho_{A_iB})$, since the projector is only acting on the states from the i'th round. Using this and the Gentle Measurement Lemma gives us that:

$$\begin{aligned} &TD(\rho_{A_1A_2...A_NB}, \Pi_{A_iB} \otimes Tr_{A_i}(\rho_{A_1A_2...A_NB})) \leq \\ &\qquad 2\sqrt{1 - Tr(\Pi_{A_iB}\rho_{A_1A_2...A_NB})} \leq O(\sqrt{\epsilon}) \end{aligned} \tag{4.83}$$

where the second inequality follows from Equation 4.82.

Now, consider the quantity we are trying to bound, which we will denote as:

$$D = TD(\rho_{A_1A_2...A_NB}, \Pi_{A_1A_2...A_NB} \otimes \rho_B) \tag{4.84}$$

Using the triangle inequality, it follows that:

$$\begin{aligned} D \leq \sum_{j=1}^{N} &TD(\Pi_{A_1A_2...A_{j-1}B} \otimes Tr_{A_1...A_{j-1}}(\rho_{A_1A_2...A_NB}), \\ &\qquad \Pi_{A_1A_2...A_jB} \otimes Tr_{A_1...A_j}(\rho_{A_1A_2...A_NB})) \end{aligned} \tag{4.85}$$

But since taking the partial trace cannot increase trace distance, the right hand side of the inequality will be upper bounded by:

$$\sum_{j=1}^{N} TD(\rho_{A_1A_2...A_NB}, \Pi_{A_iB} \otimes Tr_{A_i}(\rho_{A_1A_2...A_NB})) \tag{4.86}$$

which we know, from inequality 4.83, is of order $O(\sqrt{\epsilon})$. It therefore follows that:

$$TD(\rho_{A_1A_2...A_NB}, \Pi_{A_1A_2...A_NB} \otimes \rho_B) \leq O(N\sqrt{\epsilon}) \tag{4.87}$$

thus showing that the state shared by Alice and Bob is close to a tensor product of Bell pairs, tensored with some junk state on Bob's system.

A few comments are in order. First of all, one could ask how was the fact that we are attempting to characterise a tensor product of Bell states, rather than any other pure state, used. It is true that we made use of Equation 4.82, giving the closeness of the state in round i to a Bell pair, however the proof itself seems agnostic to which state was used. If we had a closeness relation for some other pure state, would the same argument allow us to derive a tensor product for multiple copies of that state? The answer is no. Suppose for instance that $\Pi_{A_iB} = |00\rangle$. In other words, we could determine that in each round, the state of Alice and Bob is close to $|00\rangle$. While we do have a characterisation of the tensor product structure of Alice's system (i.e. $\mathcal{H}_A = \mathcal{H}_{A_1} \otimes \mathcal{H}_{A_2} \otimes ... \otimes \mathcal{H}_{A_N}$), no such characterisation is known for Bob. Thus, it is entirely possible that Bob's system is simply a two dimensional Hilbert space $\mathcal{H}_B \cong \mathbb{C}^2$. In this case, $\Pi_{A_1A_2...A_NB}$ would be a projection on the $N+1$-qubit state $|0\rangle_{A_1} |0\rangle_{A_2} ... |0\rangle_{A_N} |0\rangle_B$. Having this state is consistent with the reduced state in each round being $|00\rangle$, but, as we can see, the joint state of Alice and Bob is not equivalent to N pairs of the form $|00\rangle$.

In our case, however, we are implicitly using the monogamy property of Bell states. If the state on $\mathcal{H}_{A_1} \otimes \mathcal{H}_B$ is a Bell pair and the state on $\mathcal{H}_{A_2} \otimes \mathcal{H}_B$ is a Bell pair, and $\mathcal{H}_{A_1} \neq \mathcal{H}_{A_2}$, then necessarily these are two distinct Bell pairs. Thus, the projector $\Pi_{A_1A_2...A_NB}$ is a projection on N distinct Bell states.

A second question one could ask is why can't one use such a simple argument to derive the tensor product structure of Bell pairs in the fully device-independent case, where both Alice and Bob are untrusted. The answer is that in that case no characterisation is known, a priori, for the Hilbert spaces of Alice and Bob, except that $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$, from the fact that they are not allowed to communicate. But this is not sufficient to derive the tensor product structure of Bell states, using the above argument. Indeed, if one assumes that the $\mathcal{H}_{A_i}$ spaces are not distinct and that, for instance $\mathcal{H}_A = \mathbb{C}^2$, then the reduced state of each round is close to a Bell pair, because it is *the same* Bell pair for all rounds[11]. In other words, one would have that $\Pi_{A_1A_2...A_NB} = \Pi_{A_iB} = |\Phi^+\rangle \langle\Phi^+|_{AB}$. A more involved argument is therefore required, as was used in [31].

[11] To clarify, this cannot happen in practice since once the state is measured it cannot be reused. However, we are not using the fact that the state is measured in the proof presented above. We are simply using the fact that the reduced state in each round is close to a Bell state. As we've seen, this is sufficient in the steering case, because we trust Alice's system.

4.8 Proof of Theorem 14

To prove the result, let us first state a lemma from [151] concerning completely steerable states:

Lemma 7. *A bipartite state, ρ_{AB} is completely steerable iff there exists a purification ρ_{ABC} such that $\rho_{BC} = \rho_B \otimes \rho_C$, where $\rho_{BC} = Tr_A(\rho_{ABC})$, $\rho_B = Tr_{AC}(\rho_{ABC})$, $\rho_C = Tr_{AB}(\rho_{ABC})$.*

It is clear that a maximally entangled Bell state satisfies the properties of total steerability. We therefore focus on proving that a totally steerable state is maximally entangled. We start by considering $|\psi_{ABC}\rangle$ as the 4-qubit purification of ρ_{AB}. All other purifications are equivalent to this one, so this suffices for our purposes. Writing $|\psi_{ABC}\rangle$ in the computational basis, we have:

$$|\psi_{ABC}\rangle = \sum_{i=0}^{15} a_i |i\rangle \tag{4.88}$$

Of course, we have the additional constraint:

$$\sum_{i=0}^{15} |a_i|^2 = 1 \tag{4.89}$$

By re-expressing the constraints from our description of totally steerable states together with Lemma 7:

$$\rho_{BC} = \rho_B \otimes \rho_C \tag{4.90}$$

$$\rho_B = I/2 \tag{4.91}$$

as constraints on the amplitudes of $|\psi_{ABC}\rangle$ we will have a large bilinear system of equations. From this system we will arrive at the following set of equations:

$$\begin{aligned}
a_{2k} &= f \cdot a_{2k+2}, \quad k \in \{0, 1, ...6\} \\
a_{2k+3} &= -f^* \cdot a_{2k+1}, \quad k \in \{0, 1, ...6\} \\
a_{4k+1} &= e^{i\phi_1} \cdot a_{4k+2}, \quad k \in \{0, 1, 2, 3\} \\
a_{4k} &= e^{i\phi_2} \cdot a_{4k+3}, \quad k \in \{0, 1, 2, 3\}
\end{aligned}$$

Where the parameters we introduced are $f \in \mathbb{C}$ and $\phi_1, \phi_2 \in [0, 2\pi]$. Computing the matrix elements of ρ_{AB}, we arrive at the most general form, given by:

$$\rho_{AB} = \frac{1}{2(|f|^2+1)} \begin{pmatrix} |f|^2 & f & f & e^{i\phi_1}|f|^2 \\ f^* & 1 & e^{i\phi_2} & -f \\ f^* & e^{-i\phi_2} & 1 & -f \\ e^{-i\phi_1}|f|^2 & -f^* & -f^* & |f|^2 \end{pmatrix} \tag{4.92}$$

It can be easily checked that $Tr(\rho_{AB}^2) = 1$ and therefore ρ_{AB} is a pure state. But since $\rho_B = I/2$, we have that ρ_{AB} is a maximally entangled state, i.e. a Bell state.

4.9 Chapter summary and outlook

Our results show that, in analogy to the rigidity of non-local correlations via CHSH games, we can prove a rigidity property of steering correlations. This allows us to establish a tensor product of Bell pairs in a setting in which one party is trusted and one is not. This setting makes the analysis simpler, than in the device-independent case, and leads to a reduced overhead in characterising the tensor product structure of Bell pairs. As we mentioned, however, the analysis in the device-independent case is not tight, and so it is possible that an optimal analysis would reveal the same bounds. Nevertheless, the simplicity of the steering case demonstrates the advantage of added trust.

We arrived at the rigidity result by first considering self-testing of a Bell state and Pauli X and Y observables from steering correlations. We also showed that the bound derived for the closeness of the states and operators, in this particular self-test, is optimal, up to constant factors. Of course, self-testing assumes ideal expectation values for the considered observables and so we had to prove a similar result in the case where these values are estimated from finite statistics. We did this in the most general case, where no assumption of independence was made. Finally, we used this result to derive the rigidity of steering correlations, characterising a tensor product of Bell pairs and the operators acting on these states. As mentioned, it is an open problem whether the bound derived for our rigidity result, of $O(N\sqrt{\epsilon})$, is optimal. Numerical data from solving SDPs suggests that this is the case, though we leave a more definitive analysis for future work.

Using the rigidity of steering correlations we constructed a one-sided device-independent protocol for verifiable delegated quantum computation. The protocol we obtained is essentially identical to the device-independent analogue, but with the added distinction of trusting the verifier's measurement device. This, of course, lead to a smaller communication complexity. Finally, we have shown that a certain class of states that would be useful for such one-sided device-independent verification protocols, are necessarily maximally entangled.

As mentioned, recent results have introduced self-tests that lead to improved bounds in characterising a tensor product structure of Bell states [36, 106] and so it would be interesting to see how one could improve on these results by applying them to the steering setting. One thing to note about these results is that they require the untrusted parties participating in the self-test to perform *collective measurements* of their states, i.e. measuring multi-qubit observables. However, ideally, we would like one of the two parties (the verifier) to perform only single-qubit measurements. Hence, another avenue worth exploring is whether those self-tests, combined with the steering setting, would allow for this, while maintaining the quasi-linear communication complexity of the resulting protocol.

Chapter 5

Fault tolerant verification of quantum computation

> **James T. Kirk:** We'd better start solving problems faster than we pick up new ones.
>
> — Star Trek: The Original Series, Season 2, Episode 18

Both with the device-independent verification protocol of Chapter 3, as well as with the one-sided device-independent one of Chapter 4, we assumed that all quantum devices work as expected and are not subject to noise or imperfections[1]. This, of course, is not a realistic assumption. Real devices can have their quantum systems being correlated with the external environment leading to a loss of coherence and the ability to perform quantum computations. This is one of the major obstacles in the development of large-scale quantum computers. Fortunately, it has been shown that as long as the noise per quantum gate is below some constant threshold (along with some additional assumptions), it is possible to detect and correct for errors occurring in the system [152–154]. We would like to incorporate quantum error correction into our verification protocols, so that they can operate fault tolerantly in a more realistic environment.

For protocols in which the verifier is fully classical, fault tolerance is not a concern since one can assume that the provers are performing their quantum operations on top of a quantum error correcting code. Since provers are assumed to have universal quantum computing power, we naturally have to assume that they are capable of fault tolerant quantum computation between themselves. We emphasise that discussions about fault tolerance only make sense in the setting in which the verifier possesses a quantum device[2].

One could imagine that we can simply take an existing verification protocol, such as the FK protocol and have it run on top of a quantum error-correcting

[1] Strictly speaking, in the device-independent case, we assumed nothing about the quantum devices. Nevertheless, in the honest setting, the measurement device might be subject to noise that would lead to incorrect results, forcing the verifier to reject most of the time.

[2] For non-universal provers, achieving fault tolerant verification is also an interesting question since, depending on the quantum capabilities of the provers, it might not be possible for them to perform their computation on top of a quantum error-correcting code. In this chapter, however, we are only interested in the case of a universal prover.

code. In other words, the verifier would prepare logical qubits, send these qubits to the prover and then instruct him to proceed, as in the FK protocol, by entangling and measuring these encoded states. However, this approach requires that the verifier is able to prepare the logical states, in a fault tolerant way. As is explained in [155], standard constructions for achieving this for a computation with m qubits and consisting of N gates require the fault tolerant circuit to act on $O(m\, polylog(mN))$ qubits. In other words, one requires $O(polylog(mN))$ physical qubits per logical qubit. We would like to avoid the verifier's quantum device becoming too powerful, since even being polylogarithmic in the size of the input to the computation would be too powerful. This is because, for polylogarithmic size quantum circuits, there is no known general means of classically simulating them, as a brute force simulation would run in quasipolynomial time. The result of Gottesman from [155] provides an alternative, namely a construction in which the number of physical qubits per logical qubit becomes constant in the limit of N going to infinity. However, this construction requires multiple logical qubits to be encoded in a single block[3] of physical qubits. This is in contrast to a naive fault tolerant version of the FK protocol in which each block of physical qubits prepared by the verifier would comprise one logical qubit. It is therefore not clear if the construction of [155] would be suitable for making a verification protocol fault tolerant. Furthermore, even if the errors of the verifier's device can be suppressed, it still needs to be proven that this is not detrimental for the verification of a quantum computation. In other words, we want to ensure that a malicious prover cannot exploit these errors in order to successfully trick the verifier into accepting incorrect results. Thus, the question we aim to answer is: can a verifier with a constant-size and imperfect quantum device verifiably delegate a quantum computation to a single prover?

We show that this is indeed possible for a verifier with a single-qubit measurement device. In the terminology of Chapter 1, our protocol will be a receive-and-measure verification protocol. Our approach is based on the Morimae and Fitzsimons (MF) *post hoc* verification protocol, which we introduced in Chapter 2. Recall that in that protocol a prover sends a quantum state to a verifier and this quantum state should be the ground state of a Hamiltonian. This ground state encodes the desired quantum computation and can be used to "read off" the outcome of that computation. If the verifier can indeed certify that this is the ground state, then the computation is verified. In our fault tolerant protocol we encode the qubits of this ground state into a *logical ground state* where each qubit of the original state is encoded into a larger number of physical qubits via a quantum error correction code. This logical state is then the ground state of a logical Hamiltonian described by the quantum computation. In the protocol, the physical qubits in this logical state are then measured one at a time, and appropriate classical corrections are made on the outcomes of these measurements in post-processing if errors are detected. An honest prover's probability of successful computation will be boosted by this error correction, but importantly we can still verify if the logical ground state was indeed prepared by the prover. This is

[3] By 'block' we simply mean a collection of qubits that are processed at the same time and therefore need to be stored in a quantum memory at the same time.

shown in Section 5.1. Then, in Section 5.2, we consider a simple example of this protocol in the honest prover scenario. That is, using the repetition code and the Steane code [156], we can simulate and characterise the protocol's behaviour under bit-flip errors and depolarizing noise. The numerical results of our simulation are presented in Section 5.3.

Let us first comment on approaches that have also addressed the aforementioned problem of fault tolerant verification. For protocols in which the verifier has a small quantum device, the question of fault tolerance has been addressed in [30, 35, 157–159]. In [30, 35, 157, 158] the authors proposed protocols in which a classical client possessing either a single qubit preparation or measurement device, susceptible to noise, could verifiably delegate quantum computations to a prover. All these protocols are blind, meaning that the delegated computation is kept secret from the prover. We will return to this issue in detail in Section 5.4. The requirement of blindness introduces new difficulties when considering fault tolerant computation. To circumvent these difficulties, the aforementioned approaches considered extra (potentially unrealistic) assumptions about the noise, which rule out the possibility of the prover utilising the noise to deceive the verifier. A discussion of the general difficulty in realizing a verifiable, blind, fault tolerant protocol is provided in [159].

Recall from Chapter 2, that in the MF protocol, for a particular decision problem, the verifier will consider an XZ-Hamiltonian, H, having the form:

$$H = \sum_i a_i S_i \tag{5.1}$$

where the a_i's are real coefficients and the S_i's are XZ-terms. When the answer to the decision problem is "yes", the ground state of the Hamiltonian will have energy below some value, b, and when the answer is "no", it will have energy above a. Importantly, it must be the case that $a - b > 1/poly(n)$, where n is the number of input bits to the decision problem (and also, the number of qubits on which H acts). The prover is required to prepare the ground state and send it to the verifier, who will use the single-qubit measurement device to measure one of the local terms of H, thus estimating the energy of that state. This inverse-polynomial gap between the acceptance and rejection cases, ensures that the verifier can distinguish between these two cases. Indeed, with a polynomial number of repetitions of the protocol, the verifier can boost the gap to a constant value.

What happens if we add noise to the verifier's measurement device? It is straightforward to show that a constant rate of noise on that device will lead to the failure of the protocol for sufficiently large computations. This is because the gap between acceptance and rejection is inverse polynomial in the size of the input. As a result of noisy devices, the acceptance threshold is shifted to $a - c$, and the rejection threshold is shifted to $b + c$, where c is some positive constant that depends on the noise rate of the devices. We can see that as long as $c < (a - b)/2$, the verifier can still distinguish reliably between acceptance and rejection. However, it is clear that for a sufficiently long input, we will have that $c \geq (a - b)/2$. At this point, the protocol no longer satisfies the correctness nor the soundness criteria. In fact, this is common to all other verification protocols in the single-prover setting [15]. To address this issue we now give a fault tolerant version of the MF protocol that works in the presence of quantum devices subject to local noise having a constant error-rate.

5.1 The fault tolerant protocol

Our construction is simple: we ask the prover to encode the history state in a CSS (Calderbank-Shor-Steane) error-correcting code [1] and send it to the verifier. The verifier will then perform a *transversal* measurement of the X and Z operators. Transversality results in the logical operators being expressed as tensor products of physical X and Z operators, i.e.:

$$\tilde{\mathsf{X}} = \bigotimes_{i=1}^{m} \mathsf{X}_i \qquad \tilde{\mathsf{Z}} = \bigotimes_{i=1}^{m} \mathsf{Z}_i \tag{5.2}$$

where $\tilde{\mathsf{X}}$ and $\tilde{\mathsf{Z}}$ are the logical (or encoded) X and Z operators. In effect, the original Hamiltonian is replaced with an encoded Hamiltonian by substituting each XZ-term with its corresponding logical form. The idea of encoding the history state in a CSS code is briefly mentioned in the independent work of [160], and CSS codes are also considered in [158], though not for post hoc verification.

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

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

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

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

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

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

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

Theorem 15. *The post hoc protocol of Morimae and Fitzsimons can be made fault tolerant by encoding the* XZ*-Hamiltonian of the protocol in a CSS code and having the verifier perform the* X *and* Z *measurements in a transversal fashion.*

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

$$E(\mathsf{X} \otimes I^{\otimes m-1})E^\dagger = \tilde{\mathsf{X}} \tag{5.3}$$

$$E(\mathsf{Z} \otimes I^{\otimes m-1})E^\dagger = \tilde{\mathsf{Z}} \tag{5.4}$$

Now let $H = \sum_i a_i S_i$ be an XZ-Hamiltonian acting on $n > 0$ qubits, and let $H' = H \otimes I^{n(m-1)}$. Clearly, H and H' have the same eigenvalues. But note that using Equations 5.3 and 5.4 we have that:

$$E^{\otimes n} S_i \otimes I^{n(m-1)} E^{\otimes n} = \tilde{S}_i \tag{5.5}$$

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

$$E^{\otimes n} H' E^{\otimes n} = \tilde{H} \tag{5.6}$$

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

$$\langle\psi| H |\psi\rangle = \langle\tilde{\psi}| \tilde{H} |\tilde{\psi}\rangle \tag{5.7}$$

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

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

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

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

$$b^k = \left(\frac{\log(2/\alpha\eta)}{\log(1/\alpha\epsilon_m)}\right)^{\log b}, \tag{5.8}$$

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

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

The idea of encoding the proof state in an error-correcting code while maintaining a single-qubit measurement device for the verifier has also been considered, in the context of general QMA problems, in [161]. In that case, however, the proof state is a graph state that is used by the verifier to perform a fault tolerant measurement-based quantum computation. The verifier is also required to test that this state corresponds to the correct graph state and this is achieved through a stabilizer test.

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

5.2 Example

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

$$D(\phi) = cos(\phi)\mathsf{Z} + sin(\phi)\mathsf{X} \tag{5.9}$$

we will choose the following one-qubit computation:

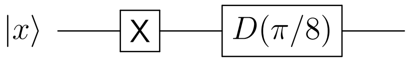


Figure 5.1: Example computation.

Note that $D(\pi/8)$ is universal for (real) single-qubit quantum computations[4]. The computation has two time steps, hence $T = 2$. Consider the case $x = 0$. The input state starts out as $|0\rangle$, it is then flipped to $|1\rangle$ and upon application of the $D(\pi/8)$ gate it becomes $sin(\pi/8)\,|0\rangle - cos(\pi/8)\,|1\rangle$. If we designate output $|1\rangle$ as acceptance, then this circuit will accept $x = 0$ with probability $cos(\pi/8)^2$. The history state, for $x = 0$, will be:

$$|\psi_{x=0}\rangle = \frac{1}{\sqrt{3}}(|0\rangle\,|00\rangle + |1\rangle\,|10\rangle + (sin(\pi/8)\,|0\rangle - cos(\pi/8)\,|1\rangle)\,|11\rangle) \tag{5.10}$$

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

$$|\psi_{x=1}\rangle = \frac{1}{\sqrt{3}}(|1\rangle\,|00\rangle + |0\rangle\,|10\rangle + (cos(\pi/8)\,|0\rangle + sin(\pi/8)\,|1\rangle)\,|11\rangle) \tag{5.11}$$

We now need to consider an XZ-Hamiltonian such that the ground state is close to $|\psi_{x=0}\rangle$. Since the 2-local construction is fairly involved and we are only interested in a simple example, we will instead consider a 3-local Hamiltonian. This, of course, does not change the protocol in any way and the verifier will still perform single-qubit X and Z measurements. Following the works of [162, 163], the Hamiltonian will have the following form:

$$H = H_{in} + H_{clock} + H_{prop} + H_{out} \tag{5.12}$$

where:

- H_{in} penalizes terms in which the input is not of the correct form, at the start of the computation ($T = 0$).

[4] Additionally, $\{CNOT, D(\pi/8)\}$ is universal for general quantum computations.

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

In our case, we have:

$$H_{in} = (I - |x\rangle \langle x|) \otimes |0\rangle \langle 0| \otimes I \tag{5.13}$$

$$H_{clock} = I \otimes |01\rangle \langle 01| \tag{5.14}$$

$$H_{prop} = H_{prop_1} + H_{prop_2} \tag{5.15}$$

where:

$$H_{prop_1} = \frac{1}{2}(I \otimes |0\rangle \langle 0| \otimes I - \mathsf{X} \otimes \mathsf{X} \otimes I + I \otimes |10\rangle \langle 10|) \tag{5.16}$$

$$H_{prop_2} = \frac{1}{2}(I \otimes I \otimes |1\rangle \langle 1| - D(\pi/8) \otimes I \otimes X + I \otimes |10\rangle \langle 10|) \tag{5.17}$$

and finally:

$$H_{out} = |0\rangle \langle 0| \otimes I \otimes |1\rangle \langle 1| \tag{5.18}$$

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

We now write H in XZ form:

$$H = \frac{7}{4}III + \frac{1}{4}(1 - (-1)^x)\mathsf{Z}II - \frac{1}{4}(-1)^x\mathsf{ZZ}I - \frac{1}{4}I\mathsf{ZZ} - \frac{1}{2}\mathsf{XX}I - \frac{1}{2}\mathsf{XXZ} -$$
$$- \frac{1}{2}sin(\pi/8)\mathsf{X}I\mathsf{X} + \frac{1}{2}sin(\pi/8)\mathsf{XZX} - \frac{1}{2}cos(\pi/8)\mathsf{Z}I\mathsf{X} + \frac{1}{2}cos(\pi/8)\mathsf{ZZX} - \frac{1}{4}\mathsf{Z}I\mathsf{Z} \tag{5.19}$$

The protocol proceeds as follows. The verifier will inform the prover that they wish to perform the computation from Figure 5.1, for input $x = 0$. The prover reports that the computation accepts (with high probability) and prepares the history state $|\psi_{x=0}\rangle$, encoded in a CSS code. This state is sent qubit by qubit to the verifier. The verifier, will choose one of the terms from Equation 5.19, with its corresponding probability, and perform the transversal measurement of the state. For instance, the term XZX will be chosen with probability $\frac{1}{2K}sin(\pi/8)$, where $K = \sum_i |a_i| \approx 4.8$. The verifier measures the X and Z operators, performs classical post-processing on their results and combines them so as to recover the outcome of measuring XZX. She accepts on outcome -1 for this measurement, since $\frac{1}{2}sin(\pi/8)$ is positive.

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

$$H_{out} = |1\rangle\langle 1| \otimes I \otimes |1\rangle\langle 1| \tag{5.20}$$

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

5.3 Numerical results

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

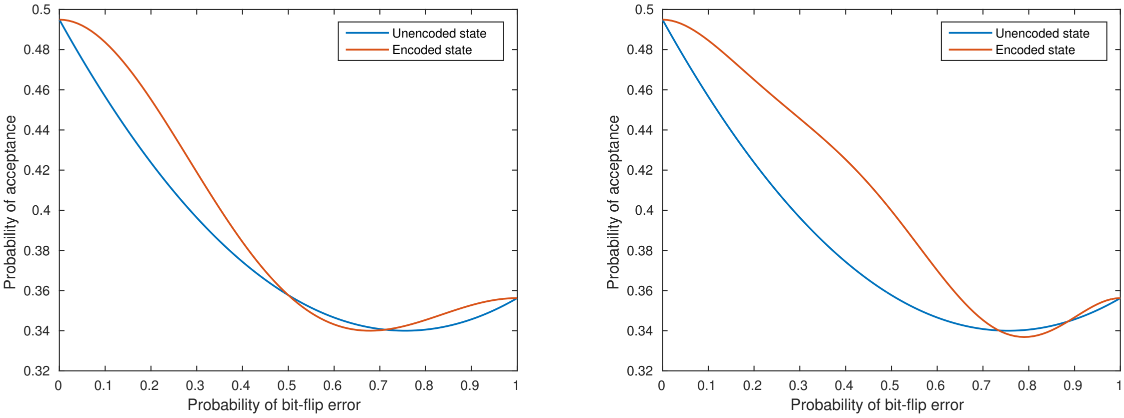
$$p_{acc} = \frac{1}{2}\left(1 - \frac{\langle\psi_{x=0}| H |\psi_{x=0}\rangle}{\sum_i |a_i|}\right) \tag{5.21}$$

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

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

$$\mathcal{F}(\rho) = (1-p)\rho + p\mathsf{X}\rho\mathsf{X} \tag{5.22}$$

acting independently on each individual qubit. The results are shown in Subfigure 5.2a.



(a) Comparison between encoded and unencoded states for the 3-qubit repetition code.

(b) Comparison between encoded and unencoded states for the 5-qubit repetition code.

Figure 5.2: Results for repetition code.

As we can see, the point where the encoded state yields the same acceptance probability as the unencoded state is $p = 0.5$. The acceptance probabilities for the unencoded state were determined by applying the channel $\mathcal{F}$ to each qubit in

$|\psi_{x=0}\rangle$, resulting in a state ρ, and then computing:

$$p_{acc} = \frac{1}{2} - \frac{Tr(H\rho)}{2\sum_i |a_i|} \tag{5.23}$$

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

$$\mathsf{Z}_M = M_0 - M_1 \tag{5.24}$$

where:

$$M_0 = |000\rangle \langle 000| + |001\rangle \langle 001| + |010\rangle \langle 010| + |100\rangle \langle 100| \tag{5.25}$$

$$M_1 = |111\rangle \langle 111| + |110\rangle \langle 110| + |101\rangle \langle 101| + |011\rangle \langle 011| \tag{5.26}$$

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

If we increase the size of the encoded state to 5 qubits, we obtain the results from Subfigure 5.2b. As expected, the noise threshold increases and is around $p \approx 0.72$.

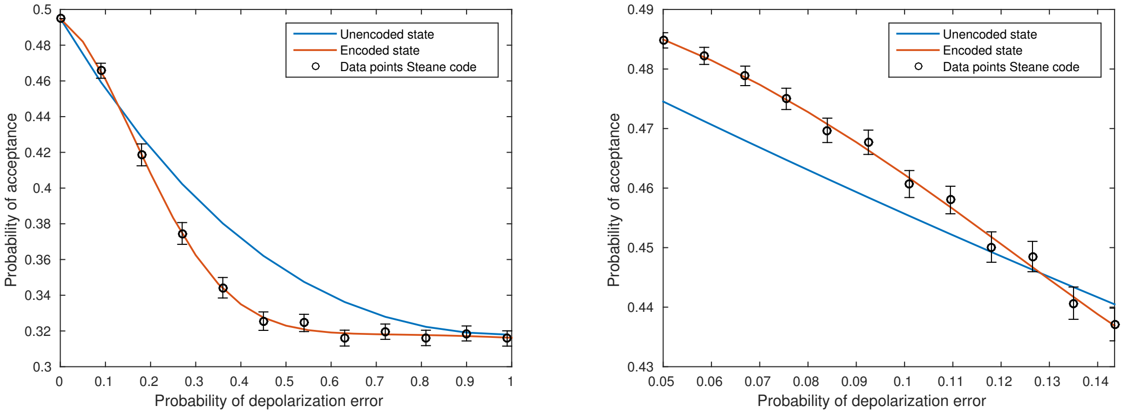
We now consider the Steane code, which can detect and correct for arbitrary errors on a single qubit, while encoding one logical state in 7 physical qubits. This means that the encoded state will be comprised of 21 qubits. For this case, we will assume that each qubit is subject to depolarizing noise,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

$$\mathcal{D}(\rho) = (1 - 3p/4)\rho + p/4(\mathsf{X}\rho\mathsf{X} + \mathsf{Y}\rho\mathsf{Y} + \mathsf{Z}\rho\mathsf{Z}) \tag{5.27}$$

Due to the large number of entries for the density matrix of the encoded state, we were unable to directly apply the channel $\mathcal{D}$. Instead, for each qubit in $|\tilde{\psi}_{x=0}\rangle$, we chose to either leave it unchanged, with probability $(1-3p/4)$ or, with probability $p/4$, apply either X, Y or Z. This process is repeated multiple times, and in each case the probability of acceptance is computed using Equation 5.23. The overall probability of acceptance is then estimated by taking the average over all of these runs. The results are shown in Subigure 5.3a.

We considered 12 data points, spread equally in the interval $[0, 1]$, and for each we performed 1000 repetitions of applying noise in order to estimate p_{acc}. The error bars represent confidence intervals for the computed values, assuming a confidence of 95%. Additionally, the orange curve represents the best fit interpolation of the given samples, when assuming a Gaussian model. As we can see, the threshold point appears to be between 0.1 and 0.2. By considering 12 samples in the range between 0.05 and 0.15, and 4000 repetitions per sample, in Subfigure 5.3b, we find that the threshold point is between 0.12 and 0.13.

The simulations were performed in MATLAB, on the Eddie Mark 3 cluster of The University of Edinburgh. The code for our simulations is available on Github [164].



(a) Comparison between encoded and unencoded states for Steane's code.

(b) Threshold for the Steane code for the considered computation.

Figure 5.3: Results for the Steane code.

5.4 Chapter summary and outlook

We have given a simple construction for a fault tolerant quantum verification protocol. In a nutshell, the construction involves taking the original post hoc verification protocol of Morimae and Fitzsimons and encoding it in a CSS error-correcting code. Since the original protocol was not blind, neither is its fault tolerant counterpart. A major open problem that remains to be addressed is whether one can achieve fault tolerant verification of blind quantum computation without resorting to additional assumptions, as in [30, 35, 157]. Specifically, the protocols from [30,35,157] assumed (either implicitly or explicitly) that the noise on the verifier's device is independent of the secret parameters that are used to achieve blindness. Additionally, the noise, on that device, should be uncorrelated with the prover's private system.

Following the discussion in [159], the authors stress that, so far, there is no protocol that simultaneously achieves all of the following properties:

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

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

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

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

Recently, a protocol has been proposed in which a classical client can delegate and verify the computations performed by a quantum server [37]. This protocol, however, relies on certain computational assumptions about whether a quantum computer can solve a particular problem. Therefore, the verifier would not need to worry about introducing errors into the prover's quantum computation, as was the concern in our work, but this comes at the cost of making these computational assumptions. Interestingly, the protocol in [37] also uses post hoc verification as a primitive, except now the prover measures the qubits in the history state and relays the outcomes to the verifier. The preparation of the history state is slightly more complex than in our case since it uses cryptographic one-way functions which introduce some overhead.

Returning to our results, the simulations are encouraging. Given that the obtained thresholds are higher than the error rates observed in current experimental implementations [165–167], a demonstration of the protocol in the near future is likely. The major obstacle to such a demonstration would be the production of these highly entangled history states. The use of CSS codes, however, means that one can encode these states in codes having even higher noise thresholds than the Steane code, such as surface codes [168].

Chapter 6

Limitations of blind quantum computation

> **Penguin:** You're blind as a bat. Sightless and helpless.
> **Batman:** You've got one right.
>
> — Batman The Animated Series, Season 1, Episode 54

In the previous chapter, we briefly mentioned the difficulty in developing a verification protocol that is both fault tolerant and blind, while having a verifier with a single-qubit device. What about the possibility of having such a protocol with a completely classical client? This is possible, with computational security (as opposed to information-theoretic security), provided certain quantum-secure one-way functions exist, as was shown in [37]. In fact, blind quantum computing based on computational security, without verification, has been demonstrated in [169, 170]. However, suppose we are interested in a protocol between a completely classical client and a quantum server that is *information-theoretically* blind. Can such a protocol exist? Essentially, what we are asking is whether it is possible to have a protocol that achieves the same functionality as UBQC, but in which the client is completely classical. In this chapter, we give complexity-theoretic evidence for why the answer is no. In addition to this result, we also provide a complexity-theoretic upper bound for the types of functions which can be evaluated by UBQC-type protocols, in which the client has some quantum capabilities. We show that, under plausible complexity assumptions, this upper bound prohibits the client from delegating NP-hard functions to the server.

This chapter employs many technical concepts from complexity theory and so we refer the reader to Subsection 2.2 from Chapter 2 for a refresher on these topics.

6.1 Main results

Our results are centred around the concept of a *generalised encryption scheme* (GES) introduced by Abadi, Feigenbaum and Killian [171] which is formally defined in Section 6.2. Roughly speaking, a GES is a protocol between a probabilistic polynomial-time *classical* client (BPP) and a computationally unbounded server for computing on encrypted data. The client sends the server a description of some function f, to be evaluated. Using some polynomial-time algorithm denoted E, the client encrypts its input x, and sends $E(x)$ to the server. The server and the client then interact for a number of rounds which is polynomial in the size of x. Finally, using a polynomial-time decryption algorithm, denoted D, the client decrypts the server's responses and obtains $f(x)$ with probability[1] $1/2 + 1/\operatorname{poly}(|x|)$. Importantly, throughout the protocol the server learns, at most, the size of x. Having the server be computationally unbounded means the scheme requires information-theoretic security.

Abadi et al. gave a complexity theoretic upper bound on the types of functions that admit such a scheme. They showed that any function f that can be evaluated using a GES is contained in the class $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. We give a simplified proof of this fact in Section 6.2. They then observed that if NP-hard functions could be computed using a GES then that would imply $\mathsf{NP} \subset \mathsf{NP/poly} \cap \mathsf{coNP/poly}$ and, in particular, $\mathsf{NP} \subset \mathsf{coNP/poly}$. Informally, it seems that determining the truthfulness of this containment is no easier than determining whether $\mathsf{P} = \mathsf{NP}$. However, using a result of Yap [172], the containment implies that the polynomial hierarchy collapses at the third level. In other words, it seems unlikely that NP-hard problems would admit a GES.

What about BQP-hard functions? Just like the case of NP-hard functions, from the Abadi et al. result, we observe that having a GES for BQP-hard functions leads to $\mathsf{BQP} \subset \mathsf{NP/poly} \cap \mathsf{coNP/poly}$. While we would similarly like to argue that such a containment leads to a collapse of the polynomial hierarchy, even $\mathsf{BQP} = \mathsf{P}$ isn't known to lead to such a collapse. However, we can still give some indication as to why the containment seems unlikely.

Suppose that in the GES, the number of rounds of interaction between the client and the server, is upper bounded by a polynomial of *fixed* degree, $d > 0$, in the size of the input[2]. In that case, it can be shown that $\mathsf{BQP} \subset \mathsf{NP/O(n^d)} \cap \mathsf{coNP/O(n^d)}$. For this case, we can prove the following:

Theorem 16. *For each $d \in \mathbb{N}$, there exists an oracle O_d, such that BQP^{O_d} is not contained in $(\mathsf{NP/O(n^d)})^{O_d}$.*

Essentially, the theorem shows that, given access to some black box function, represented by the oracle O_d, there are problems that can be solved efficiently by a quantum computer, but which do not admit a generalised encryption scheme with bounded communication. Since the oracle is parametrised by d, we are in fact giving a family of oracles. The specific problem we define, to prove this separation, is a version of *Simon's problem* [173].

Simon's problem is the following: for an input of size n, and given oracle access to a function $f : \{0,1\}^n \to \{0,1\}^n$, that is promised to be either 1-to-1, or 2-to-1 and periodic[3], decide which is the case. Simon provided a polynomial-time

[1] It should be noted that the $1/2 + 1/\operatorname{poly}(|x|)$ success probability concerns only the case of decision problems.

[2] This is a limitation *only* for the communication between the client and the server. The client itself, is still assumed to be a BPP machine whose running time can be any polynomial.

[3] In other words, there exists a period $s \in \{0,1\}^n$, $s \neq 0^n$, such that for all $x, y \in \{0,1\}^n$, $x \neq y$, it is the case that $f(x) = f(y)$ iff. $x = s \oplus y$.

quantum algorithm for solving this problem, thus showing that it belongs to BQP, relative to the function oracle. For the case in which one should accept when the function is 2-to-1, the problem can be shown to be outside of NP, relative to the oracle. Thus, Simon's problem provides an oracle separation between BQP and NP.

Note that in Simon's construction, the oracle function is the same for all inputs of size n. Such a setup would not be useful in our case, since this problem can be solved with one bit of advice: for all inputs of size n, the advice bit simply specifies whether the function is 1-to-1 or 2-to-1 and periodic. Thus, in our proof of Theorem 16, for each input, the oracle will provide access to a different function. The problem we define, relative to this oracle, is again to decide whether the function is 1-to-1 or 2-to-1 and periodic. However, we can show that by considering a sufficiently large domain for these functions, in other words $f : \{0,1\}^{n^D} \to \{0,1\}^{n^D}$, for sufficiently large D[4], the problem is not contained in $(\mathsf{NP/O(n^d)})^{O_d}$, but is nevertheless, contained in BQP. The proof uses a diagonalization argument and can be found in Section 6.3.

Unfortunately, the same oracle cannot be used to also separate BQP from $\mathsf{NP/poly}$. This is because, in our construction, D is a function of d and to prove a separation with respect to $\mathsf{NP/poly}$, we would have to consider an oracle that works for all possible values of d. We leave that as an open problem. Note that having such an oracle would also separate BQP from AM (since $\mathsf{AM} \subset \mathsf{NP/poly}$), a problem which has eluded complexity theorists for some time [14].

One can argue that oracle results do not constitute compelling evidence regarding the relationship between complexity classes. Indeed, it has been known for a while that there exist oracles such that $\mathsf{P}^O \neq \mathsf{NP}^O$ and oracles such that $\mathsf{P}^O = \mathsf{NP}^O$. Moreover there are non-relativizing results such as $\mathsf{IP} = \mathsf{PSPACE}$ while at the same time there exists an oracle such that $\mathsf{IP}^O \neq \mathsf{PSPACE}^O$. Nevertheless, oracles allow us to study the query complexity of problems in different models of computation. In fact there are situations in practice where computer programs are restricted to making black-box calls to functions in order to determine their properties [174]. Apart from this, oracle results have proven to be insightful for the development of algorithms and complexity theory. Most notably Simon's oracle separation between BPP and BQP led to Shor's algorithm for factoring and computing the discrete logarithm [3]. For more arguments in defence of oracles, see Section 1.3 of [14].

To add more weight to our conjecture, that BQP computations do not admit a GES, we consider the case of having a GES for sampling problems, rather than decision problems. To be more specific, we will consider a GES for BosonSampling. This would imply that a BPP machine could delegate the task of sampling bosons from a linear optics network to a quantum computer, while keeping the description of that network unconditionally hidden and only leaking its size to the server. To be precise, we will be interested in the case of exact sampling, rather than approximate sampling. In other words, if a GES for BosonSampling exists, the client can sample exactly from the BosonSampling distribu-

[4] Specifically, we require D to be large enough so that n^D is greater than the size of the advice string, i.e. $O(n^d)$.

tion mentioned in Subsection 2.2.2, of Chapter 2. Given this assumption, one can prove the following:

Theorem 17. *If* BosonSampling *admits a GES, then for any matrix* $X \in \{-1, 0, 1\}^{n \times n}$*, there exist circuits of size* $2^{n-\Omega\left(\frac{n}{\log n}\right)}$*, making polynomially-sized queries to an* $\mathsf{NP}^{\mathsf{NP}}$ *oracle, for computing the permanent of* X*.*

Computing the permanent of a matrix is a problem known to be #P-hard. By Toda's theorem, this means that if computing the permanent were possible at any level of the polynomial hierarchy, the hierarchy would collapse at that level. Moreover, the best known algorithm for computing the permanent, by Björklund, has a run-time of $2^{n-\Omega\left(\sqrt{n/log(n)}\right)}$ [175]. Prior to that, the leading algorithm for computing the permanent was Ryser's algorithm, developed over 50 years ago, which requires $\mathsf{O}(2^n n)$ arithmetic operations [176]. We therefore conjecture that the circuits of Theorem 17 do not exist and, thus, that there can be no GES for BosonSampling. The proof of Theorem 17 is provided in Section 6.4.

While having a GES for BQP computations seems unlikely, we know that giving the client some minimal quantum capabilities removes this limitation. This is evidenced by the existence of schemes such as UBQC. In the spirit of the Abadi et al. result, it is natural to consider *quantum generalised encryption schemes* (or QGES), in which the client is no longer classical, and investigate the complexity-theoretic upper bounds of functions which admit such a protocol. For the QGES, we are still assuming unconditional security and that the encryption scheme leaks at most the size of the input. However, unlike the GES, the client is now assumed to be a BQP machine[5]. Additionally, the client sends one quantum message to the server at the beginning of the protocol. The rest of the communication is classical. Lastly, we impose a further restriction, known as *offline-ness*. Roughly speaking, an offline protocol is one in which the client does not need to commit to any particular input (of a given size), after having sent the quantum message to the server. In other words, the quantum message only depends on the size of the input. This property is shared by UBQC and all other blind quantum computing protocols. With these assumptions, we can prove the following:

Theorem 18. *If a function,* f*, admits an offline QGES leaking at most the size of the input, then* $f \in \mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$*.*

Note that the class $\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$ can be seen as a quantum analogue of the class $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$, for GES. Again, in the spirit of the Abadi et al. result, one can ask whether NP-complete functions are contained in this class. In other words, does giving quantum capabilities to the client increase the class of functions, that can be securely evaluated, so as to contain NP? We give indication that the answer is no:

[5] The client need not be a full-fledged quantum computer and can in fact only posses a single-qubit preparation device, as in UBQC. However, in proving our results for quantum generalised encryption schemes it is simpler to assume that the client is a BQP machine. It should be noted that our results concerning the computational power of the QGES remain valid even when the client's power is strictly less than BQP, as in UBQC and related blind protocols.

Theorem 19. *Let f be an NP-hard function. If f admits an offline QGES then* $\Pi_3^{\mathsf{P}} \subseteq \mathsf{NP}^{\mathsf{NP}^{\mathsf{PromiseQMA}}}$.

This is as close to a collapse of the polynomial hierarchy as one can reasonably hope to get, given a quantum hypothesis. Hence, while a QGES does (by definition) allow for delegating BQP computations, it seems to be no more useful than the regular GES at delegating NP-hard functions.

Quantum computers could, in principle, solve NP-complete problems quadratically faster than classical computers, thanks to Grover's algorithm [177]. In fact, as is mentioned in [178], there are also NP-complete problems for which quantum computers provide a superpolynomial speedup, at least with respect to the best known classical algorithms. However, our no-go theorem indicates that clients cannot exploit such speedups by delegating the computation to the server, even when allowing some quantum communication, if we also want to keep their inputs hidden in an information-theoretic sense. Proofs of these results can be found in Section 6.5.

6.1.1 Related work

The problem of computing on encrypted data was first considered by Rivest, Adleman and Dertouzos [79], which then led to the development of *homomorphic encryption* and eventually to *fully homomorphic encryption* (FHE). In homomorphic encryption a client has efficient algorithms for encryption *Enc*, and decryption *Dec*, satisfying the property that $Dec(f, x, Eval(f, Enc(x))) = f(x)$, for any function f from some set $\mathcal{C}$, and some algorithm *Eval*. The client will instruct a server to compute an evaluation of f on the encrypted input $Enc(x)$, using the *Eval* procedure. The server must then send this evaluation to the client, who will decrypt it, resulting in $f(x)$. Of course, the server should not be able to infer information about x from $Enc(x)$, a condition which is typically expressed through *semantic security* [179]. The scheme is therefore secure under suitable *cryptographic assumptions*. If the set $\mathcal{C}$ contains all polynomial-sized circuits then the scheme becomes a fully homomorphic encryption scheme. The first such scheme was proposed by Gentry, in [80]. Since then, a number of FHE protocols have been developed, that rely on more standard cryptographic assumptions and having more practical requirements [81–83].

While FHE is similar to GES in many respects, there are also significant differences. For starters, FHE protocols have only one round of interaction between the client and the server, whereas GES allows for polynomially many rounds. Additionally, GES assumes the server is computationally unbounded and hence requires information-theoretic security. In contrast, FHE relies on computational security. More precisely FHE schemes have semantic security against polynomial-time (quantum) algorithms [80].

The problem of *quantum* computing on encrypted data was introduced by Childs [75] and Arrighi and Salvail [180]. Further development eventually led to UBQC [23, 52] and the blind quantum computing scheme of Broadbent [181]. The latter was followed by the construction of the first schemes for quantum fully homomorphic encryption (QFHE) [182, 183].

In the QFHE schemes of [182,183], the server is a BQP machine and the client has some quantum capabilities of its own, although it is not able to perform universal quantum computations. Both the size of the exchanged messages and the number of operations of the client are polynomial in the size of the input. More recently, a QFHE scheme has been proposed in which the client is completely classical [169]. Similar to FHE, these protocols rely on computational assumptions for security [184] and involve one round of back and forth interaction between the client and the server. QFHE with information-theoretic security (and a computationally unbounded server) has been considered by Yu et al. in [185], where it is shown that it is impossible to have such a scheme for arbitrary unitary operations (or even arbitrary reversible classical computations). Newmann and Shi strengthened this result, by showing that it also impossible to have such a scheme for delegating polynomial-sized quantum (or classical) circuits [186].

The possibility of a classical client delegating a blind computation to a quantum server was considered by Morimae and Koshiba [187]. They showed that such a protocol in which the client leaks no information about its input to the server and there is only one round of interaction between them, leads to $\mathsf{BQP} \subseteq \mathsf{NP}$. As mentioned, this is generally considered to be an unlikely containment. Furthermore, Simon's problem can be used to show that $\mathsf{BQP} \not\subseteq \mathsf{NP}$, relative to an oracle. In this chapter, we considered the more general setting of a GES for BQP functions, where the number of rounds can be polynomial in the size of the input and we allow the encryption to leak the size of the input. The question of whether a GES, as defined in Abadi et al., can exist for quantum computations has been raised before by Dunjko and Kashefi in [188].

We stress that our results are specifically about the implausibility of a GES for BQP computations, rather than some other encryption scheme. This is an important consideration, since relaxing the assumptions of the GES framework can allow for the secure delegation of BQP problems. Specifically:

- Relaxing the requirement of information-theoretic security. In that case, as mentioned, the QFHE scheme of Mahadev, or the QFactory protocol of Cojocaru et al. allow for delegating BQP computations, as they rely on computational security [169,170].

- Relaxing the requirement that the client leaks at most the size of the input to the server. If one is allowed to leak more information then, a scheme of Mantri et al. shows that it is again possible to delegate a quantum computation to a server while still retaining some privacy [189].

- Relaxing the requirement that the client interacts with only one quantum server. In this case, the RUV protocol and a number of other schemes allow a classical client to delegate a blind quantum computation to multiple servers as long as these servers share entanglement and are not allowed to communicate during the protocol [31,32,106].

6.2 Generalised encryption scheme

The basis of most of the results in this chapter is the generalised encryption scheme or GES. We state its definition from [171]:

Definition 27 (Generalised Encryption Scheme (GES))**.** *A generalised encryption scheme (GES) is a two party protocol between a classical client C, and an unbounded server S, characterized by:*

- *A function[6] $f : \Delta \rightarrow \Sigma$, where $\Delta, \Sigma \subseteq \{0,1\}^*$.*
- *A cleartext input $x \in Domain(f)$, for which the client wants to compute $f(x)$.*
- *An expected polynomial-time key generation algorithm K which works as follows: for any $x \in Domain(f)$, with probability greater than $1/2+1/poly(|x|)$ we have $(k, success) \leftarrow K(x)$, where $k \in \{0,1\}^{poly(|x|)}$. If the algorithm does not return success then we have $(k', fail) \leftarrow K(x)$, where $k' \in \{0,1\}^{poly(|x|)}$.*
- *A polynomial-time deterministic algorithm E which works as follows: for any $x \in Domain(f)$, $k \in \{0,1\}^{poly(|x|)}$ and $s \in \{0,1\}^{poly(|x|)}$ we have that $y \leftarrow E(x, k, s)$, where $y \in \{0,1\}^{poly(|x|)}$.*
- *A polynomial-time deterministic decryption algorithm D, which works as follows: for any $x \in Domain(f)$, $k \in \{0,1\}^{poly(|x|)}$ and $s \in \{0,1\}^{poly(|x|)}$ we have that $z \leftarrow D(s, k, x)$, where $z \in \{0,1\}^{poly(|x|)}$.*

And satisfying the following properties:

1. *There are m rounds of communication, such that $m = poly(|x|)$. Denote the client's message in round i as c_i and the server's message as s_i.*
2. *On cleartext input x, C runs the key generation algorithm until success to compute a key $(k, success) = K(x)$. This happens before the communication between C and S is initiated, and the key k is used throughout the protocol.*
3. *In round i of the protocol, C computes $c_i = E(x, k, \overline{s}_{i-1})$, where $\overline{s}_{i-1}$ denotes the server's responses up to and including round $i-1$, i.e. $\langle s_0, s_1 ... s_{i-1} \rangle$. We assume that s_0 is the empty string. C then sends c_i to S.*
4. *In round i of the protocol, S responds with s_i, such that $s_i \in \{0,1\}^{poly(|x|)}$.*
5. *At the end of the protocol, C computes $z \leftarrow D(\overline{s}_m, k, x)$ and with probability $1/2 + 1/poly(|x|)$, it is the case that $z = f(x)$.*

[6] Note that f need not be a total function since it is defined on a subset of $\{0,1\}^*$. Whenever we say that $f \in \mathcal{C}$, for some complexity class $\mathcal{C}$, we mean that there exists a total function $g \in \mathcal{C}$ such that $g(x) = f(x)$, for all $x \in Domain(f)$.

Let us provide some intuition for this definition. The purpose of a GES is to allow a client to compute some $f(x)$ which it cannot compute with its own resources. It does this by interacting with a computationally powerful server for a number of rounds which is polynomial in the size of the input. Importantly, the GES allows the client to hide some information about x from the server. We make this statement more precise through the following definition:

Definition 28. *Let X be a random variable denoting the input to a GES (in other words the input will be some string x chosen with probability $Pr(X = x)$) and $T(X)$ a random variable denoting the transcript of the protocol for input X (in other words $T(X)$ is a collection of all messages exchanged between the client and the server, in the GES, on input X). We say that a GES leaks at most $L(X)$ iff. X and $T(X)$ are independent given $L(X)$.*

Finally, we state the main theorem from [171] and provide a simple proof for it:

Theorem 20 (GES leaking size of input)**.** *If a function f admits a GES which leaks at most the size of the input (i.e. $L(X) = |X|$), then $f \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$*[7]*.*

Proof. Suppose that f admits a GES which leaks at most the size of the input. To show that $f \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$ we prove that $f \in \mathsf{NP/poly}$ by constructing an $\mathsf{NP/poly}$ algorithm for f which can also compute the complement of f (thus also proving containment in $\mathsf{coNP/poly}$). We start by first considering the one round case. In other words, the protocol works as follows:

1. The client runs $K(x)$ until success to produce an encryption key k.
2. The client computes the encrypted string $y \leftarrow E(x, k, `')$ (where the last entry is the empty string) and sends it to the server.
3. The server sends a response r.
4. The client decrypts his response obtaining $z \leftarrow D(r, k, x)$. With probability greater than $1/2 + 1/poly(|x|)$ we have that $z = f(x)$.

Given that this is true, consider the following algorithm which takes x as input and produces $f(x)$ with probability greater than $1/2 + 1/poly(|x|)$:

- Denoting $|x| = n$, the algorithm receives as advice some string $x_n \in Domain(f)$ such that $|x_n| = n$ as well as r_n, where r_n is the server's response when being sent $y_n \leftarrow E(x_n, k_n, `')$. Here k_n is simply some key which can be used to encrypt x_n. The only reason we include x_n as part of the advice is so that we can check if $x_n = x$. If this is the case then the algorithm simply decrypts r_n obtaining $f(x)$ with high probability. The next steps assume that $x_n \neq x$.

[7] Note that from the definition of f, $Range(f)$ is not necessarily $\{0, 1\}$, corresponding to a decision problem, but can be any subset of $\{0, 1\}^*$. In this case the correct containment would be $f \in \mathsf{FMA/rpoly}$, where FMA is the relational (or search) version of MA. Since we are interested in decision problems, this distinction is not important and we can simply assume that $Range(f) = \{0, 1\}$. In Section 6.4, we will however consider a GES for sampling problems. In that case, the existence of a GES for a particular sampling problem implies that an $\mathsf{MA/rpoly}$ algorithm can sample from the associated distribution.

- From the assumption that the GES leaks at most the size of the input, there must be some key k, such that $y_n \leftarrow E(x, k, `')$. This is because if there did not exist such a key and the server received y_n he would know that the input could not be x and hence more information would be leaked. More formally, it would mean that the input and the transcript of the protocol are not independent, given the length of the input, since certain transcripts (certain y's) can only occur for certain inputs. Since $|k| = poly(n)$, the algorithm can non-deterministically search for k.

- The algorithm now simply computes $z \leftarrow D(r_n, k, x)$, which by definition of the GES, will be $f(x)$ with probability greater than $1/2 + 1/poly(n)$.

We have therefore given an $\mathsf{MA/rpoly}$ algorithm for computing $f(x)$. The MA part comes from the non-deterministic search for k and the fact that the algorithm is probabilistic. The advice is rpoly because the server's response is drawn from some probability distribution (which depends only on the length of the input). However, it is known that $\mathsf{MA/rpoly} = \mathsf{NP/poly}$, from [190], therefore $f \in \mathsf{NP/poly}$. For the complement of f[8] the client can simply take the output of the protocol and flip the bit (alternatively the decryption function could perform this flip). This implies that $f \in \mathsf{coNP/poly}$ and so $f \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$ (see [171] for more details).

We now need to generalize this to the case where the client and the server interact for a polynomial number of rounds. Because the protocol is leaking at most the size of the input, denoted n, any transcript of the protocol will only depend on n. Therefore we can make the algorithm's advice to be a complete transcript of the protocol drawn from the distribution of all possible transcripts for input of length n. We would then again search non-deterministically for a key k which would make the input x compatible with this transcript. From the definition of the GES this again guarantees that we obtain the right outcome with probability $1/2 + 1/poly(|x|)$.

Note that if the total communication between the client and the server (i.e. the size of the transcript) were bounded by n^d, for some constant $d > 0$, the above argument shows that the functions computable in this setting are contained in $\mathsf{NP/O(n^d)}$. This is because, as we have seen, the transcript is given as advice[9] and so it will also be bounded in length by $O(n^d)$. □

It should be mentioned that if the client is a BPP machine and the functions computable with the GES are contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$ then we should in fact be working with the class $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}}$. However, it is not very difficult to show that $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} = \mathsf{NP/poly} \cap \mathsf{coNP/poly}$. This can be done using a result of Brassard [191] which shows that $\mathsf{P}^{\mathsf{NP} \cap \mathsf{coNP}} = \mathsf{NP} \cap \mathsf{coNP}$, together with Adleman's theorem (which states that $\mathsf{BPP} \subset \mathsf{P/poly}$) [192]:

[8] As mentioned, these containments are valid whenever f outputs one bit, corresponding to a decision problem. For general functions having a larger range, the technically correct containments would be in the relational version of $\mathsf{MA/rpoly}$.

[9] Strictly speaking, the above argument shows that such functions would be contained in MA with *randomized* advice of size n^d. However, the proof that $\mathsf{MA/rpoly} = \mathsf{NP/poly}$ can be adapted to the setting of $O(n^d)$-size advice.

Lemma 8. $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} = \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

Proof. It is clear that $\mathsf{NP/poly} \cap \mathsf{coNP/poly} \subseteq \mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}}$, so that we need only show that $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{NP/poly} \cap \mathsf{coNP/poly}$. To do this, we first use Adleman's theorem [192], that $\mathsf{BPP} \subset \mathsf{P/poly}$, which we know is relativizing and have that $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{P/poly}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}}$. Next, it is easy to show that $\mathsf{P/poly}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{P}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}}$. This is because the advice received by the $\mathsf{P/poly}$ machine can just as easily be obtained from the $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$ oracle. In other words, for any given input x and advice a for the $\mathsf{P/poly}$ machine, the P machine can simply query the $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$ oracle with x in order to obtain the same advice a[10]. It then simulates the $\mathsf{P/poly}$ machine.

We have therefore reduced our problem to showing that $\mathsf{P}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{NP/poly} \cap \mathsf{coNP/poly}$. This can be done by adapting Brassard's proof [191] that $\mathsf{P}^{\mathsf{NP} \cap \mathsf{coNP}} = \mathsf{NP} \cap \mathsf{coNP}$. The essential part of that proof is to show that $\mathsf{P}^{\mathsf{NP} \cap \mathsf{coNP}} \subseteq \mathsf{NP}$, while the containment in coNP follows by complementation. The idea is that for any $\mathsf{P}^{\mathsf{NP} \cap \mathsf{coNP}}$ algorithm, A, deciding some language, we can devise an NP algorithm, NA, which also decides that language.

The NA algorithm will simulate A until it makes a query to the $\mathsf{NP} \cap \mathsf{coNP}$ oracle. At this point NA can non-deterministically guess the response to this query. To do so, note that if some language $L \in \mathsf{NP} \cap \mathsf{coNP}$ then it is the case that $L \in \mathsf{NP}$ and $L^c \in \mathsf{NP}$, where L^c is the complement of L. In other words, there exist non-deterministic algorithms N_L and N_{L^c} for deciding L and L^c, respectively. Assuming A's query is for the language L, NA will simulate N_L, and for each non-deterministic branch of this simulation it will then also simulate N_{L^c}. Since L and L^c are complementary, it cannot happen that both the N_L and the N_{L^c} parts of the branches are accepting. We will therefore have branches in which both N_L and N_{L^c} were rejecting and branches in which either N_L was accepting or N_{L^c} was accepting. These latter branches determine the answer to the query for the $\mathsf{NP} \cap \mathsf{coNP}$ oracle. The NA algorithm will continue simulating A on these branches and reject on all others.

We can see that the above reasoning would also work if the oracle was $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$ and the algorithm NA were an $\mathsf{NP/poly}$ algorithm receiving some advice string whose length is polynomial in the size of the input. Our modified NA can continue to simulate the oracle queries if we assume that the advice it receives is the concatenation of advices received by the $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$ oracle for all queries. Since the number of queries is polynomial, the concatenation will also be polynomially bounded and hence constitutes a valid advice string for an $\mathsf{NP/poly}$ algorithm. Therefore $\mathsf{P}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{NP/poly}$ and through complementation $\mathsf{P}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

Because $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} \subseteq \mathsf{P}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}}$, our result follows immediately. □

[10] The fact that the oracle responds with a single bit (acceptance or rejection) is not a problem, since the P machine can query the oracle for each bit of a.

6.3 GES for BQP

In order to prove Theorem 16 we will construct an oracle using a version of the complement of Simon's problem [173]. Recall that Simon's problem is the following: given a function $f : \{0,1\}^n \to \{0,1\}^n$ (for some $n \in \mathbb{N}$) which is promised to be either 1-to-1 or have Simon's property (f is 2-to-1 and there exists some $s \in \{0,1\}^n$, $s \neq 0^n$, such that for $x \neq y$, $f(x) = f(y)$ iff $x = s \oplus y$), decide which is the case. In particular, for Simon's problem, the deciding algorithm should accept if the function has Simon's property and reject if is a 1-to-1 function. The complement of this problem simply flips these two conditions. If one is not given an explicit description of f but restricts access to this function through an oracle then Simon's problem can be used to separate BPP from BQP. To be precise, the oracle is some function $O : \{0,1\}^* \to \{0,1\}^*$ such that for $n \in \mathbb{N}$, if we consider O restricted to the domain $\{0,1\}^n$, denoted $O_n : \{0,1\}^n \to \{0,1\}^n$, O_n is either a 1-to-1 function or a function satisfying Simon's property. A language which is then contained in BQP^O but not in BPP^O is $L(O) = \{0^n | O_n \text{ is a function with Simon's property}\}$ as shown in [173]. In fact, as we've mentioned before, the complement of this language[11] can be used to separate BQP^O and NP^O [14]. Lemma 9, which we prove below, is essentially a proof of this fact for a slightly different version of the oracle.

In our case, we would ideally like to separate $\mathsf{NP/poly}^O$ and BQP^O. The intuition is the following: instead of considering a function O_n for each input length n, we consider a function O_x, for each input string $x \in \{0,1\}^n$. In other words, for a fixed input length, n, there will be 2^n functions which need to be decided. But the NP^O machine receives only a polynomial amount of advice, which is the same for all of these 2^n functions. Therefore this advice should be insufficient to help the NP^O machine in deciding all of these inputs. Formalizing this intuition for any polynomial is problematic, as will become clear later (see the last paragraph of the proof of Lemma 10). For this reason, we will fix the degree of the polynomial and prove that $\mathsf{BQP}^O \not\subseteq \mathsf{NP/O(n^d)}$. To do this, let us first prove the separation between BQP^O and NP^O, for our specific oracle.

Lemma 9. *There exists an oracle O, based on the complement of Simon's problem, such that* $\mathsf{BQP}^O \not\subseteq \mathsf{NP}^O$.

Proof. The separation of BQP and NP with respect to an oracle has been shown a number of times before, [12,13,193], including with the complement of Simon's problem. However, we prove this lemma for our particular version of Simon's problem where instead of assigning a function to each input length, we assign different functions to different inputs.

We proceed by defining an oracle O and a language which we refer to as the complement of Simon's problem or $coSimon(O)$, such that $coSimon(O) \in \mathsf{BQP}^O$ and $coSimon(O) \notin \mathsf{NP}^O$. We start with the latter as it also clarifies what the

[11] Note that Simon's problem is a promise problem, so when speaking about the complement of $L(O)$ we are in fact referring to $L^c(O) = \{0^n | O_n \text{ is a 1-to-1 function}\}$.

oracle should do:

$$\mathit{coSimon}(O) = \{\langle 1^n, i\rangle \, | i \in \{0,1\}^n \text{ and } f(x) \equiv O(1^n, i, x) \text{ is a 1-to-1 function}\} \tag{6.1}$$

Strictly speaking, the problem we are defining is a promise problem, so the set defined above is the set of *yes* instances to the problem, whereas the set of *no* instances is not the complement but the set:

$$\{\langle 1^n, i\rangle \, | i \in \{0,1\}^n \text{ and } f(x) \equiv O(1^n, i, x) \text{ is a Simon function}\} \tag{6.2}$$

Here, by "*Simon function*" we mean a function having Simon's property.

It is clear from this definition that the oracle O is the one providing the functions for which we want test whether they are 1-to-1 or have Simon's property. Of course, the whole point is to restrict access to the descriptions of those functions and force the algorithm solving the problem to perform queries to the oracle. It is also clear that for any such O, $\mathit{coSimon}(O)$ will be contained in BQP^O since we can just run Simon's algorithm on the given input and flip acceptance and rejection. As is standard in quantum query complexity, we assume that the behaviour of the quantum oracle is to perform the unitary operation $|1^n\rangle |i\rangle |x\rangle |y\rangle \rightarrow^O |1^n\rangle |i\rangle |x\rangle |O(1^n, i, x) \oplus y\rangle$.

The oracle O can be viewed as some function taking as input the tuple (n, i, x) and outputting $f_i(x)$, where $f_i : \{0,1\}^n \to \{0,1\}^n$ is a function which is either bijective or has Simon's property. Essentially n, which is given in unary, specifies the domain size of our functions, i is an index for a particular function and x is the value on which we evaluate f_i. These last two elements of the tuple are specified in binary and the oracle should be defined for all $n \in \mathbb{N}$ and all $i, x \in \{0,1\}^n$. We will denote the set of functions used by the oracle for domain size n as $\mathcal{F}_n$, in other words:

$$\mathcal{F}_n = \{f_i | i \in \{0,1\}^n \text{ and } f_i \text{ is defined as } f_i(x) \equiv O(1^n, i, x)\} \tag{6.3}$$

Next, we construct a so-called *adversarial oracle* O. This just means defining the family of sets $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$, in such a way that every non-deterministic Turing machine using the oracle O fails to decide correctly $\mathit{coSimon}(O)$. The proof will use a diagonalization argument.

Since the set of non-deterministic Turing machines is countable we consider the k'th machine, M_k, and check its behaviour when $n = k + n_0$, for some $n_0 \geq 0$ which we define later on. Suppose we take some index $i \in \{0,1\}^n$, and tentatively make the i'th function in $\mathcal{F}_k$ a 1-to-1 function. By simulating the behaviour of M_k on this input we can check to see whether it accepts or rejects. If it rejects, then we are done, since M_k will incorrectly decide this input. Conversely, if M_k accepts, then by definition there exists a polynomial-sized path, in the non-deterministic computation tree of the machine, which leads to acceptance. We denote this path as π, and denote the length of π as $l = \mathit{poly}(n)$. M_k can make at most l queries to O on this path which we can represent as a list of tuples: $[(x_1, f_i(x_1)), (x_2, f_i(x_2))...(x_l, f_i(x_l))]$, where $x_1, ...x_l$ are the queried variables. An example of such a path is shown in Figure 6.1.

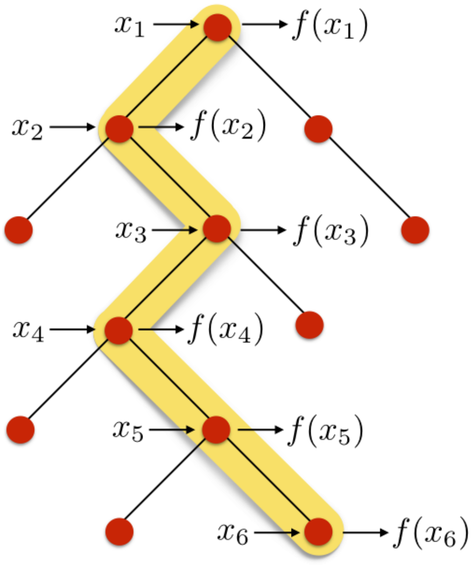


Figure 6.1: Computation tree with queries

We now simply consider a Simon function f' that matched f_i on the queried values, i.e. $f'(x_1) = f_i(x_1), ...f'(x_l) = f_i(x_l)$. How do we know such a function exists? The number of possible bit masks s such that $f(x) = f(x \oplus s)$ is $2^n - 1$ (since 0^n is excluded). By having f' match f_i on the l queried values it must be that f' produces different outputs for each of these values. Therefore for any $i, j \leq l$, $i \neq j$ it must be that $s \neq x_i \oplus x_j$. This means that there are $l(l-1)/2$ values of s which are restricted. But $l = poly(n)$ and since s can take on $2^n - 1$ possible values, if n is sufficiently large so that $2^n - 1 > l(l-1)/2$, then we can simply choose an s which is not restricted. We therefore pick n_0 to be large enough so that $2^n - 1 > l(l-1)/2$ and then take s to be some mask from the available $2^n - 1 - l(l-1)/2$. We thus have a Simon function which produces the same responses to the queries on path π as the 1-to-1 function f_i. If we now just take f_i to be f', then π will still be an accepting path and therefore M_k will decide incorrectly on the input $\langle 1^n, i\rangle$.

Through this construction, all non-deterministic Turing machines will have some input on which they decide $coSimon(O)$ incorrectly, thus $coSimon(O) \notin \mathsf{NP}^O$ concluding the proof. □

Next, we prove:

Lemma 10. *For each $d \in \mathbb{N}$, there exists an oracle O, such that $\mathsf{BQP}^O \not\subset (\mathsf{P/O(n^d)})^O$.*

Proof. To begin with, the class $\mathsf{P/O(n^d)}$ is the class of problems solved by a deterministic polynomial-time Turing machine M, which receives an advice of length $O(n^d)$, when the input is of size $O(n)$ (in our case the input size is $2n$ since we defined n as being the length of inputs to the 1-to-1 and Simon functions).

In contrast to the previous case, instead of having the ability to non-deterministically choose one of exponentially many paths, a polynomial-time

Turing machine M receives some non-uniform information to help it in deciding $coSimon(O)$. Each advice determines a new behaviour for M which can even involve a different sequence of queries to the oracle. What we want to show is that irrespective of what advice M might receive, it still cannot always correctly decide $coSimon(O)$. To do this, we consider functions over a larger domain than just n-bit strings. In other words, for each d we choose $D > d$ such that the set $\mathcal{F}_n$ contains 2^n functions of the form $f : \{0,1\}^{n^D} \to \{0,1\}^{n^D}$. The oracle, which we now denote as O_d, still receives queries of the form $(1^n, i, x)$, where $|i| = n$, but now $|x| = n^D$.

First we need to argue that the problem can still be decided in BQP^{O_d}. This is indeed the case, since expanding the domains of the functions simply changes the running time of the quantum algorithm from $O(n)$ to $O(n^D)$. But since D is just a fixed constant, the algorithm still runs in polynomial time, hence $coSimon(O_d) \in \mathsf{BQP}^{O_d}$.

The harder part is showing $coSimon(O_d) \notin \mathsf{P/O(n^d)}$. As before, we will prove this by diagonalization by considering the set of all (deterministic) Turing machines and showing that no matter which advice the k'th machine receives it cannot correctly decide $coSimon(O_d)$. Care must be taken, as each advice induces a different behaviour and one must consider the oracle so that all possible advice strings lead to failure. This is in contrast with the previous case where we were only interested in the behaviour of one accepting path of the non-deterministic computation tree.

Suppose we take the k'th deterministic polynomial-time Turing machine, M_k, and examine what happens for an input of length $n = k + n_0$, where n_0 will be chosen later (as before). Since the advice is a binary string of length $O(n^d)$ there are $2^{O(n^d)}$ possible advice strings. Whichever one M_k uses it will be the same for all 2^n inputs of length n.

Let us now consider the first index of length n, namely 0^n and assign a 1-to-1 function $f : \{0,1\}^{n^D} \to \{0,1\}^{n^D}$ to this index. We can inspect the behaviour of M_k for f and for each possible advice string. If for more than half of the advice strings M_k rejects, then we keep f at index 0^n. This means that half of all advice strings have been eliminated (there is at least one input on which those strings lead to M_k deciding incorrectly). If, however more than half of all advice strings make M_k accept f, we will attempt to turn f into a Simon function while keeping acceptance for those advice strings. This will again lead to the elimination of (at least) half of all advice strings.

For each advice a_j, where $1 \leq j \leq 2^{O(n^d)}$, M_k will make a sequence of polynomially many queries to f. Denote that sequence of queries together with the responses as:

$$\sigma_j = [(x_{1j}, f(x_{1j})); (x_{2j}, f(x_{2j})); ...(x_{lj}, f(x_{lj}))]$$

where $l = poly(n)$. We now consider a Simon function $f' : \{0,1\}^{n^D} \to \{0,1\}^{n^D}$ such that for all j in which M_k with advice a_j and queries σ_j accepts and for all $t \leq l$, we have that $f'(x_{tj}) = f(x_{tj})$. In other words f' will give identical responses to the queries which make M_k accept. Since t ranges from 1 to $l = poly(n)$ and j ranges from 1 to $2^{O(n^d)}$, the maximum number of variables which are queried is of order $2^{O(n^d)}$. But unlike in the previous lemma, this number is exponential in the size of the input, so how can we be sure that such a Simon function even exists? The trick is that we can choose the domain size through D and make it large enough to accommodate for a Simon function with this property.

As before, because f is bijective, no two queried variables will produce the same answer. Therefore, there cannot be a bit mask s ($|s| = n^D$) relating any pair of the $2^{O(n^d)}$ queries. These will be the restricted values of s. The total number of such values is also of order $2^{O(n^d)}$, however the total number of possible values is $2^{O(n^D)}$. Thus, if we simply choose D such that $2^{O(n^D)} > 2^{O(n^d)}$ then we can find a Simon function f' which matches the responses of f on the $2^{O(n^d)}$ queries.

Hence, for this case if we use f' as the function for index 0^n we will eliminate half of the possible advice strings. Thus, no matter how M_k behaves we are able to eliminate half of all possible advice strings with our first input of length $2n$. Clearly this process can be repeated for the next index and so on until the last index. We are effectively halving the number of potentially useful advice strings with each index. Since we are doing this 2^n times, to eliminate all possible advice strings we just need to ensure that $2^{O(n^d)}/2^{2^n} < 1$ or $2^{O(n^d)} < 2^{2^n}$. To achieve this, simply choose n_0 (recall that $n = k + n_0$) large enough so that the inequality holds.

We therefore have that for all k, and for all possible advice strings, there will always be an input to $coSimon(O_d)$ which is decided incorrectly, hence $coSimon(O_d) \notin \mathsf{P/O(n^d)}$.

Note that the same proof would not work for $\mathsf{P/poly}$. A crucial element in our proof was the fact that we can make D (which determines the size of the domain of each function) to be much larger than d (which determines the length of the advice). But this is only possible because d is fixed from the very beginning. If the advice length could be any arbitrary polynomial then no matter what constant value of D we decided upon for our oracle, there would always be some $d > D$ and hence some polynomial length of the advice string for which the proof does not work. A possible "fix" would be to make D part of the input in some form, so that it too can increase. So if, say, D was included in the input as a $g(n)$ unary string, where g is some monotonically increasing function, then for sufficiently large n, $g(n) > d$. But we immediately notice the problem with this approach. While it is true that in this case the problem cannot be decided in $\mathsf{P/poly}^O$ it would also no longer be decidable in BQP^O either. This is because the query complexity of the quantum algorithm becomes $O(n^{g(n)})$ which is no longer polynomial unless g is the constant function. Hence, proving separation from $\mathsf{P/poly}$ seems to require some non-trivial modification of this proof or a totally different technique. □

Finally, we can prove Theorem 16 by combining the previous two results.

Proof of Theorem 16. The oracle O_d will be defined in the exact same way as for the $\mathsf{P/O(n^d)}$ case. The same reasoning as before applies here. Take the k'th non-deterministic Turing machine and examine its behaviour for some input $\langle 1^n, i\rangle$, where $n = k + n_0$ and n_0 is chosen as before. For each index, we tentatively pick a 1-to-1 function and examine what the machine does for each advice of length $O(n^d)$. If more than half of the advice strings lead to rejection then we keep the bijective function and proceed to the next index. Otherwise we replace it with a Simon function. In this case, for each advice in which the machine accepts, there will be some polynomial-sized path leading to acceptance. We will pick one accepting path for each advice on which the machine accepts and ensure that the Simon function produces the same responses to the queries on those paths. This reduces the problem to the previous case. We know that for sufficiently large D such a function exists and therefore each index will render half of the possible advice strings useless. By also choosing n_0 large enough we can make sure that all advice strings are eliminated and thus that the problem is incorrectly decided by all non-deterministic Turing machines irrespective of the advice (of length $O(n^d)$). Thus $coSimon(O_d) \notin \mathsf{NP/O(n^d)}$, concluding the proof. □

The advantage of this proof technique is that the $\mathsf{NP/O(n^d)}$ case reduces to the $\mathsf{P/O(n^d)}$ case. We therefore conjecture that if there is some modification of our proof allowing it to work for $\mathsf{P/poly}$ it would also work for $\mathsf{NP/poly}$. Of course, this technique relies on the crucial aspect of knowing an asymptotic bound for the polynomial which determines the length of the advice. This allows us to always choose a larger polynomial for the size of the domain of the functions to be queried.

6.4 GES for BosonSampling

To prove Theorem 17, we first need to show a number of results concerning permanents of matrices[12]. The purpose of these results is to eventually show that having an oracle for estimating the squared permanent of a matrix taking values in $\{-1, 0, 1\}$, yields a polynomial-time algorithm, with random access to $n^{\mathsf{O(n)}}$ bits of advice, for exactly computing the permanent. This result together with the assumption that a GES allows the client to sample exactly from the BosonSampling distribution and a result of Björklund, from [175], will allow us to prove Theorem 17.

Let us first introduce some helpful notation: for a matrix, A, we will denote $A^{i,j}$ as the matrix obtained by deleting row i and column j from A.

Lemma 11. *Let $X = (x_{i,j}) \in \{-1, 0, 1\}^{n \times n}$. There exists a matrix $Z = (z_{i,j}) \in \{-1, 0, 1\}^{(n+2)\times(n+2)}$ such that:*

- $z_{n+2,n+2} = 0$
- $Per(Z) = -Per(X)$
- $Per(Z^{n+2,n+2}) = Per(X^{1,1})$

[12] Recall that the permanent of a matrix $M = (m_{i,j})$ is defined as $Per(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$, where S_n is the symmetric group of all permutations of the elements 1 up to n.

Proof. Let Z be the following matrix:

$$Z = \begin{bmatrix} x_{n,n} & x_{n,n-1} & \dots & x_{n,1} & 0 & 0 \\ x_{n-1,n} & x_{n-1,n-1} & \dots & x_{n-1,1} & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ x_{1,n} & x_{1,n-1} & \dots & x_{1,1} & 1 & -1 \\ 0 & 0 & \dots & 1 & 0 & 1 \\ 0 & 0 & \dots & -1 & -1 & 0 \end{bmatrix}$$

We can see that $z_{n+2,n+2} = 0$. It is also not difficult to see that $Per(Z^{n+2,n+2}) = Per(X^{1,1})$, through a Laplace expansion. We now perform a Laplace expansion along the last row of Z, to compute its permanent:

$$Per(Z) = -(Per(Z^{n+2,n+1}) + Per(Z^{n+2,n})) \tag{6.4}$$

But $Per(Z^{n+2,n}) = Per(X^{1,1})$ and $Per(Z^{n+2,n+1}) = Per(X) - Per(X^{1,1})$ hence $Per(Z) = -Per(X)$. □

Lemma 12. *Let $X = (x_{ij}) \in \{-1,0,1\}^{n\times n}$, $Z = (z_{ij}) \in \{-1,0,1\}^{m\times m}$, $m \geq 2$, such that $z_{m,m} = 0$ and $W = (w_{ij}) \in \{-1,0,1\}^{(m+n-1)\times(m+n-1)}$ defined as follows:*

$$W = \begin{bmatrix} z_{1,1} & z_{1,2} & \dots & z_{1,m} & 0 & \dots & 0 \\ z_{2,1} & z_{2,2} & \dots & z_{2,m} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ z_{m-1,1} & z_{m-1,2} & \dots & z_{m-1,m} & 0 & \dots & 0 \\ z_{m,1} & z_{m,2} & \dots & x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ 0 & 0 & \dots & x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & x_{n,1} & x_{n,2} & \dots & x_{n,n} \end{bmatrix}$$

Then, it is the case that:

$$Per(W) = Per(Z)Per(X^{1,1}) + Per(Z^{m,m})Per(X) \tag{6.5}$$

Proof. We will prove this by induction over m. For the $m = 2$ case we have:

$$W = \begin{bmatrix} z_{1,1} & z_{1,2} & 0 & \dots & 0 \\ z_{2,1} & x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ 0 & x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & x_{n,1} & x_{n,2} & \dots & x_{n,n} \end{bmatrix}$$

By doing a Laplace expansion along the first row of W, we get:

$$Per(W) = z_{1,1}Per(X) + z_{1,2}z_{2,1}Per(X^{1,1}) \tag{6.6}$$

Now note that:

$$Z = \begin{bmatrix} z_{1,1} & z_{1,2} \\ z_{2,1} & 0 \end{bmatrix}$$

So $Per(Z) = z_{1,2}z_{2,1}$ and $Per(Z^{2,2}) = z_{1,1}$, therefore:

$$Per(W) = Per(Z)Per(X^{1,1}) + Per(Z^{2,2})Per(X) \tag{6.7}$$

We now assume the relation is true for $m-1$ and prove it for m. To do this, we will first Laplace expand the permanent of W along the first row:

$$Per(W) = \sum_{i=1}^{m-1} z_{1,i}Per(W^{1,i}) + z_{1,m}Per(W^{1,m}) \tag{6.8}$$

The reason for separating the terms this way, is because $W^{1,i}$, with $i < m$, is of the same form as W and we can therefore apply the induction hypothesis. Doing so yields:

$$Per(W) = \sum_{i=1}^{m-1} z_{1,i}(Per(Z^{1,i})Per(X^{1,1}) + Per(Z^{(1,m),(i,m)})Per(X)) + z_{1,m}Per(W^{1,m}) \tag{6.9}$$

Where $Z^{(1,m),(i,m)}$ is obtained from Z by deleting rows 1 and m and columns i and m. Taking common factors we get:

$$\begin{aligned} Per(W) = Per(X^{1,1})\sum_{i=1}^{m-1} z_{1,i}Per(Z^{1,i}) + \\ + Per(X)\sum_{i=1}^{m-1} z_{1,i}Per(Z^{(1,m),(i,m)}) + z_{1,m}Per(W^{1,m}) \end{aligned} \tag{6.10}$$

But notice that:

$$\sum_{i=1}^{m-1} z_{1,i}Per(Z^{(1,m),(i,m)}) = Per(Z^{m,m}) \tag{6.11}$$

since it is a Laplace expansion along the first row of $Z^{m,m}$. This leads to:

$$Per(W) = Per(X^{1,1})\sum_{i=1}^{m-1} z_{1,i}Per(Z^{1,i}) + Per(X)Per(Z^{m,m}) + z_{1,m}Per(W^{1,m}) \tag{6.12}$$

The matrix $W^{1,m}$ is of the same form as W:

$$W^{1,m} = \begin{bmatrix} z_{2,1} & z_{2,2} & \dots & z_{2,m-1} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ z_{m-1,1} & z_{m-1,2} & \dots & z_{m-1,m-1} & 0 & \dots & 0 \\ z_{m,1} & z_{m,2} & \dots & z_{m,m-1} & x_{1,2} & \dots & x_{1,n} \\ 0 & 0 & \dots & 0 & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & x_{n,2} & \dots & x_{n,n} \end{bmatrix}$$

We can see this by taking:

$$Z_{W^{1,m}} = \begin{bmatrix} z_{2,1} & z_{2,2} & \dots & z_{2,m-1} \\ \vdots & \vdots & & \vdots \\ z_{m-1,1} & z_{m-1,2} & \dots & z_{m-1,m-1} \\ z_{m,1} & z_{m,2} & \dots & 0 \end{bmatrix} \qquad X_{W^{1,m}} = \begin{bmatrix} z_{m,m-1} & x_{1,2} & \dots & x_{1,n} \\ 0 & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & x_{n,2} & \dots & x_{n,n} \end{bmatrix}$$

Together with the induction hypothesis this gives us:

$$Per(W^{1,m}) = Per(Z_{W^{1,m}})Per(X^{1,1}_{W^{1,m}}) + Per(Z^{m-1,m-1}_{W^{1,m}})Per(X_{W^{1,m}}) \tag{6.13}$$

Now note that $Per(X_{W^{1,m}}) = z_{m,m-1}Per(X^{1,1}_{W^{1,m}})$ and $Per(X^{1,1}_{W^{1,m}}) = Per(X^{1,1})$, hence:

$$Per(W^{1,m}) = Per(X^{1,1})(Per(Z_{W^{1,m}}) + z_{m,m-1}Per(Z^{m-1,m-1}_{W^{1,m}})) \tag{6.14}$$

But the term in parenthesis is $Per(Z^{1,m})$ so:

$$Per(W^{1,m}) = Per(X^{1,1})Per(Z^{1,m}) \tag{6.15}$$

By substituting this into Equation 6.12, we get:

$$Per(W) = Per(X^{1,1}) \sum_{i=1}^{m-1} z_{1,i}Per(Z^{1,i}) + Per(X)Per(Z^{m,m}) + \\ + z_{1,m}Per(X^{1,1})Per(Z^{1,m}) \tag{6.16}$$

After grouping terms:

$$Per(W) = Per(X^{1,1}) \sum_{i=1}^{m} z_{1,i}Per(Z^{1,i}) + Per(X)Per(Z^{m,m}) \tag{6.17}$$

But:

$$\sum_{i=1}^{m} z_{1,i}Per(Z^{1,i}) = Per(Z) \tag{6.18}$$

Thus:

$$Per(W) = Per(Z)Per(X^{1,1}) + Per(Z^{m,m})Per(X) \tag{6.19}$$

This concludes the proof. □

Using the above lemmas, we can now show the following:

Theorem 21. *Let $\mathcal{O}$ be an oracle that, given a matrix $X \in \{-1, 0, 1\}^{n\times n}$, outputs a number $\mathcal{O}(X)$ such that:*

$$\frac{Per(X)^2}{g} \leq \mathcal{O}(X) \leq gPer(X)^2 \tag{6.20}$$

where $g \in [1, poly(n)]$. Then, for any $X \in \{-1, 0, 1\}^{n\times n}$ there exists a polynomial time algorithm for computing $Per(X)$, which has random access to $n^{\mathsf{O}(n)}$ bits of advice and making $poly(n)$ queries to $\mathcal{O}$.

Proof. The theorem shows that having an oracle for computing a multiplicative approximation for the squared permanent of a matrix, implies the existence of a polynomial time algorithm, with $n^{\mathsf{O(n)}}$ bits of advice, that can compute the permanent exactly.

The proof of this theorem is inspired from a similar result of Aaronson and Arkhipov (see Theorem 4.3 from [71]). In that case, the oracle was outputting a multiplicative approximation of the squared permanent of an arbitrary *real* matrix. In our case, however, the matrices are restricted to entries from $\{-1, 0, 1\}$, which means that we cannot directly use that result.

We prove the theorem by induction. For the case of $n = 1$ the algorithm simply outputs X. Suppose now that we have an algorithm for computing the permanents of $(n-1) \times (n-1)$ matrices with entries from $\{-1, 0, 1\}$. We will use this algorithm to compute the permanent of X. Firstly, if $\mathcal{O}(X) = 0$, then $Per(X) = 0$ and we are done. Additionally, we are going to use the oracle to check if any of the $(n-1) \times (n-1)$ minors of X are non-zero. If all of them are zero, then $Per(X) = 0$ again and we are done. So let's assume that $Per(X) \neq 0$ and $Per(X^{1,1}) \neq 0^{13}$.

We know from Lemma 12, that if we take a matrix $Z = (z_{ij}) \in \{-1, 0, 1\}^{m\times m}$, $m \geq 2$, such that $z_{m,m} = 0$ and then construct:

$$W = \begin{bmatrix} z_{1,1} & z_{1,2} & \dots & z_{1,m} & 0 & \dots & 0 \\ z_{2,1} & z_{2,2} & \dots & z_{2,m} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ z_{m-1,1} & z_{m-1,2} & \dots & z_{m-1,m} & 0 & \dots & 0 \\ z_{m,1} & z_{m,2} & \dots & x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ 0 & 0 & \dots & x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & x_{n,1} & x_{n,2} & \dots & x_{n,n} \end{bmatrix}$$

we have that:

$$Per(W) = Per(Z^{m,m})Per(X) + Per(Z)Per(X^{1,1}) \tag{6.21}$$

[13] The permanent is invariant under permutations of rows and columns. Thus, if X has a non-zero minor, we can simply permute the columns of X, so that $X^{1,1}$ is that minor.

If $Per(W) = 0$ and $Per(Z^{m,m}) \neq 0$, then:

$$Per(X) = -Per(X^{1,1})\frac{Per(Z)}{Per(Z^{m,m})} \tag{6.22}$$

From Lemma 11, we know that for any $n \times n$ matrix X, there exists an $(n+2) \times (n+2)$ matrix Z, such that $z_{n+2,n+2} = 0$, $Per(Z) = -Per(X)$ and $Per(Z^{n+2,n+2}) = Per(X^{1,1})$. If one used such a Z in the construction of W, then it is immediate that $Per(W) = 0$ and that $Per(Z^{n+2,n+2}) \neq 0$. The algorithm will search for such a Z, construct the corresponding W and use the oracle to test if $Per(W) = 0$. If the permanent of W is zero, then one can compute the permanent of X using Equation 6.22.

But how do we search for Z and, furthermore, how do we compute $Per(Z)/Per(Z^{n+2,n+2})$? This is where the advice enters the picture. Note that since $Z \in \{-1, 0, 1\}^{(n+2)\times(n+2)}$, we have that:

$$-(n+2)! \leq Per(Z) \leq (n+2)! \tag{6.23}$$

hence, there are at most $n^{\mathsf{O}(n)}$ possible values for the permanents of Z matrices. Similarly, there are at most $n^{\mathsf{O}(n)}$ possible values for the permanents of the $Z^{n+2,n+2}$ minors of Z matrices.

The advice, to our algorithm, will consist of tuples $(Z_i, Per(Z_i^{n+2,n+2}), f_i = Per(Z_i)/Per(Z_i^{n+2,n+2}))$, comprising of a matrix Z_i together with the permanent of its top left $(n+1)\times(n+1)$ minor and the ratio between that matrix's permanent and the permanent of its top left minor, with $i \leq n^{cn}$, for some constant $c > 0$. Here $Z_i \in \{-1, 0, 1\}^{(n+2)\times(n+2)}$, with the bottom right entry being 0 and $Per(Z_i^{n+2,n+2}) \neq 0$. The matrices in the tuples are such that all possible values for the fraction $f_i = Per(Z_i)/Per(Z_i^{n+2,n+2})$ are covered. From the above discussion, it's clear that there will be at most $n^{\mathsf{O}(n)}$ such tuples. Furthermore, the tuples are sorted in ascending order with respect to those fractions.

For a given matrix X, our algorithm should search through this advice in order to find a matrix Z_i such that $\mathcal{O}(W_i) = 0$, where W_i is constructed from X and Z_i as in Lemma 12. When such a matrix is found, we have that:

$$Per(X) = -Per(X^{1,1})f_i \tag{6.24}$$

But f_i is given in the advice tuple and $Per(X^{1,1})$ is computed recursively by our algorithm, hence we have computed the permanent of X.

To find the matrix Z_i we will perform a binary search over the advice. Suppose that i ranges from 1 to $l = n^{\mathsf{O}(n)}$. Additionally, let $\alpha_i = Per(Z_i^{n+2,n+2})$, so that:

$$Per(W_i) = \alpha_i(Per(X) + f_i Per(X^{1,1})) \tag{6.25}$$

This means that computing $\sqrt{\mathcal{O}(W_i)}/|\alpha_i|$ gives us a multiplicative approximation for $|Per(X) + f_i Per(X^{1,1})|$. Because the f_i values are sorted in ascending order, this means that the function:

$$h(i) = Per(X) + f_i Per(X^{1,1}) \tag{6.26}$$

is monotonically increasing as a function of i and furthermore that there is a unique value i such that $h(i) = 0$. In our case, however, we have a multiplicative approximation for $|h(i)|$, which we denote as $t(i) = \sqrt{\mathcal{O}(W_i)}/|\alpha_i|$. This function will be monotonically decreasing between 1 and i and increasing between i and l. We look for i using binary search as follows: compute $t(v)$ and $t(w)$ for the middle two points, v and w of the interval $[1, l]$. If either of them is 0, then we are done. Otherwise, if $t(v) < t(w)$, then search the interval $[2, v]$, otherwise the interval $[w, l]$. Repeat this recursively until the minimum is found.

Given that the advice is of length $n^{\mathsf{O}(n)}$, the algorithm will query it (and consequently $\mathcal{O}$ as well) at most $\mathsf{O}(n \log n)$ times. Additionally, the construction of each W_i takes time $\mathsf{O}(n^2)$ and since this is done at most $\mathsf{O}(n \log n)$ times, the complexity of this step is $\mathsf{O}(n^3 \log n)$. Finally, the algorithm performs recursive calls to itself (in order to compute $Per(X^{1,1})$) and if we add up the running time of each step we find that the total runtime will be $\mathsf{O}(n^4 \log n)$. □

Theorem 22. *If* BosonSampling *can be solved by a* BPP/rpoly *algorithm, then for any matrix $X \in \{-1, 0, 1\}^{n \times n}$, there exist circuits of size $2^{n-\Omega\left(\frac{n}{\log n}\right)}$, making polynomially-sized queries to an* NP *oracle, for computing $Per(X)$.*

Proof. The starting point for our proof is a result by Björklund [175]. He showed that, for $k \leq n$, the permanent of an $n \times n$ matrix, X, can be expressed as a linear combination of $poly(n)2^{n-k}$ permanents of $k \times k$ matrices. It should be noted that these matrices are not necessarily minors of the original matrix. Nevertheless, each $k \times k$ matrix can be computed efficiently given the input matrix, X.

Our task will be to compute all of these $poly(n)2^{n-k}$ permanents and then perform the linear combination so as to arrive at the permanent of X. We will use the result of Theorem 21 together with the fact that BosonSampling can be solved using a BPP/rpoly algorithm, to show that the permanent of a $k \times k$ matrix can be computed in polynomial time, using random access to $k^{\mathsf{O}(k)}$ bits of advice and polynomially-sized queries to an NP oracle. Crucially, the $k^{\mathsf{O}(k)}$-sized advice will be the same for all $k \times k$ matrices. This means that all permanents can be computed in $poly(n)2^{n-k}$ time with access to $k^{\mathsf{O}(k)}$ bits of advice. The explicit value of k, as a function of n, will be chosen later.

Consider a $k \times k$ matrix, M, and a value $\epsilon > 0$, to be chosen later. We embed ϵM, a scaled version of M, as a submatrix of a BosonSampling matrix A_M. In other words, $A_M \in \mathbb{C}^{m \times k}$, with $m = O(k^5)$ (see [71] for more details). We then have that the probability of detecting one photon in each output mode, a state which we denote as $|1\rangle$, is:

$$p = Per(\epsilon M)^2 = \epsilon^{2k} \cdot Per(M)^2 \tag{6.27}$$

Since $Per(M) \leq k!$, to ensure that $p \leq 1$, it suffices to set $\epsilon = 1/k$.

If a BPP/rpoly machine can simulate the output distribution of a BosonSampling instance, A_M, that means that:

$$\sum_y q_y Pr(\mathcal{A}(A_M, y) \text{ outputs } |1\rangle) = p \tag{6.28}$$

where $\mathcal{A}$ is a BPP algorithm and y is the rpoly advice string, of size polynomial in k, drawn from the distribution $\mathcal{D}_k = \{q_y\}_y$. Note that $\mathcal{D}_k$ only depends on k. If we can estimate p up to multiplicative error in polynomial time (potentially with the help of an NP oracle and $k^{\mathsf{O}(k)}$ bits of advice) then we will effectively be simulating the oracle $\mathcal{O}$ from Theorem 21.

To do this, first note that if $Per(M) \neq 0$, the smallest possible value of p is $1/k^{\mathsf{O}(k)}$. We will therefore consider our advice string to consist of $k^{\mathsf{O}(k)}$ samples from $\mathcal{D}_k$, along with their associated probabilities[14]. Denote the set of these samples as S. This allows us to define:

$$p_{est} = \sum_{y \in S} q_y Pr(\mathcal{A}(A_M, y) \text{ outputs } |1\rangle) \tag{6.29}$$

as a multiplicative estimate for p. But $\mathcal{A}$ is a BPP algorithm, which means that we can view it as a polynomial-time function, $f_{\mathcal{A}}(A_M, y, r)$ which receives as input (apart from A_M and y) a random string $r \in \{0,1\}^{l(k)}$, for some polynomial l. The function will output either 1, corresponding to the cases where $\mathcal{A}$ outputs $|1\rangle$, or 0, corresponding to the cases where $\mathcal{A}$ produces some other output. We therefore have that:

$$Pr(\mathcal{A}(A_M, y) \text{ outputs } |1\rangle) = \frac{1}{2^{l(k)}} \sum_{r \in \{0,1\}^{l(k)}} f_{\mathcal{A}}(A_M, y, r) \tag{6.30}$$

and this leads to our estimate of p being:

$$p_{est} = \frac{1}{2^{l(k)}} \sum_{y \in S} \sum_{r \in \{0,1\}^{l(k)}} q_y f_{\mathcal{A}}(A_M, y, r) \tag{6.31}$$

Computing p_{est} exactly would require summing exponentially many terms. However notice that p_{est} is a sum of exponentially many *positive* numbers, each of which can be evaluated in polynomial time (given access to the $k^{\mathsf{O}(k)}$ advice). For this reason, we can use Stockmeyer's approximate counting method to compute a multiplicative estimate of p_{est} [72]. This will, of course, also yield a multiplicative estimate for p itself.

We have thus given an algorithm for computing a multiplicative estimate of a $k \times k$ matrix M that works in time polynomial in k, performs queries to an NP oracle and has random access to $k^{\mathsf{O}(k)}$ bits of advice. This algorithm can now be viewed as an implementation of the oracle $\mathcal{O}$ from Theorem 21. Using that theorem, leads to a polynomial-time algorithm, with access to an NP oracle and $k^{\mathsf{O}(k)}$-size advice, for computing $Per(M)$ *exactly.* However, since the advice is the same for all $k \times k$ matrices, by repeating this procedure for all $poly(n)2^{n-k}$ $k \times k$ matrices and combining the results we obtain an algorithm for computing $Per(X)$ that runs in time $poly(n)2^{n-k}$, uses $k^{\mathsf{O}(k)}$ bits of advice and makes polynomially-sized queries to an NP oracle.

[14] The fact that the advice has size $k^{\mathsf{O}(k)}$ will ensure that, if p is non-zero, then a y such that $Pr(\mathcal{A}(A_M, y) \text{ outputs } |1\rangle) > 0$ is overwhelmingly likely to be sampled when we generate the advice.

The last step is to convert this algorithm into a circuit. Since the algorithm has a running time of $poly(n)2^{n-k}$, by choosing $k = c\, n/\log n$, for some suitable constant $c > 0$, we will have circuits of size at most $2^{n-\Omega\left(\frac{n}{log(n)}\right)}$. These circuits, must also operate on the $k^{\mathsf{O}(k)}$ bits of advice. Note that $k^{\mathsf{O}(k)} \ll 2^{n-\Omega\left(\frac{n}{log(n)}\right)}$. To reproduce the random access to these bits, we will assume that the gates have unbounded fan-in. The advice bits are therefore hardcoded into the circuit and "fed" into each part of the algorithm that makes use of them. Since only polynomially-many bits of the advice are used at any given point, this can only increase the size of our circuit by a polynomial factor. This concludes the proof. □

With the above result, the proof of Theorem 17 is immediate:

Proof of Theorem 17. We notice that the result of Theorem 22 relativises. In particular, this means that if BosonSampling can be solved by a $\mathsf{BPP}^{\mathsf{NP}}/\mathsf{rpoly}$ algorithm, then for any matrix $X \in \{-1, 0, 1\}^{n\times n}$, there exist circuits of size $2^{n-\Omega\left(\frac{n}{\log n}\right)}$, making polynomially-sized queries to an $\mathsf{NP}^{\mathsf{NP}}$ oracle, for computing $Per(X)$. If we assume that there exists a GES for BosonSampling, this means that there is an $\mathsf{MA}/\mathsf{rpoly}$ algorithm for BosonSampling. But since $\mathsf{MA}/\mathsf{rpoly} \subseteq \mathsf{BPP}^{\mathsf{NP}}/\mathsf{rpoly}$, the result of Theorem 17 follows. □

6.5 Quantum GES

6.5.1 An upper bound for QGES functions

Motivated by the existence of UBQC, and blind quantum computation in general, we would like to know which functions admit a quantum GES or QGES. In a sense, this section is dedicated to 'quantizing' the Abadi et al. result. First of all, we need to define the QGES[15]:

Definition 29 (Quantum Generalised Encryption Scheme (QGES)). *A quantum generalised encryption scheme (QGES) is a two party protocol between a quantum client C, and an unbounded server S, characterized by:*

- *A function $f : \Delta \to \Sigma$, where $\Delta, \Sigma \subseteq \{0, 1\}^*$.*
- *A cleartext input $x \in Domain(f)$, for which the client wants to compute $f(x)$.*
- *An expected polynomial-time key generation algorithm K which works as follows: for any $x \in Domain(f)$, with probability greater than $1/2+1/poly(|x|)$ we have $(k, success) \leftarrow K(x)$, where $k \in \{0, 1\}^{poly(|x|)}$. If the algorithm does not return success then we have $(k', fail) \leftarrow K(x)$, where $k' \in \{0, 1\}^{poly(|x|)}$.*

[15] Our definition of the QGES considers a single quantum message from the client to the server. One could argue that a truly quantum GES would allow for the entire interaction between client and server to be quantum. However, since we are interested in protocols that minimise the amount of quantum interaction between the client and the server, our definition of QGES restricts this interaction to a single quantum message.

- *A quantum polynomial-time algorithm QE, that takes as input classical bit strings and produces as output a quantum state, which works as follows: for any $x \in Domain(f)$, $k \in \{0,1\}^{poly(|x|)}$ we have that $|y\rangle \leftarrow QE(x, k)$, where $|y\rangle \in \mathcal{H}$ and $dim(\mathcal{H}) = 2^{poly(|x|)}$.*

- *A polynomial-time deterministic algorithm E which works as follows: for any $x \in Domain(f)$, $k \in \{0,1\}^{poly(|x|)}$ and $s \in \{0,1\}^{poly(|x|)}$ we have that $w \leftarrow E(x, k, s)$, where $w \in \{0,1\}^{poly(|x|)}$.*

- *A polynomial-time deterministic decryption algorithm D, which works as follows: for any $x \in Domain(f)$, $k \in \{0,1\}^{poly(|x|)}$ and $s \in \{0,1\}^{poly(|x|)}$ we have that $z \leftarrow D(s, k, x)$, where $z \in \{0,1\}^{poly(|x|)}$.*

And satisfying the following properties:

1. *There are m rounds of communication, such that $m = poly(|x|)$. Denote the client's message in round i as c_i and the server's message as s_i.*

2. *On cleartext input x, C runs the key generation algorithm until success to compute a key $(k, success) = K(x)$. This happens before the communication between C and S is initiated, and the key k is used throughout the protocol. C then runs $QE(x, k)$ to obtain a quantum encryption of the input, $|y\rangle$ and sends it to S*[16]*.*

3. *In round i of the protocol, C computes $c_i = E(x, k, \overline{s}_{i-1})$, where $\overline{s}_{i-1}$ denotes the server's responses up to and including round $i-1$, i.e. $\langle s_0, s_1 ... s_{i-1}\rangle$. We assume that s_0 is the empty string. C then sends c_i to S.*

4. *In round i of the protocol, S responds with s_i, such that $s_i \in \{0,1\}^{poly(|x|)}$.*

5. *At the end of the protocol, C computes $z \leftarrow D(\overline{s}_m, k, x)$ and with probability $1/2 + 1/poly(|x|)$, it must be that $z = f(x)$.*

The definition of QGES is similar to both that of the GES and that of UBQC. In fact, it is easy to see the following:

Lemma 13. *UBQC is a QGES for $f \in \mathsf{BQP}$ leaking at most the size of the input x.*

Proof. To show that UBQC is a type of QGES we only need to give implementations for the algorithms K, QE, E and D which are consistent with UBQC and the properties of a QGES leaking at most the size of the input.

- Key generation, K. This is the step in which the client chooses the random angles for the $|+_\theta\rangle$ states that it will send to the server as well as the bits for randomly flipping the measurement outcomes. Thus, K simply takes as input x and produces $M = poly(|x|)$ random angles $\langle \theta_1, \theta_2, ...\theta_M\rangle$ drawn at random from the set $\{0, \pi/4, 2\pi/4, ...7\pi/4\}$ and random bits $\langle r_1, r_2 ... r_M\rangle$. Thus the classical key is $k = \{\langle \theta_1, \theta_2, ...\theta_M\rangle, \langle r_1, r_2 ... r_M\rangle\}$.

[16] It should be noted that it makes no difference for our definition if the client sends the whole state $|y\rangle$ to the server or part of it. The state received by the prover will be mixed and the only important property we require is that the client has a purification of this state.

- Quantum encryption, QE. In this step the client uses to key to prepare the qubits that it will send to the server. In other words $QE(x, k) \rightarrow |+_{\theta_1}\rangle |+_{\theta_2}\rangle ... |+_{\theta_M}\rangle$.

- Computation, E. In round i, the output of E will be the angles $\Delta_i = \langle \delta_{i,1}, \delta_{i,2} ... \delta_{i,k} \rangle$ which the server should use to measure the qubits in layer i. These are computed based on x, k and the result of the server's previous responses. Concretely, for a particular qubit j, E will compute $\delta_j = (-1)^s_j \phi_j + \theta_j + r'_j \pi$, where ϕ_j is the computation angle (in part determined by x), θ_j is the randomization of the measurement and is contained in the key k, and r'_j is the randomization of the measurement outcomes (computed by xor'ing previous measurement outcomes and the random parameter r_j, contained in k).

- Decryption, D. The decryption procedure involves combining the measurement outcomes with the secret parameters in order to extract the output of the computation.

Since this is a BQP computation, the probability of obtaining the correct outcome will be at least $2/3 > 1/2 + 1/poly(|x|)$. This shows that UBQC is a QGES. Additionally, since we already know that UBQC leaks at most the size of x to the server, this particular QGES leaks the same information. □

UBQC is in fact an instance of a more particular type of QGES as it has the property of being an *offline* protocol. What this means is that the client can send a quantum state to the server, representing an encryption of the input, and decide afterwards which input it intends to use. Essentially the client is free to change its mind about the input and not commit to a particular input when the protocol commences. More formally, we define offliness as follows:

Definition 30 (Offline QGES). *Let $x_1, x_2 \in Domain(f)$ be two different inputs for f ($x_1 \neq x_2$) and let $|y\rangle \leftarrow QE(x_1, k)$ be a quantum encryption of x_1 with some compatible key k. A QGES is offline if there exists a polynomial-sized quantum circuit which the client can apply locally on her system after having sent $|y\rangle$ to the server, such that the state of her system and that of the servers are compatible with her having chosen as input x_2.*

One might ask whether this property is not immediately satisfied by a QGES leaking only the size of the input. Indeed, in the classical case, any encrypted string sent by the client to the server must be compatible with all possible inputs of the same size. In other words, there exists an efficient update procedure that the client can perform in order to switch from one input to another. Thus, a GES leaking only the size of the input is implicitly offline in this view.

But the situation is different in the quantum case. The condition that the QGES leaks only the size of the input to the server is equivalent to saying that the density matrix corresponding to the quantum encryption, which the server receives, is the same for all inputs of the same size. Since the density matrix is the same, that means that there exists a unitary (acting on the client's system) which can map one purification of this state, corresponding to one input, to another, corresponding to a different input. This unitary must be verifiable in the sense that the client can check (using a quantum SWAP test) whether the unitary maps to the correct purification. In the classical case, this is sufficient to ensure that the procedure is efficient, since the mapping is just flipping the bits of one encryption key into another. In the quantum case, however, this unitary need not have a polynomial-sized quantum circuit representation.

Offliness simply imposes that such a circuit exist. UBQC trivially satisfies this property, since, no matter which input the client wants to use, it will always send random $|+_\theta\rangle$ states to the server. In other words, the procedure $QE(x, k)$ does not depend explicitly on x, only on $|x|$. Because we know that functions which admit a GES are contained in the class $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$, it is natural to ask what kind of containment we can find for functions which admit an offline QGES such as UBQC. This leads us to Theorem 18, which we now prove:

Proof of Theorem 18. For an input x for which the client wants to compute $f(x)$, consider the state:

$$|\psi_x\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x)|}} \sum_{k_i^x \in \mathcal{K}_C(x)} |k_i^x\rangle_K \, |y_i^x\rangle_E \tag{6.32}$$

Where $\mathcal{K}_C(x)$ is the set of encryption keys which are compatible with x (i.e. could have resulted from the key generation algorithm acting on x, $K(x)$) and $|y_i^x\rangle_E \leftarrow QE(x, k_i^x)$ is the quantum encryption of x using the key k_i^x. The indices K and E specify whether the kets are quantum registers in the key register or the encrypted state register, respectively. Essentially $|\psi_x\rangle$ is the equal superposition of all keys and encryptions of the string x. If we trace out the key register, K, the resulting density matrix is the mixed state of possible encrypted states which the server will receive:

$$\rho_x = \frac{1}{|\mathcal{K}_C(x)|} \sum_{k_i^x \in \mathcal{K}_C(x)} |y_i^x\rangle \langle y_i^x| \tag{6.33}$$

The assumption that the protocol only leaks the size of the input x to the server implies that for any two inputs x_1, x_2 it is the case that $\rho_{x_1} = \rho_{x_2}$. In fact, something stronger is true. Recall that the definition of blindness says that the quantum state of the server's system as well as the distribution of his classical messages are independent of x, given the size of x. Therefore, we should consider a state comprising of his system and his response after receiving the quantum encryption, for a particular input, x:

$$|\phi_x\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x)|}} \sum_{k_i^x \in \mathcal{K}_C(x)} |k_i^x\rangle_K \; U_{ERS} \, |y_i^x\rangle_E \, |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.34}$$

Here, U_{ERS} is the unitary performed by the server in order to produce his response, which will be stored in the response register, initially set to $|0\rangle_R^{\otimes t}$, where $t = poly(|x|)$. This unitary will of course involve the encrypted state provided by the client and the server's private ancilla, denoted as $|anc\rangle_S$ (but will not involve the key register). Note that in the actual protocol, the key register and the encrypted state register are not necessarily entangled. For example, in UBQC they are only classically correlated. However, since we are considering the most general case, we take the state to be entangled. Essentially the client's system can be thought of as a purification system for the encrypted quantum state sent to the server. Additionally, it should be mentioned that the server's response is a classical bit string. Hence, the state in the response register, obtained through the application of the unitary U_{ERS}, will be a probabilistic mixture over computational basis states. This, however, makes no difference in our proof and we can just as well assume that his response is a general quantum state.

If we again trace out the register K we obtain some state $\sigma_x = Tr_K(|\phi_x\rangle\langle\phi_x|)$. This state encodes the distribution of possible messages exchanged by the client and the server in one round of interaction, as well as the server's private system. Since $\rho_{x_1} = \rho_{x_2}$, it is also the case that $\sigma_{x_1} = \sigma_{x_2}$. This is exactly the blindness condition. By the purification principle, this means that if we consider the states:

$$|\phi_{x_1}\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x_1)|}} \sum_{k_i^{x_1} \in \mathcal{K}_C(x_1)} |k_i^{x_1}\rangle_K \; U_{ERS} \, |y_i^{x_1}\rangle_E \, |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.35}$$

$$|\phi_{x_2}\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x_2)|}} \sum_{k_i^{x_2} \in \mathcal{K}_C(x_2)} |k_i^{x_2}\rangle_K \; U_{ERS} \, |y_i^{x_2}\rangle_E \, |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.36}$$

there exists a local unitary, V_K, acting only on the key register which can map $|\phi_{x_1}\rangle$ to $|\phi_{x_2}\rangle$, for any two inputs x_1 and x_2. In fact, let us examine the states of the system for inputs x_1 and x_2 before U_{ERS} is applied:

$$|\chi_{x_1}\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x_1)|}} \sum_{k_i^{x_1} \in \mathcal{K}_C(x_1)} |k_i^{x_1}\rangle_K \;\, |y_i^{x_1}\rangle_E \, |0\rangle_R^{\otimes t} \, |anc\rangle_S = |\psi_{x_1}\rangle_{KE} \, |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.37}$$

$$|\chi_{x_2}\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x_2)|}} \sum_{k_i^{x_2} \in \mathcal{K}_C(x_2)} |k_i^{x_2}\rangle_K \;\, |y_i^{x_2}\rangle_E \, |0\rangle_R^{\otimes t} \, |anc\rangle_S = |\psi_{x_2}\rangle_{KE} \, |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.38}$$

These states are also related by V_K. This can be inferred from the following relations. First, we know that:

$$(V_K \otimes I_{ERS}) \, |\phi_{x_1}\rangle = |\phi_{x_2}\rangle \tag{6.39}$$

And also that:

$$(I_K \otimes U_{ERS}) \, |\chi_{x_1}\rangle = |\phi_{x_1}\rangle \qquad (I_K \otimes U_{ERS}) \, |\chi_{x_2}\rangle = |\phi_{x_2}\rangle \tag{6.40}$$

Therefore:

$$(I_K \otimes U_{ERS}^\dagger)(V_K \otimes I_{ERS})(I_K \otimes U_{ERS}) \, |\chi_{x_1}\rangle = |\chi_{x_2}\rangle \tag{6.41}$$

But $(I_K \otimes U_{ERS}^\dagger)$ and $V_K \otimes I_{ERS}$ commute because they act on different systems and therefore $(I_K \otimes U_{ERS}^\dagger)$ and $(I_K \otimes U_{ERS})$ will cancel out, leaving:

$$(V_K \otimes I_{ERS}) \, |\chi_{x_1}\rangle = |\chi_{x_2}\rangle \tag{6.42}$$

This is also illustrated in the following diagram:

$$\begin{tikzcd}
\left|\chi_{x_1}\right\rangle \arrow[rr, dashed, "V_K \otimes I_{ERS}"] \arrow[d, "I_K \otimes U_{ERS}"'] & & \left|\chi_{x_2}\right\rangle \\
\left|\phi_{x_1}\right\rangle \arrow[rr, "V_K \otimes I_{ERS}"] & & \left|\phi_{x_2}\right\rangle \arrow[u, "I_K \otimes U^{\dagger}_{ERS}"']
\end{tikzcd}$$

But because the protocol is offline, we know that V_K must be a polynomial-sized quantum circuit. Note that even if we trace out the server's ancilla from the states $|\phi_{x_1}\rangle$ and $|\phi_{x_2}\rangle$, the resulting states are still related by V_K on the key register. This allows us to define a QCMA/qpoly algorithm computing any function which admits a QGES. To do so we first introduce some notation. We will consider the following states:

$$|\kappa_x\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x)|}} \sum_{k_i^x \in \mathcal{K}_C(x)} |k_i^x\rangle_K \tag{6.43}$$

$$|\kappa_{x'}\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x')|}} \sum_{k_i^{x'} \in \mathcal{K}_C(x')} \left|k_i^{x'}\right\rangle_K \tag{6.44}$$

which are simply superpositions over the valid keys for two different inputs x and x'. Next we consider:

$$|\phi_x\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x)|}} \sum_{k_i^x \in \mathcal{K}_C(x)} |k_i^x\rangle_K \; U_{ERS} \, |y_i^x\rangle_E \, |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.45}$$

$$|\phi_{x'}\rangle = \frac{1}{\sqrt{|\mathcal{K}_C(x')|}} \sum_{k_i^{x'} \in \mathcal{K}_C(x')} \left|k_i^{x'}\right\rangle_K \; U_{ERS} \left|y_i^{x'}\right\rangle_E |0\rangle_R^{\otimes t} \, |anc\rangle_S \tag{6.46}$$

which include the encrypted states and the server's response. Lastly, we trace out the server's ancilla from both these states resulting in:

$$\omega_x = Tr_S(|\phi_x\rangle \langle\phi_x|) \tag{6.47}$$

$$\omega_{x'} = Tr_S(|\phi_{x'}\rangle \langle\phi_{x'}|) \tag{6.48}$$

From the above argument the two states $|\kappa_x\rangle$ and $|\kappa_{x'}\rangle$ and the two states ω_x and $\omega_{x'}$ are related through the same polynomial-sized quantum circuit V_K acting only on the key register.

We can now present the algorithm. Let us first consider the one round case. The algorithm would work as follows:

1. The input to the algorithm is some string x for which we want to compute $f(x)$.

2. The algorithm receives as advice the string x' which is simply some string of

the same length as x. Additionally, it receives the state $\omega_{x'}$. It is clear that both of these only depend on $|x|$ and have a length which is polynomial in $|x|$ hence constituting a valid advice.

3. From the definition of the key generating function, the algorithm can efficiently produce the states $|\kappa_x\rangle$ and $|\kappa_{x'}\rangle$.

4. The classical witness is a description of the quantum circuit $V_K^\dagger$.

5. The algorithm tests that $V_K^\dagger$ maps $|\kappa_{x'}\rangle$ to $|\kappa_x\rangle$. This can be done through a quantum SWAP test.

6. Use $V_K^\dagger$ to map $\omega_{x'}$ to ω_x.

7. By measuring the response register of ω_x, the algorithm obtains the response that the server would have produced in an interaction with the client in the QGES protocol. Applying the decryption algorithm to this response will yield the correct result $f(x)$ with high probability.

The probability of success of the algorithm can be boosted by providing polynomially many copies of $\omega_{x'}$ as advice and performing multiple SWAP tests. Additionally this algorithm can be made to compute the complement of $f(x)$ as well which would gives us a $\mathsf{coQCMA/qpoly}$ containment.

For the general case of polynomially many rounds, the only difference is that the state $\omega_{x'}$ would also be entangled with a superposition of all possible transcripts of the protocol. Since we know that transcripts are polynomially bounded in length this is still a valid advice state. The application of $V_K^\dagger$ would map this state to one containing the transcripts for input x. When the state is measured the algorithm will obtain a sample transcript of the interaction between the client and the server. This is then used together with the decryption algorithm to produce $f(x)$. By the definition of the QGES we know that the possible transcripts are such that the correct $f(x)$ is obtained with high probability.

Note that depending on how we define the offline property of the protocol we can get containments in different classes. For example, in this proof we have assumed that while there is a polynomial-sized circuit allowing the client to map from one input to another the client might not be able to arrive at this circuit in polynomial time for any possible pair of inputs. This is why the description of the circuit is given as a witness (since the client can always test the validity of this circuit). However, if we additionally assumed that $V_K^\dagger$ can always be obtained efficiently by the client, then we would no longer need the witness and we would have that $f \in \mathsf{BQP/qpoly}$. □

Of course, just like with Theorem 20, one can ask whether the class of interest should in fact be $\mathsf{BQP}^{\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}}$ since the BQP client is using the QGES as an oracle. But just as $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} = \mathsf{NP/poly} \cap \mathsf{coNP/poly}$ it is the case that:

Lemma 14. $\mathsf{BQP}^{\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}} = \mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$

Proof. This proof is similar to the one showing that $\mathsf{BPP}^{\mathsf{NP/poly} \cap \mathsf{coNP/poly}} = \mathsf{NP/poly} \cap \mathsf{coNP/poly}$. Just like in that case, the inclusion $\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly} \subseteq \mathsf{BQP}^{\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}}$ is immediate and we need only show that $\mathsf{BQP}^{\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}} \subseteq \mathsf{QCMA/qpoly}$. The containment in $\mathsf{coQCMA/qpoly}$ follows by complementation.

Consider a quantum algorithm QA for deciding problems in $\mathsf{BQP}^{\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}}$. We will show that this algorithm can be simulated by a $\mathsf{QCMA/qpoly}$ algorithm, denoted NQA. Since BQP, QCMA and coQCMA have bounded error in deciding problems, we can assume, from standard amplification techniques, that this error is of order $2^{-poly(n)}$, where n is the size of the input. We will also assume that for all quantum algorithms measurements are postponed until the end of the circuit.

We will treat the case without advice first, and then explain how to deal with the quantum advice at the end. To start with, NQA will simulate QA until it makes a query to the oracle. In the standard definition of oracles the oracle is just a classical function that solves a decision problem. However, when dealing with quantum algorithms such as QA it is also possible to speak of quantum oracles, where the oracle can be viewed as some unitary operation (technically a sequence of unitary operations for each possible input length, see [194] for more details) which QA can query even in superposition. Our result will cover this more general case of quantum oracles. We would therefore like the NQA algorithm to be able to simulate this quantum oracle.

Firstly, just like in the classical case we have that if some language $L \in \mathsf{QCMA} \cap \mathsf{coQCMA}$ then $L \in \mathsf{QCMA}$ and $L^c \in \mathsf{coQCMA}$, where L^c is the complement of L. This means that there exist polynomial-sized quantum circuits Q_L and Q_{L^c} which take some input x along with classical witnesses w_1 and w_2, respectively, and decide correctly, when the output is measured, with probability at least, $1 - 2^{-poly(|x|)}$. In other words, Q_L receives as input $|x\rangle |w_1\rangle |0^m\rangle$ and Q_{L^c} receives as input $|x\rangle |w_2\rangle |0^m\rangle$, respectively, where $m = poly(|x|)$. If we were to run both Q_L and Q_{L^c} on x, because L and L^c are complementary, the output qubits, when measured, will also be complementary with high probability.

Assume that Q_L and Q_{L^c} are circuits which act on $t = poly(|x|)$ qubits. We define a new quantum circuit called $SimQuery$ which operates on $2t + 1$ qubits. $SimQuery$ applies Q_L to the first t qubits and Q_{L^c} to the next t qubits. It then applies a Pauli X to the output qubit of Q_{L^c} and a CCNOT operation from the output qubits of Q_L and Q_{L^c} onto the the $2t+1$'th qubit. It then applies X again to the output qubit of Q_{L^c} and then $Q_L^\dagger$ and $Q_{L^c}^\dagger$ on the first $2m$ qubits. An illustration of this circuit (acting on the $|00...0\rangle$ input) is given in Figure 6.2.

The CCNOT operation flips its target qubit if the control qubits are in the state $|11\rangle$. The effect of the first Pauli X is to flip the outcome when the control qubits are in the state $|10\rangle$. Roughly speaking, $SimQuery$ will flip the final qubit if Q_L accepts and Q_{L^c} rejects. The reason for then applying the two circuits in reverse is to 'uncompute' their result and only leave the $2t + 1$ qubit flipped whenever Q_L accepts and Q_{L^c} rejects.

We can now explain how NQA can use $SimQuery$ to simulate a query of QA. Suppose QA queries the oracle for some input x testing to see if it is in L, for some $L \in \mathsf{QCMA} \cap \mathsf{coQCMA}$. In other words, $|x\rangle |0\rangle \rightarrow |x\rangle |1\rangle$, with high probability,

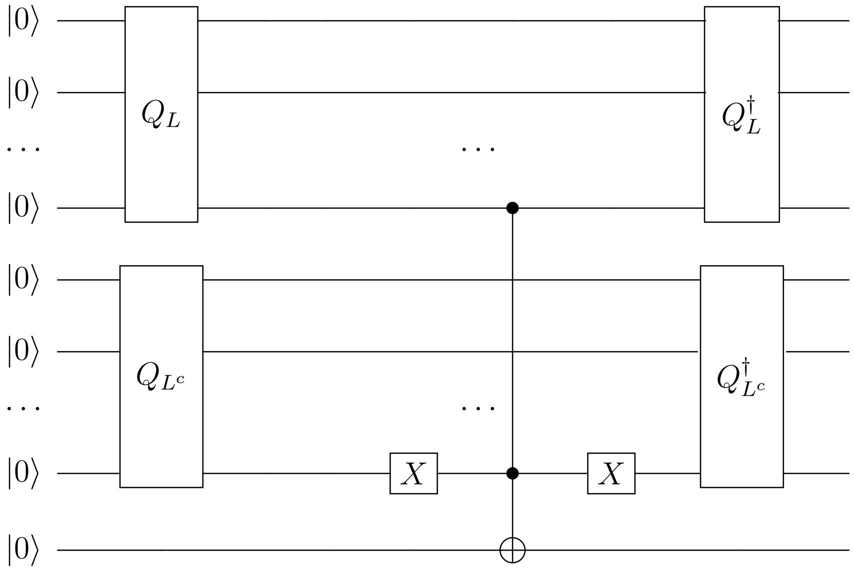


Figure 6.2: Quantum circuit for *SimQuery* acting on the $|00...0\rangle$ input.

if $x \in L$ and $|x\rangle |0\rangle \to |x\rangle |0\rangle$, with high probability, if $x \notin L$. *NQA* will then run *SimQuery* with input $|x\rangle |w_1\rangle |0^m\rangle |x\rangle |w_2\rangle |0^m\rangle |0\rangle$, where w_1 and w_2 are the witnesses from before and $m = poly(|x|)$. The effect of this will be to flip the final qubit if $x \in L$ and leave it unchanged if $x \notin L$, with high probability. This is true because of the complementarity of Q_L and Q_{L^c} (when one accepts the other rejects and viceversa, except with small probability).

This procedure will simulate the query that *QA* performs. *NQA* then uses the last qubit from *SimQuery* as the query response qubit and continues to do this for all other queries of *QA* and otherwise simulate *QA* exactly. Note that each simulated query has some small probability of not matching the actual query of *QA* when a measurement is performed. However, as mentioned, this probability is exponentially small. Since there are polynomially many queries in total, by a union bound, the probability that at least one simulated query behaves incorrectly will still be exponentially small.

Adding quantum advice to this picture does not change much. Just like in the classical case, we can assume that *NQA* receives as advice a concatenation of all advice states used by the oracle of *QA*. The quantum circuits Q_L, Q_{L^c} and *SimQuery* are then extended with polynomially many qubits to act on this advice as well.

It is therefore the case that $\mathsf{BQP}^{\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}} \subseteq \mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$ and our result follows directly. □

Note that in the QGES, the client need not be a fully BQP-capable machine (and indeed, for UBQC the only quantum capabilities of the client are to prepare single qubits).

What happens if we drop the "offline" requirement for this scheme? As mentioned, that would imply that the mapping from one purification of the encrypted quantum state to another can in principle be any unitary operation so long as the client can check that the correct mapping was performed (with high probability).

Having such a weak restriction on this unitary makes it very difficult to impose an upper bound on the types of computations that are allowed by such a scheme. Indeed, the offliness condition plays a crucial role in our proof of Theorem 18. At the same time, it is arguably a very natural condition to have in any realistic protocol. We therefore leave the online case as an open problem.

6.5.2 QGES and NP-hard functions

Theorem 18 can be viewed as a quantum version of Theorem 20 which, as mentioned, was used by Abadi et al. to show that there can be no GES for NP-hard functions unless the polynomial hierarchy collapses. As we have stated before, for quantum computers, the possibility of delegating NP-complete problems makes, arguably, even more sense since Grover's algorithm offers a quadratic speed-up in solving such problems [177]. Alas, we show that even with a QGES, delegating such problems seems unlikely.

Since we have shown that functions which admit an offline QGES are contained in $\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$, and since if $\mathsf{NP} \subset \mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$ then $\mathsf{coNP} \subset \mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$, to prove Theorem 19, it suffices to show that if $\mathsf{coNP} \subset \mathsf{QCMA/qpoly}$ then, informally speaking, the polynomial hierarchy "comes about as close to collapsing as one could reasonably hope to prove given a quantum hypothesis"—and more specifically, that $\Pi_3^{\mathsf{P}} \subseteq \left(\Sigma_2^{\mathsf{P}}\right)^{\mathsf{PromiseQMA}}$. Here a PromiseQMA oracle means an oracle for some PromiseQMA-complete promise problem $(\Pi_{\text{YES}}, \Pi_{\text{NO}})$, whose responses can be arbitrary on inputs $x \notin \Pi_{\text{YES}} \cup \Pi_{\text{NO}}$ that violate the promise. We don't even demand that the oracle's responses, on promise-violating inputs, be consistent from one query to the next. On the other hand, it does need to be possible to query the PromiseQMA oracle on some promise-violating inputs, without such queries causing the entire algorithm to abort.

The starting point for all such collapse results, of course, is the Karp-Lipton Theorem [195], which says that if $\mathsf{NP} \subset \mathsf{P/poly}$ then $\Pi_2^{\mathsf{P}} \subseteq \Sigma_2^{\mathsf{P}}$, and hence the polynomial hierarchy collapses to the second level. An easy extension of the Karp-Lipton theorem, proved by Yap [172], which we now reprove for completeness, shows that if $\mathsf{coNP} \subset \mathsf{NP/poly}$, then PH collapses to the *third* level.

Proposition 1. *If* $\mathsf{coNP} \subset \mathsf{NP/poly}$*, then* $\Pi_3^{\mathsf{P}} \subseteq \Sigma_3^{\mathsf{P}}$.

Proof. Abusing notation, here and later in this section we'll use Φ, Ψ, etc. to refer not only to SAT[17] instances but to strings encoding those instances. Also, if (say) $\Phi(x, y, z)$ is a SAT instance taking multiple strings as input, then by $\Phi(x, y)$, we'll mean the instance obtained from Φ by fixing the variables in x and y, and leaving only the variables in z as free variables.

A Π_3^{P} sentence has the form

$$S = \text{``}\forall x \exists y \forall z \; \Phi(x, y, z)\text{''}$$

[17] SAT stands for *boolean satisfiability* and a SAT instance, in our context, simply refers to a boolean formula. The problem of deciding whether a given boolean formula admits and assignment of variables that evaluates to true (i.e. a satisfying assignment) is NP-complete.

where x, y, z are strings of some given polynomial size, and Φ is a polynomial-time computable predicate (without loss of generality, a SAT instance). Under the stated hypothesis, we need to show how to decide S in Σ_3^{P}.

Let C be the assumed $\mathsf{NP/poly}$ algorithm for coNP, and let a be its advice. Then by hypothesis, for all SAT instances Ψ, if Ψ is unsatisfiable then there exists a witness w such that $C(\Psi, w, a)$ accepts, while if Ψ is satisfiable then $C(\Psi, w, a)$ rejects for all w.

Our Σ_3^{P} rewriting of S is now as follows:

There exists an advice string a, such that

(1) **(Completeness of C)** For all SAT instances Ψ, either there exists a z that satisfies Ψ, or else there exists a w such that $C(\Psi, w, a)$ accepts.

(2) **(Soundness of C)** For all SAT instances Ψ, all satisfying assignments z for Ψ, and all w, the procedure $C(\Psi, w, a)$ rejects.

(3) **(Truth of S)** For all x, there exists a y as well as a witness w such that $C(\neg\Phi(x, y), w, a)$ accepts. (In other words, there is no z that makes $\Phi(x, y, z)$ false.)

□

Proposition 1 is what we seek to imitate in the quantum setting, getting whatever leverage we can from the weaker assumption $\mathsf{coNP} \subset \mathsf{QCMA/qpoly}$.

Note that, had we assumed (say) $\mathsf{coNP} \subset \mathsf{QCMA/poly}$, it would be routine to mimic the usual Karp-Lipton argument, merely substituting the class $\mathsf{PromiseQCMA}$ for NP at appropriate points in the proof of Proposition 1. This would give us the collapse $\Pi_3^{\mathsf{P}} \subseteq \left(\Sigma_2^{\mathsf{P}}\right)^{\mathsf{PromiseQCMA}}$. However, the fundamental difficulty we face is that our hypothesized nonuniform algorithm uses *quantum advice states*. And while a $\mathsf{PromiseQMA}$ machine can simply guess a quantum advice state σ, it can't then pass σ to an oracle, at least not with conventional oracle calls. (To allow the passing of quantum states to oracles, we would need *quantum oracles*, as studied for example by Aaronson and Kuperberg [194].)

To get around this difficulty, we'll rely essentially on a 2010 result of Aaronson and Drucker [196, 197], characterizing the power of quantum advice. These authors proved that $\mathsf{BQP/qpoly}$ is contained in $\mathsf{QMA/poly}$—and even more strongly,

Theorem 23. $\mathsf{BQP/qpoly} = \mathsf{YQP/poly}$.

Here YQP, known as Yoda quantum polynomial-time, is the class of problems solvable by a polynomial-time quantum algorithm with help from a polynomial-size *untrusted* quantum advice state that depends only on the input length n. In other words, Theorem 23 says that we can simulate trusted quantum advice by trusted classical advice combined with untrusted quantum advice, by using the classical advice to verify the quantum advice for ourselves.

By using Theorem 23, to replace a quantification over quantum advice states by a quantification over classical advice strings, Aaronson and Drucker were able to show the following:

Theorem 24. *If* $\mathsf{NP} \subset \mathsf{BQP/qpoly}$*,* $\Pi_2^{\mathsf{P}} \subseteq \mathsf{QMA}^{\mathsf{PromiseQMA}}$*.*

By adapting our argument from later in this section, one can actually improve Theorem 24, to show that if $\mathsf{NP} \subset \mathsf{BQP/qpoly}$ then $\Pi_2^{\mathsf{P}} \subseteq \mathsf{NP}^{\mathsf{PromiseQMA}}$. In any case, we now seek a common generalization of the proofs of Proposition 1 and Theorem 24, to get a collapse from the assumption $\mathsf{coNP} \subset \mathsf{QCMA/qpoly}$.

As Aaronson and Drucker [197] pointed out, a simple extension of their proof of Theorem 23 gives $\mathsf{QCMA/qpoly} \subseteq \mathsf{QMA/poly}$, and even the following.

Theorem 25. $\mathsf{QCMA/qpoly} = \mathsf{YQ} \cdot \mathsf{QCMA/poly}$.

Here the $\mathsf{YQ}\cdot$ operator simply adds untrusted quantum advice to whatever (quantum) complexity class it acts on. Thus $\mathsf{YQ} \cdot \mathsf{BQP} = \mathsf{YQP}$, while for completeness:

Definition 31. $\mathsf{YQ} \cdot \mathsf{QCMA}$ *is the class of languages* L *for which there exist polynomial-time quantum algorithms* C *and* V*, such that for all input lengths* n*:*

- *There exists a polynomial-size quantum advice state* σ_n *such that* $V(0^n, \sigma_n)$ *accepts with probability at least* 0.99*. If* $V(0^n, \sigma)$ *accepts with probability at least* 0.98*, then we call the advice state* σ *"valid" for input length* n*.*

- *For all inputs* $x \in \{0,1\}^n \cap L$ *and all valid* σ*, there exists a polynomial-size classical witness* w *such that* $C(x, w, \sigma)$ *accepts with probability at least* $2/3$ *.*

- *For all inputs* $x \in \{0,1\}^n \setminus L$*, all classical witnesses* w*, and all valid* σ*, we have that* $C(x, w, \sigma)$ *accepts with probability at most* $1/3$*.*

In what follows, we'll need one additional observation about the proof of Theorem 25. Namely, in our $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$ simulation of $\mathsf{QCMA/qpoly}$, without loss of generality we can choose the classical advice string $a = a_n$ in such a way that there's essentially just *one* valid quantum advice state compatible with a. Or more precisely: we can ensure that, for all ρ_1, ρ_2 such that $V(0^n, a, \rho_1)$ and $V(0^n, a, \rho_2)$ both accept with probability at least 0.98, and all x and w, we have (say)

$$|\Pr[C(x, w, a, \rho_1) \text{ accepts}] - \Pr[C(x, w, a, \rho_2) \text{ accepts}]| < \frac{1}{20}.$$

This is because Theorem 25, like Theorem 23, is proven via the method of "majority-certificates," in which given a polynomial-time quantum algorithm Q, one verifies that an unknown quantum state ρ leads to approximately the desired values of $\Pr[Q(x, \rho) \text{ accepts}]$ for *each* of exponentially many different inputs x, via a measurement of ρ that takes only polynomial time. We note that this works only because of special structure in ρ—but for any state σ, there exists another state ρ that has the requisite special structure, as well as a modified quantum algorithm Q', such that

$$\Pr[Q'(x, \rho) \text{ accepts}] \approx \Pr[Q(x, \sigma) \text{ accepts}]$$

for all x.

We're finally ready to prove Theorem 19.

Proof of Theorem 19. Essentially, we are going to show that if $\mathsf{coNP} \subset \mathsf{QCMA/qpoly}$, then $\Pi_3^{\mathsf{P}} \subseteq \left(\Sigma_2^{\mathsf{P}}\right)^{\mathsf{PromiseQMA}}$. A Π_3^{P} sentence has the form

$$S = \text{“}\forall x \exists y \forall z \ \Phi\left(x, y, z\right)\text{”}$$

where x, y, z are strings of some given polynomial size, and Φ is a polynomial-time computable predicate. Under the stated hypothesis, we need to show how to decide S in $\mathsf{NP}^{\mathsf{NP}^{\mathsf{PromiseQMA}}}$.

By Theorem 25, the hypothesis $\mathsf{coNP} \subset \mathsf{QCMA/qpoly}$ is equivalent to $\mathsf{coNP} \subset \mathsf{YQ} \cdot \mathsf{QCMA/poly}$. In other words: we can assume that there exists a polynomial-time quantum algorithm $C\left(\Phi, w, a, \sigma\right)$, which takes as input a SAT instance Φ, a classical witness w, a classical advice string a, and a quantum advice state σ. Assuming a and σ are the correct $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$ advice, C checks whether w is a witness to Φ's *un*satisfiability. This is a sound and complete proof system for coNP, in the sense that, again assuming the correctness of a and σ,

(i) for every unsatisfiable Φ, there exists a w such that $C\left(\Phi, w, a, \sigma\right)$ accepts with probability at least $2/3$,

(ii) for no satisfiable Φ does there exist a w such that $C\left(\Phi, w, a, \sigma\right)$ accepts with probability more than $1/3$.

Moreover, as discussed above, there exists an a such that the state σ is essentially unique, in the sense that

$$\Pr\left[C\left(\Psi, w, a, \rho_1\right) \text{ accepts}\right] \approx \Pr\left[C\left(\Psi, w, a, \rho_2\right) \text{ accepts}\right]$$

for all valid ρ_1, ρ_2.

Our job is to rewrite S as an $\mathsf{NP}^{\mathsf{NP}^{\mathsf{PromiseQMA}}}$ sentence. Our rewriting will be as follows:

There exists a classical advice string a such that

(1) for all valid quantum advice states ρ_1, ρ_2, all SAT instances Ψ, and all assignments w, we have

$$\left|\Pr\left[C\left(\Psi, w, a, \rho_1\right) \text{ accepts}\right] - \Pr\left[C\left(\Psi, w, a, \rho_2\right) \text{ accepts}\right]\right| < \frac{1}{10}.$$

(In words: the classical advice string a uniquely determines the behavior of C, once we find a valid quantum advice state σ that's compatible with a.)

(2) For all SAT instances Ψ, there exists a valid quantum advice state σ, as well as either an assignment z that satisfies Ψ, or else a classical witness w such that $C\left(\Psi, w, a, \sigma\right)$ accepts with probability at least $2/3$.

(In words: the advice a leads to a complete procedure for deciding the class coNP, and specifically the UNSAT problem, in $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$. That is, once we find a valid advice state σ, the quantum algorithm C then accepts every SAT instance Ψ that has no satisfying assignment.)

(3) For all valid quantum advice states σ, all SAT instances Ψ, all z, and all w, if z satisfies Ψ then $C(\Psi, w, a, \sigma)$ rejects with probability at least 2/3.

(In words: a leads to a *sound* procedure for deciding UNSAT. That is, once we find a valid σ that's compatible with a, the quantum algorithm C accepts no SAT instance Ψ that *has* a satisfying assignment.)

(4) For all x, there exists a valid quantum advice state σ, as well as a y and a classical witness w, such that $C(\neg\Phi(x, y), w, a, \sigma)$ accepts with probability at least 2/3.

(In words: C verifies that for all x, there exists a y such that $\neg\Phi(x, y)$ is unsatisfiable. In other words, C verifies that for all x, there exists a y such that for all z, we have $\Phi(x, y, z)$. In other words, C verifies the truth of the Π_3^{P}-sentence S.)

As a point of clarification, whenever we quantify over quantum states (such as σ), we can actually take a tensor product of a polynomial number of copies of the states, as needed. Of course, we can't rule out the possibility that we'll get a state that's entangled across all the registers. Fortunately, though, we don't use the witness state registers in such a way that it ever matters whether they're entangled or not.

As a second point of clarification, in forming the statement above, whenever we have a condition that involves a quantum algorithm (say, V or C) accepting with probability at least 2/3, it's implied that if the condition fails, then the algorithm accepts with probability at most 1/3. This makes verifying the condition a quantum polynomial-time operation. Likewise, for part (1), it can be guaranteed that there exists an a such that, for all ρ_1, ρ_2 consistent with a and all Ψ and w, the difference between the two acceptance probabilities is at most (say) 1/20. In such a case, one can verify in quantum polynomial time that the difference is at most 1/10.

With these clarifications, it's not hard to see that we've given an $\mathsf{NP}^{\mathsf{NP}^{\mathsf{PromiseQMA}}}$ procedure. The NP at the bottom guesses the classical advice string a. The NP in the middle guesses Ψ for part (2) and x for part (4), and is not needed for parts (1) and (3). Finally, the $\mathsf{PromiseQMA}$ on top guesses the quantum advice state σ (or ρ_1, ρ_2 for part (1)), as well as Ψ, w, y, and z as needed. Crucially, quantum states are only ever guessed in the topmost, $\mathsf{PromiseQMA}$ quantifier: once guessed, they never need to be passed on to another quantifier, which is impossible with conventional oracle calls.

But why does the procedure we've given correctly decide the Π_3^{P}-sentence S? Well, firstly, *if* a is a correct trusted advice string, then part (4) of the procedure just directly expresses S, using the assumed $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$ algorithm for coNP

to eliminate one of the three quantifiers in the usual manner of Karp-Lipton theorems.

That leaves the problem of verifying that a is a correct trusted advice string. Parts (2) and (3) of the procedure verify the latter, by quantifying over all possible SAT instances Ψ of the appropriate polynomial size, and checking that for each one, either Ψ has a satisfying assignment or else there's a witness w that causes C to accept Φ, but not both. (In other words, C decides coNP in $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$.)

Now, for parts (2) and (4), we additionally needed an *existential* quantifier over the untrusted quantum advice state σ, which is then verified using the trusted classical advice string a. The reason is that, in parts (2) and (4), the third and final quantifier needed, over the classical strings y, z, or w, happens to be existential—so that third quantifier simply *must* do "double duty" by also guessing the state σ. As mentioned before, passing a quantum state from an earlier quantifier to a later one is impossible with conventional oracle calls.

However, this need to quantify existentially over σ opens up a problem. Namely, what if the existential quantifiers, in parts (2) or (4), can be satisfied by *different* advice states σ—states that are all compatible with a, but that lead to different behaviors of C on some inputs? For example, perhaps there exists an a such that some σ's compatible with a give rise to a complete verification procedure for UNSAT, while other σ's compatible with a give rise to a sound verification procedure for UNSAT, but the same σ never gives rise to both. If so, then the σ that we find in part (4) need not give rise to a correct $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$ algorithm for coNP.

Fortunately, we can fix this problem using part (1). In part (1), we enforced that *every* state σ compatible with a must give rise to essentially the same behavior on every input. Thus, from that point forward, it doesn't even matter whether we find σ via a universal quantifier or an existential one: every σ that passes verification will give rise to the same behavior, and parts (2), (3), and (4) are all talking about the same $\mathsf{YQ} \cdot \mathsf{QCMA/poly}$ procedure that correctly decides coNP. □

Note that in the definition of offline QGES we merely assumed that there exists some efficient quantum circuit which the client could apply to map one input to another. However, we never explicitly stated that the client could come up with this circuit in polynomial time. If we also added this condition then we would find that $f \in \mathsf{BQP/qpoly}$ (which is of course contained in $\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$). In this case, using a result of Aaronson and Drucker which is a quantum version of the Karp-Lipton theorem [196], it follows that having such a QGES for NP-hard functions leads to $\Pi_2^{\mathsf{P}} \subseteq \mathsf{QMA}^{\mathsf{PromiseQMA}}$. Our proof of Theorem 19 uses similar techniques and in fact strengthens the result of Aaronson and Drucker from $\mathsf{NP} \subset \mathsf{BQP/qpoly}$ implies $\Pi_2^{\mathsf{P}} \subset \mathsf{QMA}^{\mathsf{PromiseQMA}}$, to $\mathsf{NP} \subset \mathsf{BQP/qpoly}$ implies $\Pi_2^{\mathsf{P}} \subseteq \mathsf{NP}^{\mathsf{PromiseQMA}}$.

6.6 Chapter summary and outlook

We have seen that the existence of a classical client blind quantum computing protocol is contingent on the inclusion $\mathsf{BQP} \subset \mathsf{NP/poly} \cap \mathsf{coNP/poly}$, in the case of decision problems, and the existence of circuits of size $2^{n-\Omega\left(\frac{n}{\log n}\right)}$, making polynomial-size queries to an $\mathsf{NP}^{\mathsf{NP}}$ oracle, for computing the permanent of an $n \times n$ matrix. Both of these seem unlikely. Provided that such a protocol cannot exist, what does this mean for the prospect of verifying quantum computations with a classical client? As we have already mentioned, such a protocol has been proposed, however it requires computational assumptions[18] stemming from the use of cryptographic primitives [37]. In fact, other than the Morimae and Fitzsimons post hoc protocol and a number of other protocols that are also based on the post hoc approach [28, 35, 36], all verification protocols employ cryptographic methods to hide some information from the server [15]. Our result suggests that if we demand information-theoretic security and wish to verify quantum computations with a classical client, we must reveal more information to the server than just the size of the computation. This leads to two options:

1. There does not exist a verification protocol for quantum computation, having a classical client, a single server that is restricted to performing BQP computations and unconditional soundness. One might ask why we would require unconditional soundness if the server is restricted to BQP. The reason is that for protocols that base their security on the hardness of some computational task, in practice, the chosen instances of that task can be solved within a time frame of at most several decades. While this can be good enough for most applications, it might not be adequate for users that wish to have retrospective security (i.e. their secrets are never revealed at any point in the future).

2. It is possible to verify quantum computations even when the server knows the computation. While this option might seem questionable given our preference for using blind protocols throughout the thesis, it is not without precedent. The proof that IP = PSPACE is an example of such an option [66]. The proof shows that there is an interactive protocol for any computation that can be performed in polynomial space. Of course, the protocol requires the server to perform PSPACE-complete computations and so it cannot be used in practice for the verification of BQP. However, the existence of this protocol serves as a proof of principle. It is possible for a computationally limited client to delegate and verify computations performed by a powerful server, while revealing both the input and the computation to that server.

It will be interesting to see which of these options turns out to be true.

Apart from examining the case of a GES for BQP computations, we also investigated the computational limitations of a QGES. We saw that the addition of quantum communication makes the QGES more powerful than its classical counterpart allowing for the delegation of BQP computations, as in UBQC. Interestingly, however, the QGES seems no more powerful than a GES at delegat-

[18] More specifically, the protocol assumes that a certain problem, know as Learning With Errors (LWE) [38], is not in BQP [37]. If this assumption is true, it allows for the existence of one-way functions that are secure against BQP adversaries. These functions are then used, in the protocol of [37], to verify BQP computations.

ing NP-hard functions. The latter result is essentially a quantized version of the Abadi et. al no-go theorem.

As open problems we mention the following:

- Is it possible to strengthen our result from Theorem 16 to provide an oracle separation between BQP and $\mathsf{NP/poly}$? Perhaps by basing the oracle on something other than Simon's problem, such as *recursive Fourier sampling* [12].

- We showed that functions which admit an offline QGES are contained in $\mathsf{QCMA/qpoly} \cap \mathsf{coQCMA/qpoly}$. What upper bound, if any, can be placed on functions which admit an *online* QGES?

- What if we consider a QGES in which the client's quantum message is logarithmic or poly-logarithmic in the size of the input (while the classical communication is still polynomial)? Can such a scheme allow for the evaluation of arbitrary BQP functions? Of course, this question only makes sense if we assume that the client is not able to perform BQP computations itself. Suppose that we restrict it so that it can process at most a logarithmic amount of quantum information, apart from its ability to perform polynomial-time classical computations. In that case, by adapting the proof of Theorem 18, we would find that the class of problems solvable by this QGES should be contained in $\mathsf{MA}^{\mathsf{BQL/qlog}}$, where BQL denotes the class of problems that can be solved on a quantum computer with logarithmic space and qlog denotes logarithmic-size quantum advice. The question then becomes: is $\mathsf{BQP} \subset \mathsf{MA}^{\mathsf{BQL/qlog}}$? It can be shown that $\mathsf{BQL/qlog} \subseteq \mathsf{P/poly}$[19], hence $\mathsf{MA}^{\mathsf{BQL/qlog}} \subseteq \mathsf{MA}^{\mathsf{P/poly}} \subseteq \mathsf{NP/poly}$. Since we conjectured that $\mathsf{BQP} \not\subset \mathsf{NP/poly}$, we find that such a QGES is unlikely, at least for the case of a logarithmic amount of quantum communication. For poly-logarithmic-size quantum communication, however, it is not clear if the corresponding oracle class is contained in $\mathsf{P/poly}$ and thus the problem remains open.

- Related to the previous question: what is the minimal amount of quantum communication that the client needs to send to the server in order to delegate BQP functions (provided $\mathsf{BPP} \neq \mathsf{BQP}$)?

[19] This follows from a simple argument: the BQL machine can be simulated in classical polynomial time and the logarithmic-size quantum advice can be encoded as a list of polynomially-many amplitudes. In other words, a $\mathsf{P/poly}$ machine can simulate the $\mathsf{BQL/qlog}$ oracle.

Chapter 7

Conclusions

> **Col. O'Neill**: I suppose now is the time for me to say something profound. *[pause]* Nothing comes to mind.
>
> — Stargate SG-1, Season 1, Episode 2

The realisation 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 likely be comprised 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 devices of IBM and Rigetti [16, 17], and more powerful machines are soon to follow [167, 198]. 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 [21], 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?

We have seen that there are a number of protocols addressing the question of verification, which we have classified as either prepare-and-send, receive-and-measure or entanglement-based. In this thesis, we explored the robustness of quantum verification and this has taken us through all of these approaches. We started by looking at the prepare-and-send protocol of Fitzsimons and Kashefi and showed that it is robust with respect to deviations in the prepared quantum states. We then showed that having this property allows it to be turned into an entanglement-based device-independent protocol, using the rigidity of non-local correlations as proven by Reichardt, Unger and Vazirani. Investigating robustness with respect to varying trust assumptions then allowed us to show that a one-sided device-independent protocol can be developed using steering correlations, instead of non-local correlations. This required us to prove a similar rigidity result for steering correlations. The added trust allowed for a simpler and more efficient derivation of this than in the non-local case. We also showed that some of the derived bounds are optimal. Of course, having robustness to noise or to different trust assumptions does not mean that these protocols would function correctly in the presence of a realistic noise channel, as we would encounter in a practical setting. To address this, we turned to the receive-and-measure protocol of Morimae and Fitzsimons and gave a simple construction for making it fault tolerant. Our simulations of this protocol seem encouraging for the prospect of implementing a version of it in practice, once scalable quantum computing becomes a reality. Finally, we noted that most verification protocols rely on blind quantum computation and addressed the question of whether blind quantum computation with a classical client is possible, provided that the protocol is secure in an information-theoretic sense. We gave complexity theoretic evidence that the answer is no. We then also showed that blind quantum computing protocols, like UBQC, are unlikely to allow the client to delegate NP-hard problems to the quantum server.

What can we conclude from all this? First of all, the results of Chapters 3, 4 and 5 suggest that we should be optimistic about the prospect of verification protocols being used alongside other quantum protocols, such as QKD. In other words, provided that all the hurdles in realising scalable quantum computers will be overcome, it seems that performing these verification protocols would not add a significant technological overhead. We have also examined some foundational aspects concerning quantum correlations and how they can be used for achieving verification. Specifically, using correlations to characterise quantum states through self-testing was an integral part of both the device-independent and the one-sided device-independent approaches. This leads us to conclude that improvements in the way one performs self-testing can directly impact the performance of a verification protocol. The recent approaches of Natarajan and Vidick [36] and Coladangelo et al. [106] demonstrate this fact, by showing that self-tests achieving constant robustness can lead to verification protocols with quasi-linear overhead.

Conversely, the results of Chapter 6 seem to suggest that we should be pessimistic regarding the possibility of classical client quantum verification, with unconditional soundness. It should be stressed, however, that those results specifically concern blind protocols. There is, so far, no evidence to suggest that a non-blind version of such a protocol cannot exist. Indeed, we speculate that an interactive-proof system for BQP computations, in which the prover is restricted to BQP, might be developed by taking inspiration from the proof that $\mathsf{IP} = \mathsf{PSPACE}$ [66]. An exploration of this approach is discussed in [199]. What we can conclude from Chapter 6 is that, very likely, protocols that aim to be both blind and have information-theoretic security, must either allow the client some quantum capabilities, or allow for the possibility of multiple, entangled servers.

What is the outlook for quantum verification? All of the verification approaches discussed in this thesis assumed a setting of unconditional soundness. However, as we have previously mentioned, there are also a number of protocols that achieve either blind quantum computation or verification by using computational assumptions about post-quantum security, specifically the assumption that the Learning With Errors problem is not contained in BQP [37, 169, 170]. Additionally, there are protocols for quantum fully homomorphic encryption that also make use of such assumptions [182, 183, 200]. It is therefore apparent that a new direction has emerged in the field of delegated quantum computations, one that leverages cryptographic assumptions in order to solve information processing tasks. It is also apparent, given its success, that this program will continue, improving on existing schemes and also developing protocols for other tasks, such as producing certifiable randomness [201].

Returning to protocols having unconditional soundness, future research into this field will involve addressing many of the open questions that were mentioned at the end of each chapter. Let us briefly restate some of the more important ones. First of all, as mentioned before, arguably the most significant open problem of the field is whether one can perform quantum verification with a purely classical client and unconditional soundness. Another open problem, is whether it is possible to have a blind and fault tolerant verification protocol in which the verifier possesses a noisy single-qubit device. As mentioned, in Chapter 5, the question is whether such a protocol can exist even when assuming that the noise in the verifier's device is correlated with the prover's system. The technical difficulties in achieving this are discussed in [159]. Finally, we would like to know what is the minimal amount of quantum communication that should be exchanged between the verifier and the prover in order to have blind verification and unconditional security. This relates to the quantum generalised encryption scheme of Chapter 6. Could it be that a QGES in which the verifier sends a quantum message of size poly-logarithmic in the size of the input, is sufficient for delegating arbitrary BQP computations? For the case of a logarithmic-size quantum message, our argument at the end of Chapter 6, suggests that it is not possible. However, the poly-logarithmic case is still open and such a protocol would be highly desirable, since minimising the amount of quantum communication between the verifier and the prover would lead to more practical protocols.

Verification is an important milestone on the road to scalable quantum computing technology. As we have seen, verification protocols are, or can be made, robust. Even so, among other issues, questions still remain regarding their optimality, their ability to tolerate certain types of noise or to be used in conjunction with cryptographic primitives. Addressing all these questions will be a key challenge for both theorists and experimentalists and their resolution will shape the quantum technology landscape.

Appendix A

SDP solver

Below is the Matlab code (using the cvx package [202]) for solving the SDP specified in Chapter 4 (Subsection 4.1.3), for the case of $N = 3$ Bell pairs.

```matlab
bellp = 1/sqrt(2) * [1; 0; 0; 1];
rho_bell = bellp * bellp';
numPairs = 3;
n = 4^numPairs;

rho_AB = 1;
projectors = cell(1, numPairs);
for i = 1 : numPairs
    rho_AB = kron(rho_AB, rho_bell);
    projectors{i} = kron(kron(eye(4^(i-1)), rho_bell), eye(4^(numPairs-i)));
end

numTrials = 10;
tracedistances = cell(1, numTrials);
for i = 1 : numTrials
    sqreps = 0.01 * i;
    cvx_begin sdp
        variable rho(n,n)
        variable p
        dual variables w y z t u
        p == (trace(rho * rho_AB))
        minimize p
        subject to
            w : rho == hermitian_semidefinite(n);
            y : trace(rho) == 1;
            z : trace(projectors{1} * rho) >= 1 - sqreps;
            t : trace(projectors{2} * rho) >= 1 - sqreps;
            u : trace(projectors{3} * rho) >= 1 - sqreps;
    cvx_end
    tracedistances{i} = norm(rho - rho_AB);
end
```

Listing A.1: SDP solver

Bibliography

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[2] Richard P Feynman, Robert B Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Vol. III: The New Millennium Edition: Quantum Mechanics.* The Feynman Lectures on Physics. Basic Books, 2011.

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

[4] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6-7):467–488, 1982.

[5] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.

[6] Ivan Kassal, Stephen P Jordan, Peter J Love, Masoud Mohseni, and Alán Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proceedings of the National Academy of Sciences*, pages pnas–0808245105, 2008.

[7] Hexadecimal sudoku. https://blog.digilentinc.com/number-systems/.

[8] Andrew Wiles. Modular elliptic curves and Fermat's last theorem. *Annals of mathematics*, 141(3):443–551, 1995.

[9] Wikipedia article on regular expressions. https://en.wikipedia.org/wiki/Regular_expression.

[10] Mikhail J Atallah. *Algorithms and theory of computation handbook.* CRC press, 1998.

[11] Complexity Zoo. https://complexityzoo.uwaterloo.ca/Complexity_Zoo.

[12] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, October 1997.

[13] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 537–, Washington, DC, USA, 2000. IEEE Computer Society.

[14] Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 141–150, New York, NY, USA, 2010. ACM.

[15] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *Theory of computing systems*, July 2018. Eprint: arXiv:1709.06984.

[16] IBM quantum experience. `http://research.ibm.com/ibm-q/`.

[17] Rigetti forest. `https://www.rigetti.com/forest`.

[18] Scott Aaronson. Could a quantum computer have a subjective experience? `https://www.scottaaronson.com/blog/?p=1951`.

[19] Scott Aaronson. The Aaronson $25.00 prize. `http://www.scottaaronson.com/blog/?p=284`.

[20] Umesh Vazirani. Workshop on the computational worldview and the sciences. `http://users.cms.caltech.edu/~schulman/Workshops/CS-Lens-2/report-comp-worldview.pdf`, 2007.

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

[22] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 2003.

[23] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 453–469, 2010.

[24] Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Phys. Rev. A*, 96:012303, Jul 2017.

[25] Anne Broadbent. How to verify a quantum computation, 2015. Eprint:arXiv:1509.09180.

[26] Tomoyuki Morimae. Verification for measurement-only blind quantum computing. *Physical Review A*, 89(6):060302, 2014.

[27] Masahito Hayashi and Tomoyuki Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *Physical review letters*, 115(22):220502, 2015.

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

[29] Tomoyuki Morimae, Yuki Takeuchi, and Masahito Hayashi. Verified measurement-based quantum computing with hypergraph states. *arXiv preprint arXiv:1701.05688*, 2017.

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

[31] Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456, 2013.

[32] Matthew McKague. Interactive proofs for BQP via self-tested graph states. *Theory of Computing*, 12(3):1–42, 2016.

[33] Alexandru Gheorghiu, Petros Wallden, and Elham Kashefi. Rigidity of quantum steering and one-sided device-independent verifiable quantum computation. *New Journal of Physics*, 19(2):023043, 2017.

[34] Michal Hajdušek, Carlos A Pérez-Delgado, and Joseph F Fitzsimons. Device-independent verifiable blind quantum computation. *arXiv preprint arXiv:1502.02563*, 2015.

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

[36] Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1003–1015. ACM, 2017.

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

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

[39] `https://www.scottaaronson.com/blog/?p=3697`.

[40] Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 465, pages 1413–1439. The Royal Society, 2009.

[41] Lars Lydersen, Carlos Wiechers, Christoffer Wittmann, Dominique Elser, Johannes Skaar, and Vadim Makarov. Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature photonics*, 4(10):686, 2010.

[42] Ilja Gerhardt, Qin Liu, Antía Lamas-Linares, Johannes Skaar, Christian Kurtsiefer, and Vadim Makarov. Full-field implementation of a perfect eavesdropper on a quantum cryptography system. *Nature communications*, 2:349, 2011.

[43] Vadim Makarov, Jean-Philippe Bourgoin, Poompong Chaiwongkhot, Mathieu Gagné, Thomas Jennewein, Sarah Kaiser, Raman Kashyap, Matthieu Legré, Carter Minshull, and Shihan Sajeed. Creation of backdoors in quantum communications via laser damage. *Physical Review A*, 94(3):030302, 2016.

[44] Alexandru Gheorghiu, Matty J Hoban, and Elham Kashefi. A simple protocol for fault tolerant verification of quantum computation. *arXiv preprint arXiv:1804.06105*, 2018.

[45] Scott Aaronson, Alexandru Cojocaru, Alexandru Gheorghiu, and Elham Kashefi. On the implausibility of classical client blind quantum computing. *arXiv preprint arXiv:1704.08482*, 2017.

[46] John Watrous. Guest column: An introduction to quantum information and quantum circuits 1. *SIGACT News*, 42(2):52–67, June 2011.

[47] John Watrous. Quantum computational complexity. In *Encyclopedia of complexity and systems science*, pages 7174–7201. Springer, 2009.

[48] Nicholas Harrigan and Robert W Spekkens. Einstein, incompleteness, and the epistemic view of quantum states. *Foundations of Physics*, 40(2):125–157, 2010.

[49] Daniel Gottesman. 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*, volume 68, pages 13–58, 2009.

[50] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[51] Hans J Briegel, David E Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nat Phys*, pages 19–26, Jan 2009.

[52] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science*, FOCS '09, pages 517 – 526. IEEE Computer Society, 2009.

[53] Dominic Mayers and Andrew Yao. Self testing quantum apparatus. *Quantum Info. Comput.*, 4(4):273–286, July 2004.

[54] Andrea Coladangelo and Jalex Stark. Separation of finite and infinite-dimensional quantum correlations, with infinite question or answer sets. *arXiv preprint arXiv:1708.06522*, 2017.

[55] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[56] Boris S Cirel'son. Quantum generalizations of Bell's inequality. *Letters in Mathematical Physics*, 4(2):93–100, 1980.

[57] Stephen J Summers and Reinhard Werner. Maximal violation of Bell's inequalities for algebras of observables in tangent spacetime regions. In *Annales de l'Institut Henri Poincare Physique Theorique*, volume 49, pages 215–243, 1988.

[58] Sandu Popescu and Daniel Rohrlich. Which states violate Bell's inequality maximally? *Physics Letters A*, 169(6):411–414, 1992.

[59] Samuel L Braunstein, Ady Mann, and Michael Revzen. Maximal violation of Bell inequalities for mixed states. *Physical Review Letters*, 68(22):3259, 1992.

[60] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[61] Russell Impagliazzo and Avi Wigderson. $\mathsf{P}= \mathsf{BPP}$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 220–229. ACM, 1997.

[62] Peter Bro Miltersen and N Variyam Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 71–80. IEEE, 1999.

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

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

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

[66] Adi Shamir. $\mathsf{IP} = \mathsf{PSPACE}$. *J. ACM*, 39(4):869–877, October 1992.

[67] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 113–131. ACM, 1988.

[68] Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on*, pages 236–249. IEEE, 2004.

[69] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[70] Scott Aaronson. The equivalence of sampling and searching. In *Proceedings of the 6th International Conference on Computer Science: Theory and Applications*, CSR'11, pages 1–14, Berlin, Heidelberg, 2011. Springer-Verlag.

[71] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 333–342, New York, NY, USA, 2011. ACM.

[72] Larry Stockmeyer. The complexity of approximate counting. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 118–126. ACM, 1983.

[73] Clemens Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215–217, 1983.

[74] Joseph F. Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):23, 2017.

[75] Andrew M. Childs. Secure assisted quantum computation. *Quantum Info. Comput.*, 5(6):456–466, September 2005.

[76] Pablo Arrighi and Louis Salvail. Blind quantum computation. *International Journal of Quantum Information*, 04(05):883–898, 2006.

[77] Vittorio Giovannetti, Lorenzo Maccone, Tomoyuki Morimae, and Terry G. Rudolph. Efficient universal blind quantum computation. *Phys. Rev. Lett.*, 111:230501, Dec 2013.

[78] Atul Mantri, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Optimal blind quantum computation. *Phys. Rev. Lett.*, 111:230502, Dec 2013.

[79] Ronald L Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. 1978. *Foundations of secure computation*, 4(11):169–180.

[80] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[81] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 97–106, Washington, DC, USA, 2011. IEEE Computer Society.

[82] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

[83] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 24–43, Berlin, Heidelberg, 2010. Springer-Verlag.

[84] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.

[85] Vincent Danos and Elham Kashefi. Determinism in the one-way model. *Physical Review A*, 74(5):052310, 2006.

[86] Vedran Dunjko and Elham Kashefi. Blind quantum computing with two almost identical states, 2016. Eprint:arXiv:1604.01586.

[87] Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 406–425. Springer, 2014.

[88] Elham Kashefi and Petros Wallden. Optimised resource construction for verifiable quantum computation. *Journal of Physics A: Mathematical and Theoretical*, 50(14):145306, 2017.

[89] Robert Raussendorf, Jim Harrington, and Kovid Goyal. A fault-tolerant one-way quantum computer. *Annals of physics*, 321(9):2242–2270, 2006.

[90] Robert Raussendorf, Jim Harrington, and Kovid Goyal. Topological fault-tolerance in cluster state quantum computation. *New Journal of Physics*, 9(6):199, 2007.

[91] Theodoros Kapourniotis. Efficient verification of universal and intermediate quantum computing. 2016. Eprint:https://www.era.lib.ed.ac.uk/handle/1842/25492.

[92] Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390, 1999.

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

[94] D Hangleiter, M Kliesch, M Schwarz, and J Eisert. Direct certification of a class of quantum simulations. *Quantum Science and Technology*, 2(1):015004, 2017.

[95] Tomoyuki Morimae, Daniel Nagaj, and Norbert Schuch. Quantum proofs can be verified using only single-qubit measurements. *Physical Review A*, 93(2):022326, 2016.

[96] Theodoros Kapourniotis, Vedran Dunjko, and Elham Kashefi. On optimising quantum communication in verifiable quantum computing. *arXiv preprint arXiv:1506.06943*, 2015.

[97] How a crypto 'backdoor' pitted the tech world against the NSA. `https://www.wired.com/2013/09/nsa-backdoor/`.

[98] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 136–145. IEEE, 2001.

[99] Dominique Unruh. Universally composable quantum multi-party computation. In Henri Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 486–505. Springer Berlin Heidelberg, 2010.

[100] Renato Renner and Robert König. Universally composable privacy amplification against quantum adversaries. In Joe Kilian, editor, *Theory of Cryptography*, volume 3378 of *Lecture Notes in Computer Science*, pages 407–425. Springer Berlin Heidelberg, 2005.

[101] Michael Ben-Or, Micha Horodecki, Debbie W. Leung, Dominic Mayers, and Jonathan Oppenheim. The universal composable security of quantum key distribution. In Joe Kilian, editor, *Theory of Cryptography*, volume 3378 of *Lecture Notes in Computer Science*, pages 386–406. Springer Berlin Heidelberg, 2005.

[102] Ueli Maurer and Renato Renner. Abstract cryptography. In *In Innovations in Computer Science*. Tsinghua University Press, 2011.

[103] Vedran Dunjko, Joseph F. Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *Advances in Cryptology ASIACRYPT 2014*, volume 8874 of *Lecture Notes in Computer Science*, pages 406–425. Springer Berlin Heidelberg, 2014.

[104] Hiroyuki Hayashi, Gen Kimura, and Yukihiro Ota. Kraus representation in the presence of initial correlations. *Phys. Rev. A*, 67:062109, Jun 2003.

[105] Vedran Dunjko, Elham Kashefi, and Anthony Leverrier. Universal blind quantum computing with weak coherent pulses, 2011. Eprint:arXiv:1108.5571.

[106] Andrea Coladangelo, Alex Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: new schemes for verifiable delegated quantum computation, with quasilinear resources. *arXiv preprint arXiv:1708.07359*, 2017.

[107] Matthew McKague and Michele Mosca. Generalized self-testing and the security of the 6-state protocol. In *Proceedings of the 5th Conference on Theory of Quantum Computation, Communication, and Cryptography*, TQC'10, pages 113–130, Berlin, Heidelberg, 2011. Springer-Verlag.

[108] Mehdi Mhalla and Simon Perdrix. Graph states, pivot minor, and universality of (X, Z)-measurements. *International Journal of Unconventional Computing*, 9, 2013.

[109] Simon Perdrix and Luc Sanselme. Determinism and computational power of real measurement-based quantum computation. In *International Symposium on Fundamentals of Computation Theory*, pages 395–408. Springer, 2017.

[110] Andreas Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999.

[111] Tomohiro Ogawa and Hiroshi Nagaoka. Making good codes for classical-quantum channel coding via quantum hypothesis testing. *IEEE Transactions on Information Theory*, 53(6):2261–2266, 2007.

[112] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[113] Erwin Schrödinger. Probability relations between separated systems. *Mathematical Proceedings of the Cambridge Philosophical Society*, 32:446–452, 10 1936.

[114] Umesh Vazirani and Thomas Vidick. Fully device-independent quantum key distribution. *Phys. Rev. Lett.*, 113:140501, Sep 2014.

[115] Antonio Acín, Nicolas Brunner, Nicolas Gisin, Serge Massar, Stefano Pironio, and Valerio Scarani. Device-independent security of quantum cryptography against collective attacks. *Phys. Rev. Lett.*, 98:230501, Jun 2007.

[116] S. Pironio, A. Acín, S. Massar, A. Boyer de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmschenk, D. Hayes, L. Luo, T. A. Manning, and C. Monroe. Random numbers certified by Bell's theorem. *Nature*, 464(7291):1021–1024, Apr 2010.

[117] Jan Bouda, Marcin Pawłowski, Matej Pivoluska, and Martin Plesch. Device-independent randomness extraction from an arbitrarily weak min-entropy source. *Phys. Rev. A*, 90:032313, Sep 2014.

[118] Stefano Pironio, Antonio Acn, Nicolas Brunner, Nicolas Gisin, Serge Massar, and Valerio Scarani. Device-independent quantum key distribution secure against collective attacks. *New Journal of Physics*, 11(4):045021, 2009.

[119] Eric Gama Cavalcanti, Steve J Jones, Howard M Wiseman, and Margaret D Reid. Experimental criteria for steering and the Einstein-Podolsky-Rosen paradox. *Phys. Rev. A*, 80:032112, Sep 2009.

[120] Paul Skrzypczyk, Miguel Navascués, and Daniel Cavalcanti. Quantifying Einstein-Podolsky-Rosen steering. *Phys. Rev. Lett.*, 112:180404, May 2014.

[121] Sania Jevtic, Matthew Pusey, David Jennings, and Terry Rudolph. Quantum steering ellipsoids. *Phys. Rev. Lett.*, 113:020402, Jul 2014.

[122] Howard M Wiseman, Steve James Jones, and Andrew C Doherty. Steering, entanglement, nonlocality, and the Einstein-Podolsky-Rosen paradox. *Phys. Rev. Lett.*, 98:140402, Apr 2007.

[123] Matthew F. Pusey. Negativity and steering: A stronger Peres conjecture. *Phys. Rev. A*, 88:032313, Sep 2013.

[124] Cyril Branciard, Eric G. Cavalcanti, Stephen P. Walborn, Valerio Scarani, and Howard M. Wiseman. One-sided device-independent quantum key distribution: Security, feasibility, and the connection with steering. *Phys. Rev. A*, 85:010301, Jan 2012.

[125] Elsa Passaro, Daniel Cavalcanti, Paul Skrzypczyk, and Antonio Acn. Optimal randomness certification in the quantum steering and prepare-and-measure scenarios. *New Journal of Physics*, 17(11):113010, 2015.

[126] Bernhard Wittmann, Sven Ramelow, Fabian Steinlechner, Nathan K Langford, Nicolas Brunner, Howard M Wiseman, Rupert Ursin, and Anton Zeilinger. Loophole-free Einstein-Podolsky-Rosen experiment via quantum steering. *New Journal of Physics*, 14(5):053030, 2012.

[127] Dylan J Saunders, Steve J Jones, Howard M Wiseman, and Geoff J Pryde. Experimental EPR-steering using Bell-local states. *Nature Physics*, 6(11):845–849, 2010.

[128] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Proceedings of the 30th Annual Conference on Advances in Cryptology*, CRYPTO'10, pages 465–482, Berlin, Heidelberg, 2010. Springer-Verlag.

[129] Leonardo Disilvestro and Damian Markham. Quantum protocols within Spekkens' toy model. *Physical Review A*, 95(5):052324, 2017.

[130] Eric G. Cavalcanti, Michael J. W. Hall, and Howard M. Wiseman. Entanglement verification and steering when Alice and Bob cannot be trusted. *Phys. Rev. A*, 87:032306, Mar 2013.

[131] Sacha Kocsis, Michael J. W. Hall, Adam J. Bennet, Dylan J. Saunders, and Geoff J. Pryde. Experimental measurement-device-independent verification of quantum steering. *Nat Commun*, 6, Jan 2015.

[132] Church of the larger Hilbert space. `https://quantiki.org/wiki/church-larger-hilbert-space#Purification_of_quantum_states`.

[133] Ivan Šupić and Matty J Hoban. Self-testing through EPR-steering. *New Journal of Physics*, 18(7):075006, 2016.

[134] Matthew McKague, Tzyh Haur Yang, and Valerio Scarani. Robust self-testing of the singlet. *Journal of Physics A: Mathematical and Theoretical*, 45(45):455304, 2012.

[135] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J. (2)*, 19(3):357–367, 1967.

[136] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[137] Devin H. Smith et al. Conclusive quantum steering with superconducting transition-edge sensors. *Nature Communications*, 3, Jan 2012. doi:10.1038/ncomms1628.

[138] E. Cavalcanti, S. Jones, H. Wiseman, and M. Reid. Experimental criteria for steering and the Einstein-Podolsky-Rosen paradox. *Phys. Rev. A*, 80:032112, Sep 2009.

[139] Jean-Daniel Bancal, Miguel Navascués, Valerio Scarani, Tamás Vértesi, and Tzyh Haur Yang. Physical characterization of quantum devices from nonlocal correlations. *Phys. Rev. A*, 91:022115, Feb 2015.

[140] Tzyh Haur Yang and Miguel Navascués. Robust self-testing of unknown quantum systems into any entangled two-qubit states. *Phys. Rev. A*, 87:050102, May 2013.

[141] Cédric Bamps and Stefano Pironio. Sum-of-squares decompositions for a family of Clauser-Horne-Shimony-Holt-like inequalities and their application to self-testing. *Phys. Rev. A*, 91:052111, May 2015.

[142] Rui Chao, Ben W. Reichardt, Chris Sutherland, and Thomas Vidick. Overlapping Qubits. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48:1–48:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[143] Rui Chao, Ben W Reichardt, Chris Sutherland, and Thomas Vidick. Test for a large amount of entanglement, using few measurements. *arXiv preprint arXiv:1610.00771*, 2016.

[144] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[145] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Mixed-state entanglement and distillation: Is there a "bound" entanglement in nature? *Phys. Rev. Lett.*, 80:5239–5242, Jun 1998.

[146] Karol Horodecki, Michał Horodecki, Paweł Horodecki, and Jonathan Oppenheim. General paradigm for distilling classical key from quantum states. *IEEE Trans. Inf. Theor.*, 55(4):1898–1929, April 2009.

[147] A El Allati, Morad El Baz, and Yassine Hassouni. Quantum key distribution via tripartite coherent states. *Quantum Information Processing*, 10(5):589–602, 2011.

[148] Christian Gabriel, Christoffer Wittmann, Denis Sych, Ruifang Dong, Wolfgang Mauerer, Ulrik L. Andersen, Christoph Marquardt, and Gerd Leuchs. A generator for unique quantum random numbers based on vacuum states. *Nat Photon*, 4(10):711–715, Oct 2010.

[149] M. Jofre, M. Curty, F. Steinlechner, G. Anzolin, J. P. Torres, M. W. Mitchell, and V. Pruneri. True random numbers from amplified quantum vacuum. *Opt. Express*, 19(21):20665–20672, Oct 2011.

[150] Felipe Fernandes Fanchini, Diogo de Oliveira Soares Pinto, and Gerardo Adesso. *Lectures on General Quantum Correlations and Their Applications.* Springer, 2017.

[151] Matthew Fairbairn Pusey. Is quantum steering spooky?, September 2013. https://spiral.imperial.ac.uk/bitstream/10044/1/12926/1/Pusey-MF-2013-PhD-Thesis.pdf.

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

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

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

[155] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Information & Computation*, 14(15-16):1338–1372, 2014.

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

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

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

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

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

[161] Tomoyuki Morimae, Keisuke Fujii, and Harumichi Nishimura. Quantum Merlin-Arthur with noisy channel. *arXiv preprint arXiv:1608.04829*, 2016.

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

[163] Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.

[164] Github repository. `https://github.com/agheorghiu/FTCode`.

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

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

[167] Google bristlecone. `https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html`.

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

[169] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *arXiv preprint arXiv:1708.02130*, 2017.

[170] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Delegated pseudo-secret random qubit generator. *arXiv preprint arXiv:1802.08759*, 2018.

[171] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 195–203, New York, NY, USA, 1987. ACM.

[172] Chee K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26(3):287 – 300, 1983.

[173] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[174] Thomas Jansen. On the black-box complexity of example functions: The real jump function. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, FOGA '15, pages 16–24, New York, NY, USA, 2015. ACM.

[175] Andreas Björklund. Below All Subsets for Some Permutational Counting Problems . In Rasmus Pagh, editor, *15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016)*, volume 53 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:11, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[176] Herbert John Ryser. *Combinatorial mathematics*, volume 14. JSTOR, 1963.

[177] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.

[178] Scott Aaronson. $\mathsf{P} \overset{?}{=} \mathsf{NP}$. 2017. https://www.scottaaronson.com/papers/pnp.pdf.

[179] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography.* CRC press, 2014.

[180] Pablo Arrighi and Louis Salvail. Blind quantum computation. *International Journal of Quantum Information*, 04(05):883–898, 2006.

[181] Anne Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015.

[182] Anne Broadbent and Stacey Jeffery. 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*, pages 609–629, 2015.

[183] Yfke Dulek, Christian Schaffner, and Florian Speelman. *Quantum Homomorphic Encryption for Polynomial-Sized Circuits*, pages 3–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[184] Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. *Computational Security of Quantum Encryption*, pages 47–71. Springer International Publishing, Cham, 2016.

[185] Li Yu, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Limitations on information-theoretically-secure quantum homomorphic encryption. *Phys. Rev. A*, 90:050303, Nov 2014.

[186] Michael Newman and Yaoyun Shi. Limitations on transversal computation through quantum homomorphic encryption. *arXiv preprint arXiv:1704.07798*, 2017.

[187] Tomoyuki Morimae and Takeshi Koshiba. Impossibility of perfectly-secure delegated quantum computing for classical client, 2014. Eprint:arXiv:1407.1636.

[188] Vedran Dunjko and Elham Kashefi. Blind quantum computing with two almost identical states, 2016. Eprint:arXiv:1604.01586.

[189] Atul Mantri, Tommaso F. Demarie, Nicolas C. Menicucci, and Joseph F. Fitzsimons. Flow ambiguity: A path towards classically driven blind quantum computation, 2016. Eprint:arXiv:1608.04633.

[190] Scott Aaronson. $\mathsf{QMA/qpoly} \subseteq \mathsf{PSPACE/poly}$: De-Merlinizing quantum protocols. In *in Proceedings of 21st IEEE Conference on Computational Complexity*, 2006.

[191] G. Brassard. A note on the complexity of cryptography (corresp.). *IEEE Transactions on Information Theory*, 25(2):232–233, Mar 1979.

[192] Leonard Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, SFCS '78, pages 75–83, Washington, DC, USA, 1978. IEEE Computer Society.

[193] Scott Aaronson. Quantum lower bound for recursive Fourier sampling. *Quantum Info. Comput.*, 3(2):165–174, March 2003.

[194] Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. In *Computational Complexity, 2007. CCC'07. Twenty-Second Annual IEEE Conference on*, pages 115–128. IEEE, 2007.

[195] Richard M. Karp and Richard J. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28:191–201, 1982.

[196] Scott Aaronson and Andrew Drucker. A full characterization of quantum advice. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 131–140, New York, NY, USA, 2010. ACM.

[197] Scott Aaronson and Andrew Drucker. A full characterization of quantum advice, 2010. Eprint:arXiv:1004.0377.

[198] IBM raises the bar with a 50-qubit quantum computer. `https://www.technologyreview.com/s/609451/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/`.

[199] Dorit Aharonov and Ayal Green. A quantum inspired proof of $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{IP}$. *arXiv preprint arXiv:1710.09078*, 2017.

[200] Gorjan Alagic, Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum fully homomorphic encryption with verification. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 438–467. Springer, 2017.

[201] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. Certifiable randomness from a single quantum device. *arXiv preprint arXiv:1804.00640*, 2018.

[202] Matlab package for convex programming. `http://cvxr.com/cvx/`.