DISSERTATION | DOCTORAL THESIS

Titel | Title

Quantifying the nonlocality of two-qubit states and quantum computation with indefinite causal structures

verfasst von | submitted by

Martin Johannes Renner BSc ETH MSc ETH

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Doktor der Naturwissenschaften (Dr.rer.nat.)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 796 605 411
Dissertationsgebiet lt. Studienblatt | Field of study as it appears on the student record sheet:	Physik
Betreut von | Supervisor:	Univ.-Prof. Mag. Dr. Caslav Brukner

Abstract

Bell's theorem shows that quantum correlations cannot be explained by local hidden variables. However, the effect of quantum nonlocality can be explained when the shared local hidden variables are augmented with some classical communication. The amount of classical communication required to simulate the correlations arising from local measurements on a given quantum state serves as a quantifier for the nonlocality of that state. In the first part of this thesis, we prove that all quantum correlations arising from arbitrary local measurements on an arbitrary state of two qubits can be simulated at the classical cost of two bits of communication. Our proof is based on a classical simulation of general qubit strategies in prepare-and-measure scenarios. In addition, we show that if a subclass of all measurements, namely projective measurements, are considered, already a single trit, a classical message with three symbols, is sufficient to simulate all local measurements on two-qubit states. A closely related question to the simulation cost is the question of which quantum states can be simulated without any communication and therefore admit a local hidden variable model. We introduce a novel local hidden variable model for a specific family of two-qubit states known as Werner states.

In the second part of this thesis, we study quantum computations with indefinite causal structures. Quantum computations are expected to solve several problems significantly faster than any classical machine. Conventional quantum algorithms can be represented within the quantum circuit model, where quantum gates are applied one after the other on a given quantum system. In recent years, it was discovered that the ordering of the quantum gates can be controlled with an additional quantum system. In the second part of this thesis, we study, how these quantum-controlled gate orderings can enhance quantum computing beyond what is achievable with fixed-order quantum circuits. We prove that for a recently established class of problems, the so-called Fourier promise problem, the advantage of these quantum-controlled gate orderings is smaller than previously expected. In addition, we study another class of problems, called Hadamard promise problems, and show that they provide an advantage of a quantum superposition of gate orderings. Problems of the latter class only require qubit gates and are therefore suitable to demonstrate such an advantage experimentally.

Kurzfassung

Bell's Theorem zeigt, dass Quantenkorrelationen nicht durch lokale verborgene Variablen erklärt werden können. Dieses Phänomen der Quanten-Nichtlokalität kann jedoch erklärt werden, wenn die lokalen Variablen durch klassische Kommunikation ergänzt werden. Die Menge der klassischen Kommunikation, die erforderlich ist, um die Korrelationen zu simulieren, die durch lokale Messungen an einem gegebenen Quantenzustand entstehen, dient als Maß für die Nichtlokalität dieses Zustands. Im ersten Teil dieser Arbeit beweisen wir, dass alle Quantenkorrelationen, die aus beliebigen Messungen an einem beliebigen Zustand von zwei Qubits resultieren, mit zwei klassischen Bits simuliert werden können. Unser Beweis basiert auf einer klassischen Simulation allgemeiner Qubit-Zustände in einem Szenario, in dem eine Partei einen beliebigen Zustand an eine andere Partei schickt, die den erhaltenen Zustand beliebig messen kann. Zusätzlich zeigen wir, dass bereits ein einzelnes Trit, eine klassische Nachricht mit drei Symbolen, ausreicht, um die Korrelationen von Zwei-Qubit-Zuständen zu simulieren, wenn nur projektive Messungen betrachtet werden. Eine eng verwandte Frage ist, welche Quantenzustände ohne jegliche Kommunikation simuliert werden können und daher ein Modell mit lokalen verborgenen Variablen zulassen. Wir stellen ein neuartiges Modell mit lokalen verborgenen Variablen für eine spezifische Familie von Zwei-Qubit-Zuständen vor, die als Werner-Zustände bekannt sind.

Im zweiten Teil dieser Arbeit untersuchen wir Quantenberechnungen mit unbestimmten kausalen Strukturen. Es wird erwartet, dass Quantenberechnungen mehrere Probleme signifikant schneller lösen können als herrkömmliche klassische Computer. Konventionelle Quantenalgorithmen können innerhalb des Quanten-Schaltkreismodells dargestellt werden, bei dem Quantengatter nacheinander auf ein gegebenes Quantensystem angewendet werden. In den letzten Jahren wurde entdeckt, dass die Reihenfolge der Quantengatter mit einem zusätzlichen Quantensystem gesteuert werden kann. Im zweiten Teil dieser Arbeit untersuchen wir, wie diese quantengesteuerten Gatterreihenfolgen die Quantenberechnungen über das hinaus verbessern können, was mit herrkömmlichen Quanten-Schaltkreisen erreichbar ist. Wir beweisen, dass für eine kürzlich etablierte Klasse von Problemen, das sogenannte Fourier-Versprechen-Problem, der Vorteil dieser quantengesteuerten Gatterreihenfolgen kleiner ist als bisher erwartet. Darüber hinaus untersuchen wir eine andere Klasse von Problemen, die als Hadamard-Versprechen-Probleme bekannt sind, und zeigen, dass sie einen Vorteil einer Quanten-Superposition von Gatterreihenfolgen bieten. Diese Probleme erfordern nur Qubit-Gatter und sind daher geeignet, einen solchen Vorteil experimentell zu demonstrieren.

Acknowledgements

I would like to begin by acknowledging Časlav Brukner for supervising part of this thesis, giving me all the freedom and opportunities, and allowing me to work on the questions I was falling in love with.

Most importantly, I would like to express my deepest gratitude to Marco Túlio Quintino, for his unwavering support throughout my PhD. His continuous guidance, insightful feedback, and encouragement have been instrumental in the development of this thesis. Without his expertise and patience, I would not have been able to navigate the complexities of this research. His enthusiasm for the subject matter has been contagious and has inspired me to pursue my research with vigor and dedication. Thank you, for believing in my potential and being an exceptional mentor! Your contributions to this thesis are immeasurable, and I am deeply grateful for your supervision and patience, always believing that something will come out of these projects.

Among the many nice people I met in Vienna, I would especially like to thank Armin Tavakoli for many interesting discussions and collaborations and Borivoje Dakić for his mentoring and continuous support, particularly in the last stage of my PhD. I would also like to extend my heartfelt thanks to Antonio Acín, Nicolas Brunner, Nicolas Gisin, Otfried Gühne, Robert König, Barbara Kraus, and Armin Tavakoli for inviting me to visit their institutions during my PhD. These visits provided me with invaluable opportunities to broaden my research perspectives, gain new insights, and collaborate with other experts in the field. Thank you all for your hospitality and for making my visits productive and memorable.

I acknowledge financial support from the Austrian Science Fund (FWF) through "BeyondC: Quantum Information Systems Beyond Classical Capabilities" (F7103-N38 and Grant DOI: 10.55776/F71).

Preamble

Quantum mechanics profoundly differs from classical physics. While classical physics relies on deterministic principles where objects have precise properties and predictable behaviors, quantum mechanics introduces concepts that defy this classical intuition. One of the key principles in quantum mechanics is Heisenberg's uncertainty principle [1]. It implies that it is impossible to simultaneously know both, the exact position and the exact momentum of a particle. Another fundamental difference is the concept of entanglement [2]. Entanglement implies that particles can become interconnected in such a way that the state of one particle is always correlated with the state of another particle. This particular aspect of quantum mechanics is most famously illustrated by the Einstein-Podolsky-Rosen (EPR) argument [3]. In their thought experiment, two particles are entangled such that both, the position and the momentum are always correlated. Hence, when we measure the position (momentum) of one particle, we can infer the position (momentum) of the other particle. Therefore, we could measure the momentum of one and the position of the other particle. Since these two properties are correlated, it seems that we can nevertheless know both properties simultaneously. Since this is an apparent contradiction to Heisenberg's uncertainty principle, Einstein and his colleagues suggested that quantum mechanics is not complete and that both properties are simultaneously well-defined. Alternatively, one has to assume that measuring one particle instantaneously changes the properties of the other entangled particle. Such a nonlocal behavior, famously known as "spooky action at a distance," is starkly different from the classical view, where objects are only influenced by their immediate surroundings. Hence, it was more natural for them to assume that both properties, position and momentum, are well-defined and that a complete description of quantum mechanics should consider these local hidden properties.

In 1964 a breakthrough was made by John Stuart Bell, who showed that such a local hidden variable (LHV) description of entangled quantum systems is impossible [4]. He proved that when measurements are performed on two entangled particles they can produce correlations that are stronger than any predictions that could evolve from such a local hidden variable description of the quantum states. Hence, no theory of local hidden variables can explain the correlations arising from quantum measurements. It indeed seems that measuring a property of one particle instantaneously changes the measurement outcomes of the other particle. This effect is now known as quantum nonlocality. What was first considered purely foundational considerations soon found applications in important information processing tasks. In fact, we can use these nonlocal correlations to ensure the security of quantum cryptography [5], to certify randomness [6], and even to prove that quantum computation is superior to classical computation [7]. Quantum nonlocality was soon established as a powerful resource for many information-processing tasks and it became important to quantify the strength of these nonlocal correlations.

A natural measure to quantify Bell nonlocality is to study the amount of classical communication required to reproduce the nonlocal quantum correlations. Bell's theorem only establishes that quantum correlations cannot be reproduced by local hidden variables alone. One can then ask whether these correlations can be reproduced when LHVs are supplemented with a certain amount of classical information transmitted from one side to the other. If so, how much communication is necessary for a given quantum state? If a state requires a large amount of information transfer between the two particles, it is arguably more nonlocal than a state requiring only a small amount of communication. In this sense, the communication cost is a natural measure to quantify the nonlocality of a given quantum state. This is the topic of the first part of this thesis.

This question was first posed by Tim Maudlin in 1992 [8]. He even speculated that, in the simplest case of two maximally entangled qubits, an infinite amount of communication would be required. This conjecture stemmed from the fact that a continuous (and therefore infinite) set of measurements can be performed on a single qubit. Since each measurement on one side leads to a different state on the other side, it seemed plausible that the information about the entire measurement basis would need to be transferred from one particle to the other. However, this intuition was shown to be incorrect. Shortly thereafter, Brassard, Cleve, and Tapp [9] developed the first protocol capable of simulating the most fundamental quantum state with a finite amount of communication. They demonstrated that eight bits are sufficient to simulate the statistics of arbitrary local projective measurements on two maximally entangled qubits. They also proved that high-dimensional entangled states require an amount of communication that scales exponentially with the system size. Hence, according to this measure of nonlocality, larger systems are indeed more nonlocal than smaller ones. Later, these models were further improved, and a significant breakthrough was made by Toner and Bacon in 2003 [10]. They showed that only a single bit of information is necessary to simulate arbitrary projective measurements on a pair of maximally entangled qubits. However, many questions remain unresolved. For instance, all of these models consider only a restricted class of measurements known as projective measurements. For the most general class of measurements, namely positive-operator valued measures (POVM), it was not known if the communication cost is always finite [11]. It was also unclear whether non-maximally entangled states can be simulated with a single bit of communication.

In Chapter 1, we prove that also for the most general class of measurements, the communication cost is finite and two classical bits can reproduce all correlations resulting from arbitrary local measurements on two entangled qubits. Our result is even more general. Namely, we consider general qubit strategies in so-called prepare-and-measure scenarios. In such a scenario, there is one party, who can send an arbitrary qubit state to another party. The party, who receives the qubit can then measure the received state with an arbitrary quantum measurement. This scenario is interesting since it is among the simplest ones that demonstrate the power of quantum compared to classical resources. Namely, it is known that sending a quantum state can boost communication beyond what is achievable with classical resources [12, 13]. A famous and simple example is Random Access Coding [14]. This task demonstrates that sending a qubit is advantageous compared to sending a classical bit. In fact, there is a strategy of sending one qubit that performs provably better than anything that can be achieved by sending a single bit of communication. At the same time, one might be interested in the question of how large such an advantage can be. Namely, is it also possible to find a task where sending a quantum bit performs better than sending two classical bits or even more? Our result proves that this is not the case and whatever correlations you can achieve with a qubit, you can also achieve by sending a classical message of two bits (given that additional shared randomness is available). At the same time, we can also prove that a classical message shorter than two classical bits is not sufficient to simulate a qubit in this prepare-and-measure scenario. Hence, our work implies tight bounds for this setting and sets an important upper bound on the quantum-to-classical advantage in these prepare-and-measure scenarios. Furthermore, this scenario is directly linked to the one of entangled quantum states. When two parties can simulate all qubit strategies in a prepare-and-measure scenario, they can also simulate all entangled two-qubit states with the same amount of classical communication [15].

In Chapter 2, we continue on this question and ask how much communication is required to reproduce the quantum correlations resulting from local projective measurements on a given two-qubit state. Previously, the result of Toner and Bacon [10] proved that one bit is sufficient for a maximally entangled state. They also showed that two bits are sufficient for a general two-qubit state. This raised the question of why non-maximally entangled states are seemingly more costly to simulate than the maximally entangled one, or if better models exist. In Chapter 2, we introduce a new method to simulate entangled quantum states. This allows us to reproduce known results like the one from Toner and Bacon and to find new simulation protocols for entangled two-qubit states. In fact, we can show that if the state is weakly entangled, already a single bit of classical information is sufficient. On average, the communication cost becomes even less than one bit. Therefore, we could prove that weakly entangled states require strictly less information to be simulated than the maximally entangled one. In addition, we prove that a classical trit is always sufficient to simulate all local projective measurements on any entangled qubit pair.

In Chapter 3, we turn our attention to a closely related question. It is known that entanglement is a necessary ingredient to demonstrate the effect of quantum nonlocality. It is also known that every pure entangled state can demonstrate quantum nonlocality [16]. However, for practical applications and experiments, pure quantum states are an idealization. The unavoidable noise in experiments raises the important question of how robust the nonlocal quantum correlations are to noise, and if a given noisy (or mixed) quantum state can or cannot be used to demonstrate the effect of nonlocality. The pioneering work by Werner showed that some noisy entangled states are not able to violate any Bell inequality [17]. In fact, he provides an explicit model that can reproduce all quantum correlations using just local hidden variables. This model, however, only applies to a subclass of all measurements, namely projective measurements. Therefore, it is crucial to determine if the same states are also local when considering the most general class of measurements (POVMs). In Chapter 3, we extend Werner's model for the two-qubit case to these generalized measurements. Our model is not only a local hidden variable model but even a so-called local hidden state model and determines exactly which two-qubit Werner states can demonstrate quantum steering [3, 18, 19]. In addition, this question is also directly linked to another central concept of quantum mechanics, namely that of measurement incompatibility. Due to Heisenberg's uncertainty principle, we cannot know the spin of two orthogonal directions simultaneously [1]. However, it was realized that if the measurements suffer from noise, they may become jointly measurable [20]. In Chapter 3, we ask the question of how much white noise is necessary such that all qubit measurements become jointly measurable and determine the precise threshold. The same methods are then used to derive a local hidden state model for two-qubit Werner states. To summarize, in the first part of this thesis, we introduce novel models to simulate the nonlocal behavior for all two-qubit states, as well as the local behavior for some noisy entangled states.

Although quantum mechanics is an established field and the experimental predictions are remarkably accurate, many fundamental aspects are still not entirely understood. It is likely that in the future it will present us with conceptually even more challenging puzzles. For instance, when we combine the realm of quantum mechanics and general relativity, there is still no commonly accepted theory that unifies these two fundamental theories. However, there are good chances that whatever such a theory might look like, it will challenge our classical notions of reality even further. If quantum mechanics is a universally valid theory, nothing forbids us to put large gravitating objects into a superposition. However, when large masses are in superposition, this has consequences for the underlying spacetime. Since spacetime determines how different events are connected, it is then likely to expect that events do not have to occur in a well-defined causal order. In recent works, certain frameworks were introduced that allow us to study these indefinite causal orders [21, 22, 23]. Albeit providing us with challenging puzzles, these new structures might also offer interesting avenues to enhance the capabilities of quantum resources even further.

In the second part of this thesis, we connect these indefinite causal structures with another important branch of quantum information sciences, namely quantum computing. Quantum computing is a rapidly evolving field that has the potential to make a huge societal impact. It is expected to solve certain problems much faster than any classical machine. The first and most famous example is Shor's algorithm which can factor large numbers in polynomial time [24]. For the same problem, all known classical algorithms require an exponential amount of time. Similarly, Grover's algorithm offers a quadratic speedup when searching through a large dataset [25]. Quantum algorithms are naturally represented by a quantum circuit. A quantum circuit consists of wires that represent the quantum systems. On these systems, quantum gates are applied. It is a common feature of conventional quantum algorithms that these gates are applied one after the other in a well-defined order. Using the framework of indefinite causal structures, it was discovered that such a constraint can be relaxed. In fact, it is possible to control the gate ordering with an additional quantum state. For the simplest case of two gates, gate A is applied before gate B, if the control system is in the state zero, and gate B is applied before gate A if the control system is in the state one. If the control system is now initialized in a superposition of zero and one, the gates A and B are applied in a superposition of these two gate orderings.

This new structure is intrinsically different from conventional quantum algorithms. This raises the question of whether they can be used to enhance quantum computation even further. In fact, we ask if there are tasks that can be solved by these indefinite causal structures more efficiently than with conventional quantum algorithms. In the simplest case, we can consider two gates. If it is promised that these two gates either commute or anticommute, it is known that a quantum-controlled ordering of gates can solve this problem by using each of these gates only once [26]. At the same time, any fixed-order quantum algorithm needs to call at least one of the two gates twice. This is a demonstration that indefinite causal structures can indeed be used to make quantum computation even more powerful. However, one can ask if the advantage is still preserved when we consider several gates. In 2014, Araújo, Costa, and Brukner [27] introduced a task that was expected to generalize such an advantage to an arbitrary number of gates. In their work, a set of gates are given and it is promised that they satisfy one property from a given list of properties. The task is to find out which property is the correct one. It was shown that a quantum superposition of gate orderings can solve these tasks efficiently and each gate needs to be called only once. At the same time, the best-known conventional algorithm called each gate n times, where n is the total number of different gates. In Chapter 4, we found conventional quantum algorithms that can solve the tasks introduced by Araújo, Costa, and Brukner [27] more efficiently. In fact, our algorithm can perfectly solve the task by calling each of the gates only $O(\log n)$ times. This raises the question if other problems can provide a larger advantage using these structures. In Chapter 5, we generalize a class of problems that was recently introduced by Taddei et al. [28]. Although they do not offer a larger advantage, they can be used to demonstrate such an effect experimentally. This stems from the fact that they only require the control of quantum systems with small dimensions, namely qubits.

List of Publications

This thesis is based on the following five articles:

- Martin J. Renner, Armin Tavakoli, Marco Túlio Quintino "Classical Cost of Transmitting a Qubit," Physical Review Letters 130 (12), 120801 (2023)
- Martin J. Renner, Marco Túlio Quintino "The minimal communication cost for simulating entangled qubits," Quantum 7, 1149 (2023)
- Martin J. Renner "Compatibility of Generalized Noisy Qubit Measurements," Physical Review Letters 132 (25), 250202 (2024)
- Martin J. Renner, Časlav Brukner "Reassessing the computational advantage of quantum-controlled ordering of gates," Physical Review Research 3 (4), 043012 (2021)
- Martin J. Renner, Časlav Brukner "Computational Advantage from a Quantum Superposition of Qubit Gate Orders," Physical Review Letters 128 (23), 230503 (2022)

Articles that were published during the time of my PhD but are not part of this thesis:

- Florian Pimpel, Martin J. Renner, Armin Tavakoli "Correspondence between entangled states and entangled bases under local transformations," Physical Review A 108 (2), 022220 (2023)

Contents

Part I.

Quantifying the nonlocality of two-qubit states

1. Classical Cost of Transmitting a Qubit

This chapter is based on the article:

Martin J. Renner, Armin Tavakoli, Marco Túlio Quintino "Classical Cost of Transmitting a Qubit," Physical Review Letters 130 (12), 120801 (2023)
(Received: 14 July 2022, Accepted: 17 January 2023, Published: 22 March 2023)

Contributions: Marco supervised the work and introduced the problem to me. I found the two-bit protocol (Section 1.3 and Section 1.A), the one-bit protocol for Bell scenarios (Section 1.5), and the proof for the no-one-bit part (Section 1.B). In these sections, I carried out the proofs and calculations and wrote most of the manuscript. Marco and Armin found that a classical trit cannot simulate these scenarios (Section 1.4 and Section 1.C). They carried out the proofs and calculations, and wrote most of the corresponding sections.

Abstract

We consider general prepare-and-measure scenarios in which Alice can transmit qubit states to Bob, who can perform general measurements in the form of positive operator-valued measures (POVMs). We show that the statistics obtained in any such quantum protocol can be simulated by the purely classical means of shared randomness and two bits of communication. Furthermore, we prove that two bits of communication is the minimal cost of a perfect classical simulation. In addition, we apply our methods to Bell scenarios, which extends the well-known Toner and Bacon protocol. In particular, two bits of communication are enough to simulate all quantum correlations associated to arbitrary local POVMs applied to any entangled two-qubit state.

1.1. Introduction

Quantum resources enable a sender and a receiver to break the limitations of classical communication. When entanglement is available, classical [29, 30, 31, 32] as well as quantum communication [33, 34] can be boosted beyond purely classical models. A seminal example is dense coding, in which two classical bits can be substituted for a single qubit and shared entanglement [35]. However, entanglement is not necessary for quantum advantages. Communicating an unassisted d-dimensional quantum system frequently outperforms the best conceivable protocols based on a classical d-dimensional system [36, 37, 38, 39, 40]; even yielding advantages growing exponentially in d [12, 13].

Already in the simplest meaningful scenario, namely that in which the communication of a bit is substituted for a qubit, sizable advantages are obtained in important tasks like Random Access Coding [14, 41, 42]. These qubit advantages propel a variety of quantum information applications [43, 44, 45, 46, 47].

It is natural to explore the fundamental limits of quantum over classical advantages. In order to do so, one has to investigate the amount of classical communication required to model the predictions of quantum theory. Previous works consider not only the scenario of sending quantum systems [15, 48, 49, 10, 11], but also simulating bipartite [8, 9, 50, 51, 48, 49, 15, 10, 11, 52, 53, 54], as well as multipartite entangled quantum systems [55, 56, 57, 58]. While such classical simulation of quantum theory is in general challenging, a breakthrough was made by Toner and Bacon [10]. Their protocol shows that any quantum prediction based on standard, projective, measurements on a qubit can be simulated by communicating only two classical bits. However, this does not account for the full power of quantum theory. More precisely, there exists qubit measurements that cannot be reduced to stochastic combinations of projective ones [59]. The most general measurements are known as positive operator-valued measures (POVMs). Physically, they correspond to the receiver interacting the message qubit with a locally prepared auxiliary qubit, and then performing a measurement on the joint system [60]. Such POVMs are even indispensable for important tasks like unambiguous state discrimination [61, 62] and hold a key role in many quantum information protocols (see e.g. [63, 64, 65, 66, 67, 68, 69, 70, 71]). Importantly, they also give rise to correlations that cannot be modelled in any qubit experiment based on projective measurements [72, 73, 74, 75, 76].

This naturally raises the question of identifying the classical cost of simulating the most general predictions of quantum theory, based on POVMs. In the minimal qubit communication scenario, one may suspect that this cheap price of only two bits is due to the restriction to the, fundamentally binary, projective measurements. In contrast, when measurements are general POVMs, it is even unclear whether the classical simulation cost is finite. Notably, previous work has shown that there exists a classical simulation that requires 5.7 bits of communication on average [15, 11]. However, that protocol has a certain probability to fail in each round, leading to an unbounded amount of communication in the worst case.

In this work, we explicitly construct a classical protocol that simulates all qubit-based correlations in the prepare-and-measure scenario by using only two bits of communication. Thus, we find that the cost of a classical simulation remains the same when considering the most general class of measurements, although POVMs enable more general quantum correlations than projective measurements. Moreover, we show that two bits is the minimal classical simulation cost, i.e. there exists no classical simulation that uses less communication than our protocol. This is shown through an explicit quantum protocol, based on qubit communication, that eludes simulation with a ternary classical message. Finally, we apply our methods to Bell nonlocality scenarios [77]. We present novel protocols that simulate the statistics of local measurements on entangled qubit pairs.

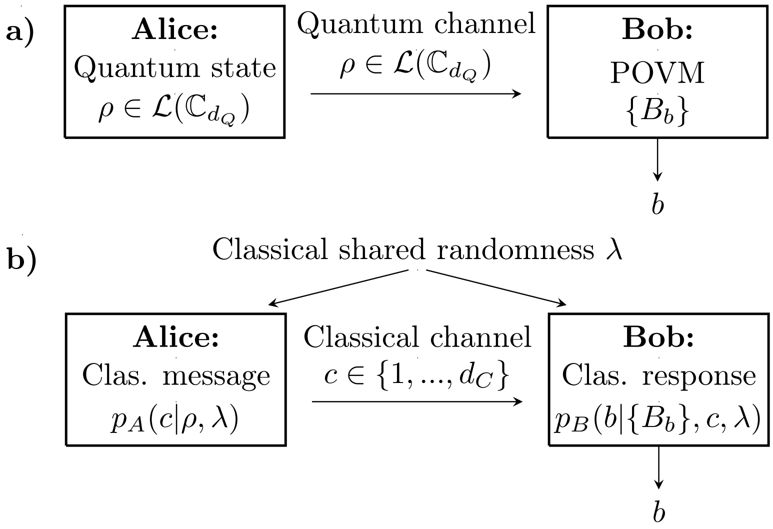


Figure 1.1.: **a)** Quantum PM scenario: Alice sends a d_Q-dimensional state to Bob who performs a POVM to obtain his outcome. **b)** Classical PM scenario for simulating the quantum PM scenario: The classical simulation is successful if, for every state and POVM, the probability that Bob outputs b is the same as in the quantum protocol.

1.2. The prepare-and-measure scenario

A quantum prepare-and-measure (PM) scenario (see Fig. 1.1 a)) consists of two steps. First, Alice prepares an arbitrary quantum state of dimension d_Q and sends it to Bob. The state is described by a positive semidefinite $d_Q \times d_Q$ complex matrix $\rho \in \mathcal{L}(\mathbb{C}_{d_Q})$, $\rho \geq 0$ with unit trace $\mathrm{tr}(\rho) = 1$. Second, Bob receives the state and performs an arbitrary quantum measurement on it, obtaining an outcome b. General quantum measurements are described by a POVM, which is a set of operators $\{B_b\}$ that are positive semidefinite, $B_b \geq 0$ and sum to the identity, $\sum_b B_b = \mathbb{1}$. In quantum theory, the probability of outcome b when performing the POVM $\{B_b\}$ on the state ρ is given by Born's rule,

$$p_Q(b|\rho, \{B_b\}) = \mathrm{tr}(\rho \, B_b) \,. \tag{1.1}$$

We are interested in constructing classical models for the PM scenario that simulate the predictions of quantum theory, i.e. classical models that reproduce the probability distribution (1.1). In a classical simulation (see Fig. 1.1 b)), Alice and Bob may share a random variable λ subject to some probability function $\pi(\lambda)$. This allows them to correlate their classical communication strategies. Alice uses λ and her knowledge of the quantum state ρ to choose a classical message c selected from a d_C-valued alphabet $\{1, \dots, d_C\}$. Since the selection can be probabilistic, her actions are described by the conditional probability distribution $p_A(c|\rho, \lambda)$. When Bob receives the message, he uses λ and his knowledge of the POVM $\{B_b\}$ to choose his outcome b. Again, this choice can be probabilistic and is therefore described by a conditional probability distribution $p_B(b|\{B_b\}, c, \lambda)$. All together, the correlations obtained from the classical model become

$$p_C(b|\rho, \{B_b\}) = \int_\lambda \mathrm{d}\lambda\ \pi(\lambda) \sum_{c=1}^{d_C} p_A(c|\rho, \lambda) p_B(b|\{B_b\}, c, \lambda)\,. \tag{1.2}$$

The simulation is successful if, for any choice of d_Q-dimensional states and POVMs, the quantum predictions p_Q can be reproduced with a classical model using messages that attain at most d_C different values. That is, if there exists a d_C and suitable encodings p_A and decodings p_B, such that

$$\forall \rho, \{B_b\}: \quad p_C(b|\rho, \{B_b\}) = p_Q(b|\rho, \{B_b\})\,. \tag{1.3}$$

If this holds, we say that the classical model simulates quantum theory. In particular, we say that the classical simulation is minimal if no classical simulation is possible using a smaller message alphabet size d_C. Furthermore, we remark that for some PM scenarios, shared randomness may be charged as a non-free resource, leading to different results and problems [48, 78, 42, 79, 80, 81, 70, 75, 82]. In fact, for the PM scenario we study here, it is known that an infinite amount of shared randomness is required in order to perform the task with finite classical communication [48].

Our focus is on the most fundamental scenario, namely that based on qubits ($d_Q = 2$). Notice that there exists a trivial classical simulation in which Alice sends the Bloch vector coordinates of her quantum state to Bob. After that, he can classically compute the Born rule and samples his outcome accordingly. However, sending the coordinates requires an infinite amount of communication (d_C unbounded). Whether a classical simulation is possible with a finite value of d_C is much less trivial. Notably, the simulation protocol of Toner and Bacon showed that if we additionally restrict the quantum measurements to be projective, i.e. $B_b^2 = B_b$, a classical simulation with $d_C = 4$ (two bits) is possible [10].

We also remark that here we consider a scenario where Bob does not know Alice's state and Alice does not know Bob's measurement beforehand. This scenario, where Alice and Bob can independently choose between different states and measurements, is even required to provide quantum over classical advantages in several tasks [14, 41, 42, 13, 12]. An interesting related scenario is the one where Bob's measurement is known by Alice, or, equivalently, Bob has only a single choice of measurement. In that case, Frenkel and Weiner [83] proved that, in the presence of shared randomness, a d-dimensional quantum system can always be perfectly simulated by a d-dimensional classical system. This powerful result inspired proposals such as the "No-Hypersignaling" principle [84], which is respected by quantum theory. In what follows, we find a minimal classical simulation for general qubit protocols.

1.3. Classical simulation protocol

Qubit states ρ can be represented as $\rho = (\mathbb{1} + \vec{x} \cdot \vec{\sigma})/2$, where $\vec{x} \in \mathbb{R}^3$ is a three-dimensional real vector such that $|\vec{x}| \leq 1$, and $\vec{\sigma} = (\sigma_X, \sigma_Y, \sigma_Z)$ are the standard Pauli matrices. We may, without loss of generality, restrict ourselves to quantum protocols based on pure states. This corresponds to unit vectors $|\vec{x}| = 1$. Since mixed states are convex combinations of pure states, every classical simulation protocol applicable to pure states can immediately be extended to apply also to mixed states. The classical randomness in the convex combination can simply be absorbed in the shared randomness of the simulation protocol. Similarly, because every qubit POVM can be written as a coarse graining of rank-1 projectors [85], we may restrict ourselves to POVMs proportional to rank-1 projectors. Thus, we write Bob's measurements as $B_b = 2p_b\,|\vec{y}_b\rangle\langle\vec{y}_b|$, where $p_b \geq 0$, $\sum_b p_b = 1$ and $|\vec{y}_b\rangle\langle\vec{y}_b| = \big(\mathbb{1} + \vec{y}_b \cdot \vec{\sigma}\big)/2$ for some normalized vector $\vec{y}_b \in \mathbb{R}^3$. In Bloch notation we have

$$\mathrm{tr}(\rho B_b) = p_b(1 + \vec{x} \cdot \vec{y}_b). \tag{1.4}$$

We now present a classical simulation protocol in which Alice and Bob can perfectly simulate all qubit correlations at the cost of two bits of communication. To this end, it is handy to first define the Heaviside function, defined by $H(z) = 1$ when $z \geq 0$ and $H(z) = 0$ when $z < 0$, as well as the related function $\Theta(z) := z \cdot H(z)$. Consider now the following protocol.

1. Alice and Bob share two normalized vectors $\vec{\lambda}_1, \vec{\lambda}_2 \in \mathbb{R}^3$, which are uniformly and independently distributed on the unit radius sphere S_2.

2. Instead of sending a pure qubit $\rho = \big(\mathbb{1} + \vec{x} \cdot \vec{\sigma}\big)/2$, Alice prepares two bits via the formula $c_1 = H(\vec{x} \cdot \vec{\lambda}_1)$ and $c_2 = H(\vec{x} \cdot \vec{\lambda}_2)$ and sends them to Bob.

3. Bob flips each vector $\vec{\lambda}_i$ when the corresponding bit c_i is zero. More formally, he sets $\vec{\lambda}_i^{\,\prime} := (-1)^{1+c_i}\vec{\lambda}_i$.

4. Instead of performing a POVM with elements $B_b = 2p_b\,|\vec{y}_b\rangle\langle\vec{y}_b|$, Bob picks one vector $\vec{y}_b$ from the set $\{\vec{y}_b\}$ according to the probabilities $\{p_b\}$. Then he sets $\vec{\lambda} := \vec{\lambda}_1^{\,\prime}$ if $|\vec{\lambda}_1^{\,\prime} \cdot \vec{y}_b| \geq |\vec{\lambda}_2^{\,\prime} \cdot \vec{y}_b|$ and $\vec{\lambda} := \vec{\lambda}_2^{\,\prime}$ otherwise. Finally, Bob outputs b with probability

$$p_B(b|\{B_b\}, \vec{\lambda}) = \frac{p_b\ \Theta(\vec{y}_b \cdot \vec{\lambda})}{\sum_j\ p_j\ \Theta(\vec{y}_j \cdot \vec{\lambda})}\,. \tag{1.5}$$

The proof that the protocol perfectly reproduces the qubit correlations (1.4) is given in Appendix 1.A. A sketch of the first three steps of the protocol is given in Fig. 1.2. After the third step, the two vectors $\vec{\lambda}_1^{\,\prime}$ and $\vec{\lambda}_2^{\,\prime}$ are uniformly and independently distributed in the positive hemisphere defined by $\vec{x}$, i.e. their probability densities are $\rho(\vec{\lambda}_i^{\,\prime}) = H(\vec{x} \cdot \vec{\lambda}_i^{\,\prime})/(2\pi)$. As we show, this distribution is enough for Bob to classically reproduce the statistics of every POVM applied to the qubit state associated to $\vec{x}$. Furthermore, in Appendix 1.A we also present a modified version of that protocol. There, Bob sends first one bit to Alice and then Alice sends one bit back to Bob.

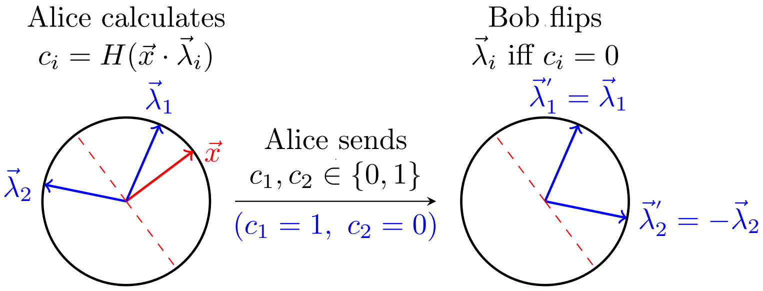

Alice calculates
$c_i = H(\vec{x} \cdot \vec{\lambda}_i)$
$\vec{\lambda}_1$
$\vec{x}$
$\vec{\lambda}_2$
Alice sends
$c_1, c_2 \in \{0,1\}$
$(c_1 = 1,\ c_2 = 0)$
Bob flips
$\vec{\lambda}_i$ iff $c_i = 0$
$\vec{\lambda}_1' = \vec{\lambda}_1$
$\vec{\lambda}_2' = -\vec{\lambda}_2$

Figure 1.2.: A two-dimensional illustration of the first three steps in the classical simulation protocol based on two bits.

1.4. Two bits are necessary for a classical simulation

We have shown that two classical bits are sufficient to simulate qubit correlations. We now prove that they are also necessary, i.e. that the above classical simulation protocol is minimal.

To this end, we show that there exists correlations in the qubit PM scenario that cannot be modelled in any classical protocol (1.2) that uses ternary messages ($d_C = 3$). For this purpose, we consider PM scenarios with a fixed number of inputs for Alice and Bob. Alice selects her input from a set $x \in \{1, \ldots, I_A\}$ and prepares the qubit ρ_x. Bob selects his input from a set $y \in \{1, \ldots, I_B\}$ and performs the two-outcome projective measurement $\{B_{b|y}\}$ with outcomes labelled by $b \in \{1, 2\}$. The qubit correlations are then given by $p_Q(b|x,y) = \operatorname{tr}\big(\rho_x B_{b|y}\big)$. Notice that although Bob could perform POVMs, we are restricting ourselves to projective measurements. These turn out to be sufficient for the proof.

It is key to recognise that the task of deciding whether a given $p(b|x,y)$ admits a classical simulation with a d_C-dimensional message alphabet can be solved by means of linear programming. From the duality theory of linear programming [86], we can obtain a classical dimension witness that certifies that the probabilities $p_Q(b|x,y)$ cannot be simulated by sending classical ternary messages. A classical dimension witness [13, 43] is a linear inequality which is respected by *all* classical models in the PM scenario for a given d_C. This can in general be written as

$$\sum_{b,x,y} \gamma(b|x,y) p_C(b|x,y) \leq C_d\,, \tag{1.6}$$

for some coefficients $\gamma(b|x,y) \in \mathbb{R}$. Here, C_d is the classical bound. A violation of this inequality certifies that no classical model using d_C symbols can simulate $p_Q(b|x,y)$. In Appendix 1.C, we detail these linear programming methods. Inspired by the efficient method to find local bounds of Bell inequalities presented in Ref. [87], we provide a new and efficient algorithm to obtain the classical d_C-dimensional bound $\leq C_d$ for any given set of coefficients $\{\gamma(b,x,y)\}$. Also, drawing inspiration from Ref. [88], we developed computational methods to convert the numerical solutions obtained from standard solvers to rigorous computer-assisted proofs which do not suffer from numerical precision issues due to floating point arithmetic.

In this way, we have obtained several examples of qubit states and measurements that generate quantum correlations $p_Q(b|x,y)$ that do not admit a classical model for $d_C = 3$. An elegant example is obtained from considering $I_A = 6$ states that form an octahedron on the Bloch sphere. They correspond to the eigenstates of the three Pauli operators $(\sigma_X, \sigma_Y, \sigma_Z)$. We let Bob perform $I_B = 24$ different projective measurements. The Bloch vectors of these measurements are oriented such that they point to the vertices of a snub cube [89], which is an Archimedean solid, inscribed in the Bloch sphere. This may be viewed as a PM variant of Platonic Bell inequality violations [90]. Specifically, the 24 measurement directions are obtained as follows. Let τ be the one real root of the polynomial $x^3 - x^2 - x - 1$, known as the Tribonacci constant. Take all even (odd) permutations of, $(\pm 1, \pm 1/\tau, \pm\tau)$ and for each permutation, take only the four sign combinations that have an even (odd) number of "+". This gives all vertices of the snub cube. Finally, do a global rotation by 60 degrees in the XY-plane, i.e. apply the unitary $U = |0\rangle\langle 0| + e^{\frac{i\pi}{3}} |1\rangle\langle 1|$ to all projectors. The linear programming methods reveal that the resulting p_Q has no classical model for $d_C = 3$.

In Appendix 1.C, we discuss a heuristic aproach to find states and measurements leading to probabilities which do not admit a classical simulation for $d_C = 3$. Fixing the above six preparations, the sparsest proof we have found uses eleven measurements that correspond to the solution of the Thomson problem [91]. All our computational code is openly available at the online repository [92].

Although no ternary message protocol is sufficient, it may still be that a classical simulation is possible by sending less than two bits on average. For example, Alice may restrict herself to send in some fraction of rounds only a trit, a bit or no communication at all. For the case of sometimes sending a bit or less, we show in Appendix 1.B that no classical simulation is possible. The reason is closely connected to the zero local weight of the singlet state, also known as the EPR2 decomposition [93, 94]. Our argument shows that, if one could simulate qubit correlations by sometimes sending only a bit or less, one could construct a protocol that simulates the singlet state without communication in these rounds. This would induce a local part for the singlet state, which contradicts the EPR2 decomposition.

1.5. Simulating nonlocality

It is straightforward to adapt our classical protocol to simulate the statistics obtained from arbitrary local POVMs on any entangled qudit-qubit state. Indeed, all PM protocols can be adapted to Bell scenarios [15]. For that, Alice chooses her measurement, an arbitrary POVM on a d_Q-dimensional quantum system. Then, she produces an output according to the marginal distribution of her POVM elements and, depending on her outcome, calculates the post-measurement state of Bob's qubit. Finally, she simply uses the classical protocol for the PM scenario to send that qubit state to Bob. Thus, our protocol immediately extends the best previously known one, due to Toner and Bacon

Scenario	This work	Ref. [10]
PM with qubit	2 bits, POVMs	2 bits, only Proj.
Bell with 2 qubits	2 bits, POVMs	2 bits, only Proj.
Bell with singlet	1 bit, Proj.-POVM	1 bit, Proj.-Proj.

Table 1.1.: Comparison between our protocol and the one by Toner and Bacon, previously the best protocol for these scenarios but restricted to only projective measurements (denoted as Proj. in this table) on Bob's side. Our protocols use the same resources, but Bob is allowed to perform POVMs.

[10], to Bell scenarios involving POVMs. At the same time, we use the same amount of classical communication, in fact, two bits.

However, Toner and Bacon also show that only a single bit is necessary to simulate local projective measurements on a qubit pair in the singlet state $|\Psi^-\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$. We can also extend that result by constructing a novel one bit protocol. Here, Alice is restricted to projective measurements with outcomes $a = \pm 1$, but Bob can perform arbitrary POVMs.

1. Alice and Bob share two normalized vectors $\vec{\lambda}_1', \vec{\lambda}_2 \in \mathbb{R}^3$, which are uniformly distributed on the unit radius sphere S_2.
2. Instead of performing a projective measurement with projectors $|\pm\vec{x}\rangle\langle\pm\vec{x}| = (\mathbb{1} \pm \vec{x}\cdot\vec{\sigma})/2$, Alice outputs $a = -\operatorname{sgn}(\vec{x}\cdot\vec{\lambda}_1')$ and sends the bit $c = \operatorname{sgn}(\vec{x}\cdot\vec{\lambda}_1')\cdot\operatorname{sgn}(\vec{x}\cdot\vec{\lambda}_2)$ to Bob. Here, $\operatorname{sgn}(z) = 1$ when $z \geq 0$ and $\operatorname{sgn}(z) = -1$ when $z < 0$.
3. Bob flips the vector $\vec{\lambda}_2$ if and only if $c = -1$. More formally, he sets $\vec{\lambda}_2' := c\,\vec{\lambda}_2$.
4. Same as "Step 4" in the original prepare-and-measure protocol.

Since $\vec{\lambda}_1'$ is uniformly distributed on S_2, we obtain the correct marginal probabilities $p(a) = 1/2$ for Alice. Furthermore, when Alice outputs $a = +1$, $\vec{\lambda}_1'$ and $\vec{\lambda}_2'$ are distributed on S_2 according to $\rho(\vec{\lambda}_i') = H(-\vec{x}\cdot\vec{\lambda}_i')/(2\pi)$. This corresponds precisely to a classical description of Bob's post-measurement state $-\vec{x}$ (compare with the text below Fig. 1.2). When Alice outputs $a = -1$, the two vectors are distributed according to $\rho(\vec{\lambda}_i') = H(+\vec{x}\cdot\vec{\lambda}_i')/(2\pi)$, which corresponds to the correct post-measurement state $+\vec{x}$. Therefore, Bob can apply the same response function ("Step 4") as in the original PM protocol, which immediately yields the correct quantum probabilities. Additionally, since singlet correlations have no local part [93, 94], one bit of communication is necessary in each round, ensuring the optimality of this protocol. Clearly, this protocol can be easily adapted to any maximally entangled qubit pair by rotating either Alice's or Bob's measurement basis.

1.6. Conclusion

We have proven that two bits of communication are necessary and sufficient in order to classically simulate the most general predictions of quantum theory in a qubit prepare-and-measure scenario. Our results also have immediate implications for simulations of nonlocality in scenarios featuring POVMs. In this way, we generalised the well-known protocols of Toner and Bacon [10] from projective measurements to the most general qubit measurements (POVMs). Interestingly, this comes with no increase in the classical cost. See Table 1.1 for an overview.

A natural direction is to consider classical simulations for higher-dimensional quantum PM scenarios ($d_Q > 2$), or scenarios involving entanglement. Notably, the latter can sometimes be isomorphic to the former [95]. Although this has received some attention [53, 54, 58, 96], few general results are known. Most notably, it is still an open problem whether a qutrit ($d_Q = 3$) PM scenario can be classically simulated with a finite message alphabet ($d_C < \infty$).

Acknowledgments

We thank Časlav Brukner, Valerio Scarani, Peter Sidajaya, Isadora Veeren, Bai Chu Yu for fruitful discussions. M.J.R. and M.T.Q. acknowledge financial support from the Austrian Science Fund (FWF) through BeyondC (F7103-N38), the Project No. I-2906, as well as support by the John Templeton Foundation through Grant 61466, The Quantum Information Structure of Spacetime (qiss.fr), the Foundational Questions Institute (FQXi) and the research platform TURIS. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801110. It reflects only the authors' view, the EU Agency is not responsible for any use that may be made of the information it contains. ESQ has received funding from the Austrian Federal Ministry of Education, Science and Research (BMBWF). A. T. is supported by the Wenner-Gren Foundation and by the Knut and Alice Wallenberg Foundation through the Wallenberg Center for Quantum Technology (WACQT).

1.A. Proof of classical simulation protocol

In this section, we prove that the classical protocol based on two bits simulates qubit correlations in the PM scenario. First, we give a modified version of the protocol and show that both versions lead to the same statistics.

1.A.1. Modified version of the protocol

The modified protocol, in which Bob sends one bit to Alice and afterwards Alice sends only one bit back to Bob, is:

1. Alice and Bob share two normalized vectors $\vec{\lambda}_1, \vec{\lambda}_2 \in \mathbb{R}^3$, which are uniformly distributed on the unit radius sphere S_2.
2. Instead of performing a POVM with elements $B_b = 2p_b\, |\vec{y}_b\rangle\langle\vec{y}_b|$, Bob picks one vector $\vec{y}_b$ from the set $\{\vec{y}_b\}$ according to the probabilities $\{p_b\}$. Then he sets $k = 1$ if $|\vec{\lambda}_1 \cdot \vec{y}_b| \geq |\vec{\lambda}_2 \cdot \vec{y}_b|$ and $k = 2$ otherwise. Afterwards, he sends the bit k to Alice.
3. Given that $\rho = \big(\mathbb{1} + \vec{x} \cdot \vec{\sigma}\big)/2$ is the pure qubit state Alice wants to send, she only sends the bit $c_k = H(\vec{x} \cdot \vec{\lambda}_k)$ to Bob.
4. Bob flips the vector $\vec{\lambda}_k$ if the bit c_k is zero. More formally, he sets $\vec{\lambda} := (-1)^{1+c_k}\vec{\lambda}_k$.
5. Finally, Bob outputs b with probability

$$p_B(b|\{B_b\}, \vec{\lambda}) = \frac{p_b\ \Theta(\vec{y}_b \cdot \vec{\lambda})}{\sum_j\ p_j\ \Theta(\vec{y}_j \cdot \vec{\lambda})}\,. \tag{1.7}$$

In the original version, Alice sends the two bits c_1 and c_2 that Bob needs to define the two vectors $\vec{\lambda}_i^{\,'} := (-1)^{1+c_i}\vec{\lambda}_i$. Afterwards, Bob chooses one of the two vectors $\vec{\lambda}_i^{\,'}$ according to the test $|\vec{\lambda}_1^{\,'} \cdot \vec{y}_b| \geq |\vec{\lambda}_2^{\,'} \cdot \vec{y}_b|$ and proceeds only with the chosen vector $\vec{\lambda} := \vec{\lambda}_k^{\,'}$. However, Bob's choice does only depend on the two vectors $\vec{\lambda}_1$ and $\vec{\lambda}_2$ but not on the bits c_i he received from Alice since $|\vec{\lambda}_i^{\,'} \cdot \vec{y}_b| = |(-1)^{1+c_i}\vec{\lambda}_i \cdot \vec{y}_b| = |\vec{\lambda}_i \cdot \vec{y}_b|$. This observation allows us to modify the protocol. In the modified version, he makes his choice between $\vec{\lambda}_1$ and $\vec{\lambda}_2$ first. Afterwards, he informs Alice of his choice $\vec{\lambda}_k$ and Alice sends only the bit c_k to Bob. This is enough for Bob to define the same $\vec{\lambda} := (-1)^{1+c_k}\vec{\lambda}_k$.

1.A.2. Proof of simulation protocol

Before we present the proof, we show that the protocol is well-defined. More precisely, we can check that $p_B(b|\{B_b\}, \vec{\lambda})$ are well-defined probabilities. In order to see this, note that $0 \leq p_B(b|\{B_b\}, \vec{\lambda}) \leq 1$ follows from $\Theta(z) \geq 0$ (for every $z \in \mathbb{R}$) and $p_j \geq 0$ (for every j). Furthermore, we can check that

$$\sum_b p_B(b|\{B_b\}, \vec{\lambda}) = \frac{\sum_b\ p_b\ \Theta(\vec{y}_b \cdot \vec{\lambda})}{\sum_j\ p_j\ \Theta(\vec{y}_j \cdot \vec{\lambda})} = 1\,, \tag{1.8}$$

to ensure normalisation.

Theorem 1.1. *The above protocol reproduces the correct quantum probabilities. More precisely, for a given pure qubit state $\rho = \big(\mathbb{1} + \vec{x}\cdot\vec{\sigma}\big)/2$ and POVM elements $B_b = 2p_b\,|\vec{y}_b\rangle\langle\vec{y}_b|$, the total probability that Bob outputs b is*

$$p_C(b|\rho,\{B_b\}) = p_b(1+\vec{x}\cdot\vec{y}_b) = \mathrm{tr}(\rho\,B_b) = p_Q(b|\rho,\{B_b\})\,. \tag{1.9}$$

Proof. To check that the protocol outputs the correct probabilities, it is slightly more convenient to follow the modified version. We go through all the steps of the protocol and determine first the distribution of the vector $\vec{\lambda}_k$ after "Step 2", second the distribution of Bob's chosen vector $\vec{\lambda}$ after he performed "Step 4" and third we calculate the total probability that he outputs b in "Step 5".

1. Distribution of $\vec{\lambda}_k$ after "Step 2":
Alice and Bob share two vectors uniformly and independently distributed along the unit sphere $\vec{\lambda}_1, \vec{\lambda}_2 \in S_2$. Consider a round in which Bob has picked the POVM element that corresponds to the vector $\vec{y}_b$. Then he is choosing the vector $\vec{\lambda}_1$ if $|\vec{\lambda}_1\cdot\vec{y}_b| \geq |\vec{\lambda}_2\cdot\vec{y}_b|$ and $\vec{\lambda}_2$ otherwise. It follows from Degorre et al. "Theorem 6 (The "choice" method)" [52] that the resulting distribution of the chosen vector $\vec{\lambda}_k$ is exactly:

$$\rho_b(\vec{\lambda}_k|\vec{y}_b) = \frac{1}{2\pi}|\vec{y}_b\cdot\vec{\lambda}_k|\,. \tag{1.10}$$

Since he is choosing $\vec{y}_b$ with probability p_b the total distribution of the chosen vector $\vec{\lambda}_k$ is:

$$\rho(\vec{\lambda}_k|\{B_b\}) = \sum_b p_b\;\rho_b(\vec{\lambda}_k|\vec{y}_b) = \frac{1}{2\pi}\sum_b p_b\;|\vec{y}_b\cdot\vec{\lambda}_k|\,. \tag{1.11}$$

2. Distribution of $\vec{\lambda}$ after "Step 4":
Now Bob checks the received bit $c_k = H(\vec{x}\cdot\vec{\lambda}_k)$. He flips his chosen vector $\vec{\lambda}_k \to -\vec{\lambda}_k$ if and only if the received bit is zero. As a result, the distribution becomes:

$$\rho(\vec{\lambda}_k|\vec{x},\{B_b\}) = 2\cdot H(\vec{x}\cdot\vec{\lambda}_k)\cdot\rho(\vec{\lambda}_k|\{B_b\}) = \frac{H(\vec{x}\cdot\vec{\lambda}_k)}{\pi}\sum_b p_b\;|\vec{y}_b\cdot\vec{\lambda}_k|\,. \tag{1.12}$$

To see that this is true, note that if $H(\vec{x}\cdot\vec{\lambda}_k) = 1$, Bob does not flip the vector $\vec{\lambda}_k$ and the distribution remains unchanged. If $H(\vec{x}\cdot\vec{\lambda}_k) = 0$, Bob flips the vector. However, the distribution $\rho(\vec{\lambda}_k|\{B_b\})$ is point symmetric:

$$\rho(-\vec{\lambda}_k|\{B_b\}) = \frac{1}{2\pi}\sum_b p_b\;|-\vec{y}_b\cdot\vec{\lambda}_k| = \frac{1}{2\pi}\sum_b p_b\;|\vec{y}_b\cdot\vec{\lambda}_k| = \rho(\vec{\lambda}_k|\{B_b\})\,, \tag{1.13}$$

from which the above expression follows. From here one, we can drop the index k in $\vec{\lambda}_k$. We show below (Lemma 1.2) that $\sum_b p_b \; |\vec{y}_b \cdot \vec{\lambda}| = 2 \sum_b p_b \; \Theta(\vec{y}_b \cdot \vec{\lambda})$ and we use this to rewrite the distribution $\rho(\vec{\lambda}|\vec{x}, \{B_b\})$ into:

$$\rho(\vec{\lambda}|\vec{x}, \{B_b\}) = \frac{2H(\vec{x} \cdot \vec{\lambda})}{\pi} \sum_b \; p_b \; \Theta(\vec{y}_b \cdot \vec{\lambda}) \,. \tag{1.14}$$

3. Total probability that Bob outputs b in "Step 5":
Finally, we are in a position to calculate the total probability that Bob outputs b in "Step 5". Here, we use the expressions given in (1.7) and (1.14) to obtain:

$$\begin{aligned} p(b|\vec{x}, \{B_b\}) &= \int_{S_2} p_B(b|\{B_b\}, \vec{\lambda}) \cdot \rho(\vec{\lambda}|\vec{x}, \{B_b\}) \; \mathrm{d}\vec{\lambda} & (1.15)\\ &= \frac{2p_b}{\pi} \int_{S_2} H(\vec{x} \cdot \vec{\lambda}) \cdot \; \Theta(\vec{y}_b \cdot \vec{\lambda}) \; \mathrm{d}\vec{\lambda} = p_b(1 + \vec{x} \cdot \vec{y}_b) \,. & (1.16) \end{aligned}$$

We evaluate the integral in Lemma 1.1 below. This equals exactly the required quantum statistics. □

1.A.3. Evaluation of the integral

Lemma 1.1. *Given two normalized vectors $\vec{x}, \vec{y} \in \mathbb{R}^3$ on the unit sphere S_2, it holds that:*

$$\frac{1}{\pi} \int_{S_2} H(\vec{x} \cdot \vec{\lambda}) \cdot \; \Theta(\vec{y} \cdot \vec{\lambda}) \; \mathrm{d}\vec{\lambda} = \frac{1}{2}(1 + \vec{x} \cdot \vec{y}) \,, \tag{1.17}$$

where $H(z)$ is the Heaviside function ($H(z) = 1$ if $z \geq 0$ and $H(z) = 0$ if $z < 0$) and $\Theta(z) := H(z) \cdot z$.

Proof. Note that both functions in the integral $H(\vec{x} \cdot \vec{\lambda})$ and $\Theta(\vec{y} \cdot \vec{\lambda})$ have support in only one half of the total sphere (the hemisphere centred around $\vec{x}$ and $\vec{y}$, respectively). For example, if $\vec{y} = -\vec{x}$ these two hemispheres are exactly opposite of each other and the integral becomes zero. For all other cases, we can observe that the value of the integral depends only on the angle between $\vec{x}$ and $\vec{y}$, because the whole expression is spherically symmetric. Therefore, it is enough to evaluate the integral for $\vec{x} = (0, 1, 0)^T$ and $\vec{y} = (-\sin\beta, \cos\beta, 0)^T$, where we can choose without loss of generality $0 \leq \beta \leq \pi$. Furthermore, we can use spherical coordinates for $\vec{\lambda} = (\sin\theta \cdot \cos\phi, \sin\theta \cdot \sin\phi, \cos\theta)$ (note that $|\vec{\lambda}| = 1$). With this choice of coordinates, the region in which both factors have non-zero support becomes exactly $\beta \leq \phi \leq \pi$ (at the same time, θ is unrestricted, $0 \leq \theta \leq \pi$). More precisely, $0 \leq \phi \leq \pi$ is the support for $H(\vec{x} \cdot \vec{\lambda})$ and $\beta \leq \phi \leq \pi + \beta$ is the support for $\Theta(\vec{y} \cdot \vec{\lambda})$. In this way, the integral becomes:

$$\begin{aligned} \frac{1}{\pi} \int_0^{2\pi} \int_0^{\pi} H(\vec{x} \cdot \vec{\lambda}) \cdot \; \Theta(\vec{y} \cdot \vec{\lambda}) \cdot \sin\theta \; \mathrm{d}\theta \; \mathrm{d}\phi =& \frac{1}{\pi} \int_\beta^{\pi} \int_0^{\pi} \; \sin\phi \cdot \sin^2\theta \; \mathrm{d}\theta \; \mathrm{d}\phi & (1.18)\\ =& \frac{1}{2}(1 + \cos\beta) = \frac{1}{2}(1 + \vec{x} \cdot \vec{y}) \,. & (1.19) \end{aligned}$$

□

It was recognized many times in the literature [50, 15, 52] that the last expression exactly reproduces the statistics of measurements on qubits. However, in previous protocols, Alice was choosing a vector to create a distribution according to $\Theta(\vec{x}\cdot\vec{\lambda})$ and Bob outputs according to $H(\vec{y}\cdot\vec{\lambda})$. Here, we use the self-duality of quantum mechanics, which allows us to interchange the roles of states and measurements. In this sense, instead of Alice, Bob is choosing a vector to create a distribution like $\Theta(\vec{y}_b\cdot\vec{\lambda})$ and Alice contributes the term $H(\vec{x}\cdot\vec{\lambda})$ by telling Bob to flip that vector or not.

1.A.4. Proof of a useful identity

For the following Lemma, it is important to notice that for every POVM it holds that $\sum_b p_b\,\vec{y}_b=\vec{0}$. This follows from $\sum_b B_b=\mathbb{1}$ with $B_b=2p_b\,|\vec{y}_b\rangle\langle\vec{y}_b|$, where $\sum_b p_b=1$ and $|\vec{y}_b\rangle\langle\vec{y}_b|=(\mathbb{1}+\vec{y}_b\cdot\vec{\sigma})/2$:

$$\mathbb{1}=\sum_b 2p_b\,|\vec{y}_b\rangle\langle\vec{y}_b|=\sum_b p_b(\mathbb{1}+\vec{y}_b\cdot\vec{\sigma})=\sum_b p_b\;\mathbb{1}+\sum_b p_b\;\vec{y}_b\cdot\vec{\sigma}=\mathbb{1}+\sum_b p_b\;\vec{y}_b\cdot\vec{\sigma}\,. \tag{1.20}$$

From this we conclude that $\sum_b p_b\;\vec{y}_b\cdot\vec{\sigma}=0$, which can only hold if $\sum_b p_b\;\vec{y}_b=\vec{0}$.

Lemma 1.2. *Given a set of vectors $\vec{y}_b\in S_2$ that satisfy $\sum_b p_b\;\vec{y}_b=\vec{0}$ and the function $\Theta(z)$, which is defined by $\Theta(z)=z$ if $z\geq 0$ and $\Theta(z)=0$ if $z<0$, it holds for every $\vec{\lambda}\in S_2$ that:*

$$\sum_b p_b\;|\vec{y}_b\cdot\vec{\lambda}|=2\sum_b p_b\;\Theta(\vec{y}_b\cdot\vec{\lambda})\,. \tag{1.21}$$

Proof. First we prove that $\sum_b p_b\;\Theta(\vec{y}_b\cdot\vec{\lambda})=\sum_b p_b\;\Theta(-\vec{y}_b\cdot\vec{\lambda})$. Here, we use that $z=\Theta(z)-\Theta(-z)$ (for all $z\in\mathbb{R}$):

$$\vec{0}=\sum_b p_b\;\vec{y}_b\quad\Longrightarrow\quad 0=\vec{0}\cdot\vec{\lambda}=\sum_b p_b\;\vec{y}_b\cdot\vec{\lambda}=\sum_b p_b\;(\Theta(\vec{y}_b\cdot\vec{\lambda})-\Theta(-\vec{y}_b\cdot\vec{\lambda})) \tag{1.22}$$

$$=\sum_b p_b\;\Theta(\vec{y}_b\cdot\vec{\lambda})-\sum_b p_b\;\Theta(-\vec{y}_b\cdot\vec{\lambda})\,. \tag{1.23}$$

In the second step, we use this observation and $|z|=\Theta(z)+\Theta(-z)$ (for all $z\in\mathbb{R}$) to calculate:

$$\sum_b p_b\;|\vec{y}_b\cdot\vec{\lambda}|=\sum_b p_b\;(\Theta(\vec{y}_b\cdot\vec{\lambda})+\Theta(-\vec{y}_b\cdot\vec{\lambda}))=\sum_b p_b\;\Theta(\vec{y}_b\cdot\vec{\lambda})+\sum_b p_b\;\Theta(-\vec{y}_b\cdot\vec{\lambda}) \tag{1.24}$$

$$=2\sum_b p_b\;\Theta(\vec{y}_b\cdot\vec{\lambda})\,. \tag{1.25}$$

□

1.B. No classical simulation with a one-bit part

In this section, we show that every protocol that simulates a qubit in a PM scenario cannot have a part in which Alice communicates only a single bit to Bob. Interestingly, for our argument it is enough to consider only projective measurements. Hence, we can write the two projection operators for Bob as $|\pm\vec{y}\rangle\langle\pm\vec{y}| = (\mathbb{1} \pm \vec{y}\cdot\vec{\sigma})/2$. As before, Alice can choose an arbitrary qubit state $\rho = (\mathbb{1} + \vec{x}\cdot\vec{\sigma})/2$ that we simply denote with its Bloch vector $\vec{x}$. With that notation, in a classical protocol that simulates a qubit in a PM scenario, Bob has to output $b = \pm 1$ with probability:

$$p^{PM}(b|\vec{x},\vec{y}) = \frac{1}{2}(1 + b\;\vec{x}\cdot\vec{y})\,. \tag{1.26}$$

We show that, given a protocol that simulates the qubit in the PM scenario with a non-zero one-bit part exists, it can be rewritten into a protocol that simulates the singlet with a non-zero local part. The latter is prohibited by the result of Elitzur, Popescu and Rohrlich [93] (see also Barrett et al. [94]) and our hypothesis follows by contradiction. To fix the notation, if Alice and Bob want to reproduce the statistics of local projective measurements on the singlet state $|\Psi^-\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$, they can choose measurement projectors $|\pm\vec{x}\rangle\langle\pm\vec{x}| = (\mathbb{1} \pm \vec{x}\cdot\vec{\sigma})/2$ (for Alice) and $|\pm\vec{y}\rangle\langle\pm\vec{y}| = (\mathbb{1} \pm \vec{y}\cdot\vec{\sigma})/2$ (for Bob). Then, the task becomes to output $a, b = \pm 1$ with probabilities:

$$p^{\Psi^-}(a,b|\vec{x},\vec{y}) = \frac{1}{4}(1 - (b\cdot a)\;\vec{x}\cdot\vec{y})\,. \tag{1.27}$$

The similar form of these two expressions already suggests a connection between a protocol that simulates a qubit in a PM scenario and a protocol that simulates the singlet. We use the index "PM" and "Ψ^-" to distinguish theses two scenarios. We want to mention that the following statement also covers the scenario of no communication in some fraction of rounds. This is just a special case of a one-bit strategy in which Bob's response does not depend on the received message.

Lemma 1.3. *Given a protocol that exactly simulates any qubit strategy in a prepare-and-measure scenario. The fraction of rounds in which Alice is communicating only a single bit to Bob has measure zero. More precisely, we can decompose such a protocol into:*

$$\begin{aligned} p^{PM}(b|\vec{x},\vec{y}) = &\int_\lambda d\lambda\,\pi(\lambda) \sum_{c=\pm 1} p_A(c|\vec{x},\lambda) p_B(b|\vec{y},c,\lambda) \\ &+ \int_{\tilde{\lambda}} d\tilde{\lambda}\,\pi(\tilde{\lambda}) \sum_{m=1}^{d} p_A(m|\vec{x},\tilde{\lambda}) p_B(b|\vec{y},m,\tilde{\lambda})\,, \end{aligned} \tag{1.28}$$

and it has to hold that $\int_\lambda d\lambda\,\pi(\lambda) = 0$. Here, the first term are all the strategies that can be implemented with a single bit $c = \pm 1$ of communication and the second term contains all the strategies that require a longer message m (with $d > 2$).

Proof. To recapitulate, in those rounds, where Alice is allowed to send only a single bit to Bob, Alice's bit $c = \pm 1$ can depend only on her input $\vec{x}$ and the shared random variable

λ (denoted as $p_A(c|\vec{x},\lambda)$) and Bob has to determine his output b based on his input $\vec{y}$, the message c and the shared random variable λ (denoted as $p_B(b|\vec{y},c,\lambda)$). The important observation is, given that Alice sends the bit $c=+1$, if she wants to transmit the state $\vec{x}$, she necessarily has to send the bit $c=-1$, if she wants to transmit the state $-\vec{x}$. To see that this must be true, consider that Bob chooses in that round the measurement basis $\vec{y}=\vec{x}$. In that situation, he necessarily has to discriminate between the two states $\vec{x}$ and $-\vec{x}$. This is not possible if Alice sends the same bit for both states $\vec{x}$ and $-\vec{x}$. Therefore, $p_A(c|\vec{x},\lambda)=p_A(-c|-\vec{x},\lambda)$.

On the other hand, given that Bob chooses the measurement basis $\vec{y}$ and wants to produce the output b, it might be that Alice has chosen the state $\vec{x}=\vec{y}$. If $c=+1$ is the message for the state $\vec{x}=\vec{y}$ in this round, it has to hold that $p_B(b=+1|\vec{y},c=+1,\lambda)=1$ and since $c=-1$ is then necessarily the message for the state $-\vec{x}=-\vec{y}$ it has to hold that $p_B(b=-1|\vec{y},c=-1,\lambda)=1$. Analogously, it is also possible that $c=-1$ is the message for the state $\vec{x}=\vec{y}$ and $c=+1$ is the message for the state $-\vec{x}=-\vec{y}$, in which a similar argument leads to $p_B(b=+1|\vec{y},c=+1,\lambda)=p_B(b=-1|\vec{y},c=-1,\lambda)=0$. In any case, it has to hold that $p_B(b|\vec{y},c=+1,\lambda)=p_B(-b|\vec{y},c=-1,\lambda)$.

Now they can use a protocol that simulates a qubit in a PM scenario to simulate the singlet state [15]. More precisely, Alice chooses her measurement basis $\vec{x}$ and tosses a balanced coin (heads and tails with probability 1/2 each). If the coin shows heads, she outputs $a=+1$ and uses the PM protocol from above to send the state $-\vec{x}$ to Bob, whereas if the coin shows tails, she outputs $a=-1$ and uses the protocol to sends the state $+\vec{x}$ to Bob (be aware of the anti-correlation in the singlet state). This procedure simulates the singlet state since:

$$\begin{aligned} p^{\Psi^-}(a,b|\vec{x},\vec{y}) &= \frac{1}{2}\cdot\delta_{a,+1}\cdot p^{PM}(b|-\vec{x},\vec{y})+\frac{1}{2}\cdot\delta_{a,-1}\cdot p^{PM}(b|+\vec{x},\vec{y}) && (1.29)\\ &= \frac{1}{4}\cdot\delta_{a,+1}\cdot(1-b\ \vec{x}\cdot\vec{y})+\frac{1}{4}\cdot\delta_{a,-1}\cdot(1+b\ \vec{x}\cdot\vec{y}) && (1.30)\\ &= \frac{1}{4}(1-(b\cdot a)\ \vec{x}\cdot\vec{y})\,. && (1.31)\end{aligned}$$

However, we can also write down the explicit protocol:

$$p^{\Psi^-}(a,b|\vec{x},\vec{y}) =\frac{1}{2}\cdot\delta_{a,+1}\cdot p^{PM}(b|-\vec{x},\vec{y})+\frac{1}{2}\cdot\delta_{a,-1}\cdot p^{PM}(b|+\vec{x},\vec{y}) \qquad (1.32)$$

$$\begin{aligned} &=\frac{1}{2}\cdot\delta_{a,+1}\cdot\int_\lambda \mathrm{d}\lambda\,\pi(\lambda)\sum_{c=\pm1}p_A(c|-\vec{x},\lambda)p_B(b|\vec{y},c,\lambda)\\ &+\frac{1}{2}\cdot\delta_{a,+1}\cdot\int_{\tilde\lambda}\mathrm{d}\tilde\lambda\,\pi(\tilde\lambda)\sum_{m=1}^{d}p_A(m|-\vec{x},\tilde\lambda)p_B(b|\vec{y},m,\tilde\lambda)\\ &+\frac{1}{2}\cdot\delta_{a,-1}\cdot\int_\lambda \mathrm{d}\lambda\,\pi(\lambda)\sum_{c=\pm1}p_A(c|+\vec{x},\lambda)p_B(b|\vec{y},c,\lambda)\\ &+\frac{1}{2}\cdot\delta_{a,-1}\cdot\int_{\tilde\lambda}\mathrm{d}\tilde\lambda\,\pi(\tilde\lambda)\sum_{m=1}^{d}p_A(m|+\vec{x},\tilde\lambda)p_B(b|\vec{y},m,\tilde\lambda)\,.\end{aligned} \qquad (1.33)$$

As before, the protocol to simulate the singlet is decomposed into a one-bit part and a part that requires more communication. The one bit part is the sum of all the terms that contain λ (and not $\tilde{\lambda}$) as the shared variable. Together, they can be written as $\int_\lambda \mathrm{d}\lambda\, \pi(\lambda)\, p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda)$ where:

$$\begin{aligned} p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda) :=& \frac{1}{2}\cdot \delta_{a,+1}\cdot p_A(c=+1|-\vec{x},\lambda) p_B(b|\vec{y},c=+1,\lambda) \\ &+ \frac{1}{2}\cdot \delta_{a,+1}\cdot p_A(c=-1|-\vec{x},\lambda) p_B(b|\vec{y},c=-1,\lambda) \\ &+ \frac{1}{2}\cdot \delta_{a,-1}\cdot (p_A(c=+1|+\vec{x},\lambda) p_B(b|\vec{y},c=+1,\lambda) \\ &+ \frac{1}{2}\cdot \delta_{a,-1}\cdot p_A(c=-1|+\vec{x},\lambda) p_B(b|\vec{y},c=-1,\lambda)\,. \end{aligned} \tag{1.34}$$

With the above relations $p_A(c|\vec{x},\lambda) = p_A(-c|-\vec{x},\lambda)$ and $p_B(b|\vec{y},c=+1,\lambda) = p_B(-b|\vec{y},c=-1,\lambda)$, we can rewrite this expression into:

$$\begin{aligned} p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda) =& \frac{1}{2}\cdot \delta_{a,+1}\cdot p_A(c=-1|\vec{x},\lambda) p_B(b|\vec{y},c=+1,\lambda) \\ &+ \frac{1}{2}\cdot \delta_{a,+1}\cdot p_A(c=+1|\vec{x},\lambda) p_B(-b|\vec{y},c=+1,\lambda) \\ &+ \frac{1}{2}\cdot \delta_{a,-1}\cdot p_A(c=+1|\vec{x},\lambda) p_B(b|\vec{y},c=+1,\lambda) \\ &+ \frac{1}{2}\cdot \delta_{a,-1}\cdot p_A(c=-1|\vec{x},\lambda) p_B(-b|\vec{y},c=+1,\lambda)\,. \end{aligned} \tag{1.35}$$

The important observation is now that these correlations $p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda)$ can be realized with purely local strategies. More precisely, Alice and Bob share an additional random bit $r = \pm 1$ (with probability 1/2 each). Given her measurement setting $\vec{x}$ and the shared random variable λ, Alice samples $c = \pm 1$ according to the probabilities $p_A(c|\vec{x},\lambda)$ (as for the case of the message c in the PM scenario). However, instead of sending the message c to Bob, she outputs $a = -r \cdot c$. At the same time, Bob outputs $b = r \cdot b_+$ where b_+ is sampled according to the probabilities $p_B(b_+|\vec{y},c=+1,\lambda)$. If both follow that strategy and $r = +1$, they implement the behaviour

$$\begin{aligned} p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda,r=+1) :=& \delta_{a,+1}\cdot p_A(c=-1|\vec{x},\lambda)\cdot p_B(b|\vec{y},c=+1,\lambda) \\ &+ \delta_{a,-1}\cdot p_A(c=+1|\vec{x},\lambda)\cdot p_B(b|\vec{y},c=+1,\lambda)\,. \end{aligned} \tag{1.36}$$

On the other hand, if $r = -1$ they implement

$$\begin{aligned} p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda,r=-1) :=& \delta_{a,+1}\cdot p_A(c=+1|\vec{x},\lambda)\cdot p_B(-b|\vec{y},c=+1,\lambda) \\ &+ \delta_{a,-1}\cdot p_A(c=-1|\vec{x},\lambda)\cdot p_B(-b|\vec{y},c=+1,\lambda)\,. \end{aligned} \tag{1.37}$$

It is easy to check that the weighted sum of these two expressions equals exactly the expression $p(a,b|\vec{x},\vec{y},\lambda)$ given in Eq. (1.35):

$$p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda) = \frac{1}{2}\cdot p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda,r=+1) + \frac{1}{2}\cdot p^{\Psi^-}(a,b|\vec{x},\vec{y},\lambda,r=-1)\,. \tag{1.38}$$

Therefore, we have optimized the above protocol given in Eq. (1.33): Whenever Alice and Bob draw a λ that corresponds to a one-bit part for the PM scenario, they can switch to the local strategy if they want to simulate the singlet. In the remaining rounds (according to shared randomness $\tilde{\lambda}$), where Alice was allowed to send more information, they do the same as in the case of the PM scenario: Alice outputs $-r$ and sends the message according to the state $r\vec{x}$. Bob outputs b according to his message m and his measurement basis $\vec{y}$.

Hence, given a simulation of the PM scenario with a non-zero one-bit part exists, we found a simulation of the singlet state with a non-zero local part. Since this is in contradiction with the result of Elitzur, Popescu, and Rohrlich [93] (see also Barrett et al. [94]), it proves our hypothesis.

□

1.C. Linear programming and classical PM scenarios

Let us consider a fixed PM scenario where Alice can prepare $I_A \in \mathbb{N}$ different inputs and Bob has $I_B \in \mathbb{N}$ different measurements with $O_B \in \mathbb{N}$ outcomes each. The problem of deciding if a set of probabilities $\{p(b|x,y)\}$ can be obtained by Alice sending classical d_C-dimensional systems to Bob is phrased as:

$$\begin{aligned}
\text{given: } & \{p(b|x,y)\},\ d_C && (1.39)\\
\text{find } & \pi, p_A, p_B && (1.40)\\
\text{s.t.: } & p(b|x,y) = \int_\lambda \mathrm{d}\lambda \sum_{c=1}^{d_C} \pi(\lambda)\, p_A\big(c|x,\lambda\big)\, p_B\big(b|y,c,\lambda\big), \quad \forall b,x,y && (1.41)\\
& \pi(\lambda) \geq 0,\ \forall\lambda && (1.42)\\
& \int_\lambda \mathrm{d}\lambda \pi(\lambda) = 1 && (1.43)\\
& p_A\big(c|x,\lambda\big) \geq 0,\ \forall c,x,\lambda && (1.44)\\
& \sum_{c=1}^{d_C} p_A\big(c|x,\lambda\big) = 1,\ \forall x,\lambda, && (1.45)\\
& p_B\big(b|y,c,\lambda\big) \geq 0, \quad \forall b,x,y && (1.46)\\
& \sum_{b=1}^{O_B} p_B\big(b|y,c,\lambda\big) = 1,\ \forall y,c,\lambda\,. && (1.47)
\end{aligned}$$

Since in Eq. (1.41) we have a product of the optimisation variables, the above problem is not in a linear programming form. In order to rewrite it as a linear programming, we note that, similarly to Bell nonlocality [77], the classical message c sent by Alice may be chosen deterministically for a given x and λ. This is true because the choice of the set of distributions $\{p_A(c|x,\lambda)\}$ form a polytope where the vertices are given by deterministic distributions $D_A(c|x,\lambda)$, where $\lambda \in \{1,\ldots,d_C^{I_A}\}$. We then define

$p'_B\big(b|y,c,\lambda\big) := \pi(\lambda)p_B\big(b|y,c,\lambda\big)$, a transformation which allows us to write the problem described in Eqs. (1.39) as,

$$\text{given: } \{p(b|x,y)\},\ d_C,\ \{D_A(c|x,\lambda)\} \tag{1.48}$$

$$\text{find} \quad \pi, p'_B \tag{1.49}$$

$$\text{s.t.:} \quad p(b|x,y) = \sum_{\lambda=1}^{d_C^{I_A}} \sum_{c=1}^{d_C} D_A\big(c|x,\lambda\big)\, p'_B\big(b|y,c,\lambda\big), \quad \forall b,x,y \tag{1.50}$$

$$p'_B\big(b|y,c,\lambda\big) \geq 0, \quad \forall b,y,\lambda \tag{1.51}$$

$$\sum_{b=1}^{O_B} p'_B\big(b|y,c,\lambda\big) = \pi(\lambda),\ \forall y,c,\lambda. \tag{1.52}$$

where $\sum_\lambda \pi(\lambda) = 1$ follows from the fact that $\sum_b p(b|x,y) = 1$. Note that now, all constraints are linear or positivity constraints, hence, the problem in Eqs. (1.48) is a linear program.

We remark that, the set of distributions $\{p_B(b|c,y,\lambda\}$ also form a polytope where the vertices are given by deterministic distributions $D_B(c|x,\lambda)$, where $\lambda \in \{1,\ldots,O_B^{I_B}\}$. Hence, one may construct a different linear program where both Alice and Bob have deterministic response functions. For practical reasons, this is often not a good choice, since it leads to a linear program with a big number of variables. In particular, the variable λ would be allowed to take $d_C^{I_A} O_B^{I_B d_C}$ different values as opposed to $d_C^{I_A}$. However, when considering a scenario where $d_c^{I_A} > O_B^{I_B d_C}$, it might be more efficient to set Bob as the part which performs deterministic strategies and to construct a different linear program by setting $p'_A\big(c|x,\lambda\big) := \pi(\lambda)p_A\big(c|x,\lambda\big)$.

1.C.1. Primal formulation in terms of white noise robustness

The linear program presented in Eqs. (1.48) is a simple feasibility problem, since it only requires the existence of a feasible solution. We now adapt this feasibility problem to obtain a robustness optimisation problem. Instead of simply asking whether a set of probabilities $\{p(b|x,y)\}$ may be simulated by classical systems of dimension d_C, we look for the critical visibility parameter $\eta \in [0,1]$ such that the probabilities given by $\eta\, p(b|x,y) + (1-\eta)\frac{1}{O_B}$ admit a classical d_C-dimensional description. For that, we write the following linear program:

$$\text{given: } \{p(b|x,y)\},\ \{D_A(.|x,\lambda)\},\ d_C \tag{1.53}$$

$$\max \quad \eta \tag{1.54}$$

$$\text{s.t.:} \quad \eta\, p(b|x,y) + (1-\eta)\frac{1}{O_B} = \sum_{c=1}^{d_C} \sum_{\lambda}^{d_C^{I_A}} p'_B(b|y,c,\lambda) D_A(c|x,\lambda), \quad \forall b,y,x \tag{1.55}$$

$$p'_B(b|y,c,\lambda) \geq 0, \quad \forall b,y,c,\lambda \tag{1.56}$$

$$\sum_b p'_B(b|y,c,\lambda) = \ \pi(\lambda)\,. \tag{1.57}$$

1.C.2. Classical dimension witness emerging from the dual problem

We will now show how to obtain a classical dimension witness from the linear program presented in Eqs. (1.53). For the sake of concreteness, we will explicitly obtain the dual form from the Lagrangian method, see [86] for an introduction. We start by setting the dual variables as:

$$\text{given: } \{p(b|x,y)\},\ \{D_A(c|x,\lambda)\}\ d_C \tag{1.58}$$

$$\max\quad \eta \tag{1.59}$$

$$\text{s.t.:}\quad \eta\, p(b|x,y) + (1-\eta)\frac{1}{O_B} \tag{1.60}$$

$$= \sum_{c=1}^{d_C}\sum_{\lambda=1}^{d_C^{I_A}} p'_B(b|y,c,\lambda) D_A(c|x,\lambda), \quad \forall b,y,x \qquad \Big[\text{dual: } \gamma(b|x,y)\Big] \tag{1.61}$$

$$p'_B(b|y,c,\lambda) \geq 0, \quad \forall b,y,c,\lambda \qquad \Big[\text{dual: } \rho(b|y,c,\lambda)\Big] \tag{1.62}$$

$$\sum_b p'_B(b|y,c,\lambda) = \ \pi(\lambda), \quad \forall y,c,\lambda \qquad \Big[\text{dual: } s(y,c,\lambda)\Big]. \tag{1.63}$$

The Lagrangian is then given by

$$L = \eta + \sum_{b,x,y}\gamma(b|xy)\Big(\eta\, p(b|x,y) + \frac{1}{O^B} - \frac{\eta}{O_B} - \sum_{c,\lambda} p'_B(b|y,c,\lambda) D_A(c|x,\lambda)\Big) \tag{1.64}$$

$$+ \sum_{b,y,a,\lambda} p'_B(b|y,c,\lambda)\rho(b|y,c,\lambda) \tag{1.65}$$

$$+ \sum_{y,c,\lambda} s(y,c,\lambda)\Big(\Big[\sum_b p'_B(b|y,c,\lambda)\Big] - \pi(\lambda)\Big)\,; \tag{1.66}$$

If we factorise the primal variables we have:

$$L = \eta\Big(1 + \Big[\sum_{b,x,y}\gamma(b|x,y)p(b|x,y)\Big] - \Big[\sum_{b,x,y}\frac{\gamma(b|x,y)}{O_B}\Big]\Big) \tag{1.67}$$

$$+ \sum_{b,y,c,\lambda} p'_B(b|y,c,\lambda)\Big(\Big[-\sum_x \gamma(b|x,y) D_A(c|x,\lambda)\Big] + \rho(b|y,c,\lambda) + s(y,c,\lambda)\Big) \tag{1.68}$$

$$+ \sum_\lambda \pi(\lambda)\Big(-\sum_{y,c} s(y,c,\lambda)\Big) \tag{1.69}$$

$$+ \sum_{b,x,y}\frac{\gamma(b|x,y)}{O_B}\,. \tag{1.70}$$

This leads to the dual:

$$\text{given: } \{p(b|x,y)\},\ \{D_A(c|x,\lambda)\},\ d_C \tag{1.71}$$

$$\min \quad \sum_{b,x,y} \frac{\gamma(b|x,y)}{O_B} \tag{1.72}$$

$$\text{s.t.:} \quad \rho(b|y,c,\lambda) \geq 0 \quad \forall b,y,c,\lambda \tag{1.73}$$

$$1 + \Big[\sum_{b,x,y} \gamma(b|x,y)p(b|x,y)\Big] = \sum_{b,x,y} \frac{\gamma(b|x,y)}{O_B} \tag{1.74}$$

$$\rho(b|y,c,\lambda) = \Big[\sum_{x} \gamma(b|x,y)D_A(a|x,\lambda)\Big] - s(y,c,\lambda) \quad \forall b,y,c,\lambda \tag{1.75}$$

$$\sum_{y,a} s(y,c,\lambda) = 0 \quad \forall\lambda \tag{1.76}$$

and, we can also combine Eq. (1.73) with Eq. (1.75) to write:

$$\text{given: } \{p(b|x,y),\ \{D_A(c|x,\lambda)\},\ d_C \tag{1.77}$$

$$\min \quad \sum_{b,x,y} \frac{\gamma(b|x,y)}{O_B} \tag{1.78}$$

$$\text{s.t.:} \quad \sum_{x} \gamma(b|x,y)D_A(c|x,\lambda) \geq s(y,c,\lambda) \quad \forall b,y,c,\lambda \tag{1.79}$$

$$1 + \Big[\sum_{b,x,y} \gamma(b|x,y)p(b|x,y)\Big] = \sum_{b,x,y} \frac{\gamma(b|x,y)}{O_B} \tag{1.80}$$

$$\sum_{y,c} s(y,c,\lambda) = 0 \quad \forall\lambda \tag{1.81}$$

Additionally, in order to have a more explicit hyperplane formulation, we use Eq. (1.78) and Eq. (1.74) to write:

$$\text{given: } \{p(b|x,y)\},\ \{D_A(c|x,\lambda)\},\ d_C \tag{1.82}$$

$$\min \quad 1 + \sum_{b,x,y} \gamma(b|x,y)p(b|x,y) \tag{1.83}$$

$$\text{s.t.:} \quad \sum_{x} \gamma(b|x,y)D_A(c|x,\lambda) \geq s(y,c,\lambda) \quad \forall b,y,c,\lambda \tag{1.84}$$

$$\sum_{b,x,y} \frac{\gamma(b|x,y)}{O_B} = 1 + \sum_{b,x,y} \gamma(b|x,y)p(b|x,y) \tag{1.85}$$

$$\sum_{y,a} s(y,c,\lambda) = 0 \quad \forall\lambda \tag{1.86}$$

We then see that the $\gamma(b|x,y)$ are the coefficients of the inequality which witness a non-classical d_C-dimensional behaviour $\{p(b|x,y)\}$. Additionally, since strong duality holds, the solution of the primal and dual coincides. Now, for behaviours $\{p(b|x,y)\}$ which are realisable with classical systems of dimension d_C, the visibility η respects $\eta \geq 1$, hence

$$1 + \Big[\sum_{b,x,y} \gamma(b|x,y)p(b|x,y)\Big] \geq 1\,, \tag{1.87}$$

and $\sum_{b,x,y} \gamma(b|x,y)p(b|x,y) \geq 0$, with 0 being the bound of the inequality $\{\gamma(b|x,y)\}$ for d_C-dimensional systems. Also, for behaviours $\{p(b|x,y)\}$ which are realisable with classical systems of dimension d_C, the visibility $\eta = 1$ is attainable, we have that the bound $\sum_{b,x,y} \gamma(b|x,y)p(b|x,y) = 0$ is attainable by classical systems of dimension d_C.

1.C.3. Heuristic method to find quantum probabilities without a d_C-dimensional classical simulation

A brute force method to generate quantum probabilities is simply to sample random states ρ_x and measurements $\{B_{b|y}\}$ and then using linear programming to check weather $p(b|x,y) = \mathrm{tr}\big(\rho_x\, B_{b|y}\big)$ may be simulated by d_C-dimensional classical systems. A more guided strategy may be to consider states and measurements that are rather uniformly spread. For qubits, one may choose vectors rather uniformly spread in the Bloch sphere and then construct states and projective measurements for it.

In order to find an example of a set of qubit probabilities $p(b|x,y) = \mathrm{tr}\big(\rho_x\, B_{b|y}\big)$ which makes use of $I_A = 6$ states and $I_B = 11$ projective measurements, we have chosen states and measurements corresponding to the Thomson problem [91], a family of vectors on the sphere which is defined for any number of vectors $N \in \mathbb{N}$. In our online repository [92] we provide an implementation for this heuristic method and also the exact qubit states and measurements in which $p(b|x,y) = \mathrm{tr}\big(\rho_x\, B_{b|y}\big)$ cannot be simulated by classical trits.

1.C.4. Efficient algorithm for obtaining the classical bound C_d

We now present a novel and efficient algorithm for obtaining the classical bound C_d for any given set of real numbers $\{\gamma(b|x,y)\}$. Our method is based on the scheme for finding the local bound of Bell inequalities presented in Ref. [87]. We let D_A and D_B be the set of all deterministic strategies which can be performed by Alice and Bob. Hence, by convexity, we can write:

$$C_d := \max_{\lambda} \left[\sum_{b,x,y,c} \gamma(b|x,y)D_A(c|x,\lambda)D_B(b|c,y,\lambda)\right] \tag{1.88}$$

$$= \max_{\lambda} \left[\sum_{b,y} D_B(b|c,y,\lambda) \sum_{x,c} \gamma(b|x,y)D_A(c|x,\lambda)\right] \tag{1.89}$$

$$= \max_{\lambda} \left[\sum_{b,y} \max_{b} \Big[\sum_{x,c} \gamma(b|x,y)D_A(c|x,\lambda)\Big]\right]. \tag{1.90}$$

We then see that we only need to generate Alice's deterministic strategies and to obtain the largest value of a vector, steps which can be done very efficiently.

1.C.5. Computer-assisted proof

In order to avoid numerical errors from floating point arithmetic, we show how to certify that a set of quantum probabilities given by $p(b|x,y) = \mathrm{tr}\big(\rho_x\, B_{b|y}\big)$ cannot be simulated by classical systems of dimension d_C by making use of only integers. The first step is to ensure that the probabilities $p(b|x,y) = \mathrm{tr}\big(\rho_x\, B_{b|y}\big)$ are stored with only integers or fractions, for that we will ensure that the states ρ_x and the measurements given by $B_{b|y}$ do not make use of floating point. We may adapt the Algorithm 1 of Ref. [88] to obtain a quantum state, $\rho_{\texttt{OK}}$ which is described by fractions of integers, and it is close to ρ_{float}:

Algorithm 1:

1. Construct the non-floating-point matrix ρ_{frac} by truncating the matrix ρ_{float}
2. Define the matrix $\rho := \dfrac{\rho_{\texttt{frac}} + (\rho_{\texttt{frac}})^\dagger}{2}$ to obtain a self-adjoint matrix ρ
3. Find a coefficient η such that $\rho' := \eta\rho + (1-\eta)\mathbb{1}$ is positive semidefinite
4. Output the operator $\rho_{\texttt{OK}} = \dfrac{\rho'}{\mathrm{tr}(\rho')}$.

Notice that checking whether a matrix with integers is positive semidefinite may be done efficiently by the Sylvester's criterion.

We now adapt Algorithm 2 of Ref. [88] to transform any set of floating point POVM $\{B_{b,\texttt{float}}\}_{b=1}^{O_B}$ into a POVM described by fractions of integers.

Algorithm 2:

1. Construct the non-floating-point matrices $B_{b,\text{frac}}$ by truncating the matrices $B_{b,\text{float}}$
2. For the outcomes $b \in \{1,\ldots,O_B - 1\}$, define the matrix $B_b := \dfrac{B_{b,\texttt{frac}} + (B_{b,\texttt{frac}})^\dagger}{2}$. For $b = O_B$, define $B_{O_B} := \mathbb{1} - \sum_{b=1}^{O_B-1} B_b$.
3. Find a coefficient η such that the matrices $B'_b := \eta B_b + (1-\eta)\mathbb{1}$ are positive semidefinite for every $b \in \{1,\ldots,O_B\}$.
4. Output the operator $B_{b,\texttt{OK}} = B'_b$.

Now, using a set of probabilities $\{p(b|x,y)\}$ which does not make use of floating point, we can then proceed as follows.

Algorithm 3:

1. Solve the dual problem presented in Eqs. (1.82) by standard efficient floating point linear programming methods and obtain the inequality with coefficients $\gamma_{\text{float}}(b|x, y)$.
2. Truncate $\gamma_{\text{float}}(b|x, y)$ to obtain $\gamma_{\text{frac}}(b|x, y)$.
3. Use the algorithm presented in Section 1.C.4 to obtain C_d, the classical dimension bound for the witness given by $\gamma_{\text{float}}(b|x, y)$.
4. Verify that $\sum_{b,x,y} \gamma_{\text{float}}(b|x, y) p_B(b|x, y) > C_d$.

A Matlab implementation of all code presented in this section and used in this paper is openly available at the online repository [92].

2. The minimal communication cost for simulating entangled qubits

This chapter is based on the article:

Martin J. Renner, Marco Túlio Quintino "The minimal communication cost for simulating entangled qubits," Quantum 7, 1149 (2023)
(Received: 13 July 2023, Accepted: 15 September 2023, Published: 24 October 2023)

Contributions: Marco supervised the work and introduced the problem to me. I found the protocols, carried out the proofs and calculations, and wrote most of the manuscript.

Abstract

We analyze the amount of classical communication required to reproduce the statistics of local projective measurements on a general pair of entangled qubits, $|\Psi_{AB}\rangle = \sqrt{p}\,|00\rangle + \sqrt{1-p}\,|11\rangle$ (with $1/2 \leq p \leq 1$). We construct a classical protocol that perfectly simulates local projective measurements on all entangled qubit pairs by communicating one classical trit. Additionally, when $\frac{2p(1-p)}{2p-1}\log\left(\frac{p}{1-p}\right) + 2(1-p) \leq 1$, approximately $0.835 \leq p \leq 1$, we present a classical protocol that requires only a single bit of communication. The latter model even allows a perfect classical simulation with an average communication cost that approaches zero in the limit where the degree of entanglement approaches zero ($p \to 1$). This proves that the communication cost for simulating weakly entangled qubit pairs is strictly smaller than for the maximally entangled one.

2.1. Introduction

Bell's nonlocality theorem [4] shows that quantum correlations cannot be reproduced by local hidden variables. This discovery has significantly changed our understanding of quantum theory and correlations allowed by nature. Additionally, Bell nonlocality found application in cryptography [5] and opened the possibility for protocols in which security can be certified in a device-independent way [97, 6, 98, 99].

Since quantum correlation cannot be explained by local hidden variables it is interesting to ask which additional resources are required to reproduce them. For instance, can the statistics of local measurements on two entangled qubits be simulated if the local hidden variables are augmented with some classical communication? However, since measurements

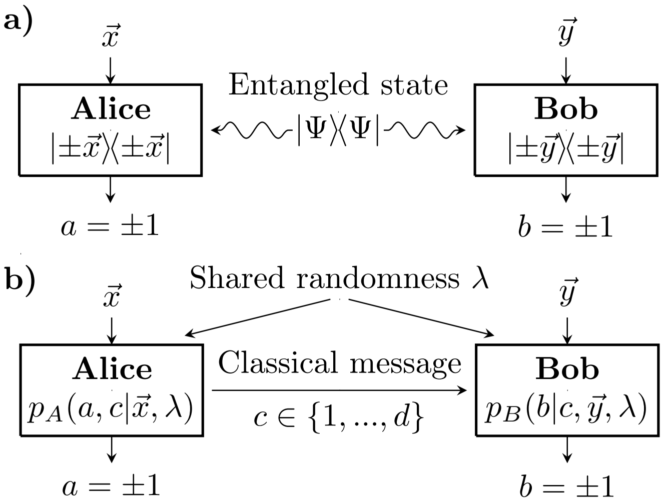

a)
$\vec{x}$
Alice
$|\pm\vec{x}\rangle\langle\pm\vec{x}|$
Entangled state
$|\Psi\rangle\langle\Psi|$
$\vec{y}$
Bob
$|\pm\vec{y}\rangle\langle\pm\vec{y}|$
$a = \pm 1$
$b = \pm 1$
b)
$\vec{x}$
Shared randomness λ
$\vec{y}$
Alice
$p_A(a, c|\vec{x}, \lambda)$
Classical message
$c \in \{1, ..., d\}$
Bob
$p_B(b|c, \vec{y}, \lambda)$
$a = \pm 1$
$b = \pm 1$

Figure 2.1.: **a)** Alice and Bob perform local projective measurements on an entangled qubit pair. **b)** Classical scenario where Alice can send a classical message to Bob.

are described by continuous parameters, one might expect that the communication cost to reproduce these quantum correlations is infinite [8]. After a sequence of improved protocols for entangled qubits [9, 51, 15, 49, 48], a breakthrough was made by Toner and Bacon in 2003 [10]. They showed that a single classical bit of communication is sufficient to simulate the statistics of all local projective measurements on a maximally entangled qubit pair. Classical communication has then been established as a natural measure of Bell nonlocality [100, 52, 53, 54, 55, 56, 101, 57, 58, 102] and found applications in constructing local hidden variable models [52].

For non-maximally entangled qubit pairs, somehow counterintuitively, all known protocols require strictly more resources. In terms of communication, the best-known result is also due to Toner and Bacon [10]. They present a protocol for non-maximally entangled qubits, which requires two bits of communication (see Ref. [103] for a two-bit protocol that considers general POVM measurements). The asymmetry of partially entangled states and other evidence suggested that simulating weakly entangled states may be harder than simulating maximally entangled ones. For instance, in Ref. [104] the authors prove that at least two uses of a PR-box are required for simulating weakly entangled qubit pairs. At the same time, a single use of a PR-box is sufficient for maximally entangled qubits [105]. Additionally, weakly entangled states are strictly more robust than maximally entangled ones when the detection loophole is considered [106, 107, 108, 109].

In this work, we present a protocol that simulates the statistics of arbitrary local projective measurements on weakly entangled qubit pairs with only a single bit of communication. Then, we construct another protocol to simulate local projective measurements on any entangled qubit pair at the cost of a classical trit.

2.2. The task and introduction of our notation

Up to local unitaries, a general entangled qubit pair can be written as

$$|\Psi_{AB}\rangle = \sqrt{p}\,|00\rangle + \sqrt{1-p}\,|11\rangle \ , \tag{2.1}$$

where $1/2 \leq p \leq 1$. At the same time, projective qubit measurements can be identified with a normalized three-dimensional real vector $\vec{x} \in \mathbb{R}^3$, the Bloch vector, $\vec{x} = (x_x, x_y, x_z)$ (with $|\vec{x}| = 1$) via the equation $|\vec{x}\rangle\langle\vec{x}| = \big(\mathbb{1} + \vec{x}\cdot\vec{\sigma}\big)/2$. Here, $\vec{\sigma} = (\sigma_X, \sigma_Y, \sigma_Z)$ are the standard Pauli matrices. In this way, we denote Alice's and Bob's measurement projectors as $|\pm\vec{x}\rangle\langle\pm\vec{x}|$ and $|\pm\vec{y}\rangle\langle\pm\vec{y}|$, which satisfy $|+\vec{x}\rangle\langle+\vec{x}| + |-\vec{x}\rangle\langle-\vec{x}| = |+\vec{y}\rangle\langle+\vec{y}| + |-\vec{y}\rangle\langle-\vec{y}| = \mathbb{1}$. According to Born's rule, when Alice and Bob apply their measurements on the entangled state $|\Psi_{AB}\rangle$, they output $a, b \in \{-1, +1\}$ according to the statistics:

$$p_Q(a, b|\vec{x}, \vec{y}) = \mathrm{Tr}[\,|a\vec{x}\rangle\langle a\vec{x}| \otimes |b\vec{y}\rangle\langle b\vec{y}| \ \ |\Psi_{AB}\rangle\langle\Psi_{AB}|]. \tag{2.2}$$

In this work, we consider the task of simulating the statistics of Eq. (2.2) with purely classical resources. More precisely, instead of Alice and Bob performing measurements on the actual quantum state, Alice prepares an output a and a message $c \in \{1, 2, ..., d\}$ that may depend on her measurement setting $\vec{x}$ and a shared classical variable λ that follows a certain probability function $\rho(\lambda)$ (see Fig. 2.1 b)). Therefore, we can denote Alice's strategy as $p_A(a, c|\vec{x}, \lambda)$. Afterwards, Alice sends the message c to Bob, who produces an outcome b depending on the message c he received from Alice, his measurement setting $\vec{y}$, and the shared variable λ. In total, we denote his strategy as $p_B(b|c, \vec{y}, \lambda)$. We want to remark that in our setting, Alice has no knowledge about Bob's measurement and vice versa. Therefore her strategy cannot depend on $\vec{y}$ and his strategy cannot depend on $\vec{x}$. Altogether, the total probability that Alice and Bob output $a, b \in \{-1, +1\}$ becomes:

$$p_C(a, b|\vec{x}, \vec{y}) = \int_\lambda \mathrm{d}\lambda\ \rho(\lambda) \sum_{c=1}^{d} p_A(a, c|\vec{x}, \lambda) p_B(b|\vec{y}, c, \lambda)\,. \tag{2.3}$$

The simulation is successful if, for any choice of projective measurements and any outcome, the classical statistics match the quantum predictions:

$$p_C(a, b|\vec{x}, \vec{y}) = p_Q(a, b|\vec{x}, \vec{y})\,. \tag{2.4}$$

We want to remark that the roles of Alice and Bob are interchangeable due to the symmetry of the state $|\Psi_{AB}\rangle$ in Eq. (2.1). Therefore, any protocol of this work can be rewritten into a protocol where Bob communicates a message (of the same length) to Alice.

For what follows, we also introduce the Heaviside function, defined by $H(z) = 1$ if $z \geq 0$ and $H(z) = 0$ if $z < 0$, as well as the related functions $\Theta(z) := H(z) \cdot z$ and the sign function $\mathrm{sgn}(z) := H(z) - H(-z)$.

2.3. Revisiting known protocols

Our methods are inspired by the best previously known protocol to simulate general entangled qubit pairs, the so-called "classical teleportation" protocol [15, 10]. To understand the idea, we first rewrite the quantum probabilities in Eq. (2.2) by using the rule of conditional probabilities $p(a,b|\vec{x},\vec{y}) = p(a|\vec{x},\vec{y}) \cdot p(b|\vec{x},\vec{y},a)$. More precisely, we denote with $p_\pm := \sum_b p(a = \pm 1, b|\vec{x},\vec{y})$ the marginal probabilities of Alice's output that read as follows:

$$p_\pm = \mathrm{Tr}[\,|\pm\vec{x}\rangle\langle\pm\vec{x}| \otimes \mathbb{1}\;\; |\Psi_{AB}\rangle\langle\Psi_{AB}|]\,. \tag{2.5}$$

Note that, due to non-signalling, the marginals $p_\pm$ do not depend on $\vec{y}$. At the same time, given Alice's outcome $a = \pm 1$, Bob's qubit collapses into a pure post-measurement state, that we denote here as:

$$|\vec{v}_\pm\rangle\langle\vec{v}_\pm| := \mathrm{Tr}_A[\,|\pm\vec{x}\rangle\langle\pm\vec{x}| \otimes \mathbb{1}\;\; |\Psi_{AB}\rangle\langle\Psi_{AB}|]/p_\pm\,. \tag{2.6}$$

If now Bob measures his qubit with the projectors $|\pm\vec{y}\rangle\langle\pm\vec{y}|$, he outputs b according to Born's rule:

$$p(b|\vec{x},\vec{y},a) = \mathrm{Tr}[|b\vec{y}\rangle\langle b\vec{y}|\,|\vec{v}_a\rangle\langle\vec{v}_a|] = |\,\langle\vec{v}_a|b\vec{y}\rangle\,|^2\,. \tag{2.7}$$

With the introduced notation, we can rewrite the quantum probabilities from Eq. (2.2) into:

$$p_Q(a,b|\vec{x},\vec{y}) = p_a \cdot |\,\langle\vec{v}_a|b\vec{y}\rangle\,|^2\,. \tag{2.8}$$

This directly implies a strategy to simulate entangled qubit pairs. Alice outputs $a = \pm 1$ according to her marginals $p_\pm$. Then, given her outcome a, she prepares a qubit in the correct post-measurement state $|\vec{v}_a\rangle\langle\vec{v}_a|$ and sends it to Bob. Finally, he measures the qubit with his projectors $|\pm\vec{y}\rangle\langle\pm\vec{y}|$.

However, in a classical simulation, Alice cannot send a physical qubit to Bob. Nevertheless, it is possible to simulate a qubit in that prepare-and-measure (PM) scenario with only two classical bits of communication [10]. In order to do so, Alice and Bob share four normalized three-dimensional vectors $\vec{\lambda}_1, \vec{\lambda}_2, \vec{\lambda}_3, \vec{\lambda}_4 \in S_2$. The first two $\vec{\lambda}_1$ and $\vec{\lambda}_2$ are uniformly and independently distributed on the sphere, whereas $\vec{\lambda}_3 = -\vec{\lambda}_1$ and $\vec{\lambda}_4 = -\vec{\lambda}_2$. From these four vectors, Alice chooses the one that maximizes $\vec{\lambda}_i \cdot \vec{v}_a$ and communicates the result to Bob. This requires a message with four different symbols ($d = 4$), hence, two bits of communication. It turns out that the distribution of the chosen vector becomes $\Theta(\vec{v}_a \cdot \vec{\lambda})/\pi$ (see Appendix 2.B for an independent proof). Finally, Bob takes the chosen vector $\vec{\lambda}$ and outputs $b = \mathrm{sgn}(\vec{y} \cdot \vec{\lambda})$. This precisely reproduces quantum correlations as specified by the following Lemma (see Appendix 2.A for a proof):

Lemma 2.1. *Bob receives a vector $\vec{\lambda} \in S_2$ distributed as $\rho(\vec{\lambda}) = \Theta(\vec{v} \cdot \vec{\lambda})/\pi$ and outputs $b = \mathrm{sgn}(\vec{y} \cdot \vec{\lambda})$. For every qubit state $\vec{v} \in S_2$ and measurement $\vec{y} \in S_2$ this reproduces quantum correlations:*

$$p(b = \pm 1|\vec{y},\vec{v}) = (1 \pm \vec{y} \cdot \vec{v})/2 = |\,\langle\pm\vec{y}|\vec{v}\rangle\,|^2\,. \tag{2.9}$$

That the distribution $\Theta(\vec{\lambda}\cdot\vec{v}_a)/\pi$ serves as a classical description of the qubit state $|\vec{v}_a\rangle\langle\vec{v}_a|$ was already observed by Kochen and Specker [110]. Later, this Kochen-Specker model was used for the task of simulating qubit correlations, see e.g. Ref. [50, 15, 52].

2.4. Our approach

The previous approach of using the prepare-and-measure scenario to simulate entangled qubits [15, 10] has a natural limitation. In fact, simulating a qubit in a PM scenario requires at least two bits of communication [103]. However, in this work, we introduce a method that supersedes such a constraint.

The goal for Alice is still to prepare the distribution $\rho(\vec{\lambda})=\Theta(\vec{v}_a\cdot\vec{\lambda})/\pi$ to Bob. The improvement here comes from the way to achieve that. In the previous approach, Alice chooses her output first (according to the probabilities $p_\pm$) and then samples the corresponding distribution $\Theta(\vec{\lambda}\cdot\vec{v}_\pm)/\pi$. In our approach, Alice samples first the weighted sum $p_+\ \Theta(\vec{\lambda}\cdot\vec{v}_+)/\pi+p_-\ \Theta(\vec{\lambda}\cdot\vec{v}_-)/\pi$ of these two distributions. Afterwards (Step 3), she chooses her output $a=\pm1$ in such a way that, conditioned on her output a, the resulting distribution of $\vec{\lambda}$ becomes exactly $\Theta(\vec{v}_a\cdot\vec{\lambda})/\pi$. At the same time, the weights $p_\pm$ ensure that Alice outputs according to the correct marginals. More formally, all our simulation protocols fit into the following general framework:

Protocol 2.0. *General framework:*

1. *Alice chooses her basis $\vec{x}$ and calculates $p_\pm,\vec{v}_\pm$.*
2. *Alice and Bob share two (or three) vectors $\vec{\lambda}_i\in S_2$ according to a certain distribution (specified later). Alice informs Bob to choose one of these vectors such that the resulting distribution of the chosen vector $\vec{\lambda}$ becomes:*

$$\rho_{\vec{x}}(\vec{\lambda}):=p_+\ \Theta(\vec{v}_+\cdot\vec{\lambda})/\pi+p_-\ \Theta(\vec{v}_-\cdot\vec{\lambda})/\pi\,. \tag{2.10}$$

3. *Given that $\vec{\lambda}$, Alice outputs $a=\pm1$ with probability*

$$p_A(a|\vec{x},\vec{\lambda})=\frac{p_a\ \Theta(\vec{\lambda}\cdot\vec{v}_a)/\pi}{\rho_{\vec{x}}(\vec{\lambda})}\,. \tag{2.11}$$

4. *Bob chooses his basis $\vec{y}$ and outputs $b=\operatorname{sgn}(\vec{y}\cdot\vec{\lambda})$.*

Proof. To see that this is sufficient to simulate the correct statistics, we first calculate the total probability that Alice outputs $a=\pm1$ in Step 3:

$$p_A(a|\vec{x})=\int_{S_2}p_A(a|\vec{x},\vec{\lambda})\cdot\rho_{\vec{x}}(\vec{\lambda})\,\mathrm{d}\vec{\lambda}=\int_{S_2}p_a\ \Theta(\vec{\lambda}\cdot\vec{v}_\pm)/\pi\,\mathrm{d}\vec{\lambda}=p_a\,. \tag{2.12}$$

For the last step, see Eq. (2.27) in App. 2.A. Now we can show that, given Alice outputs $a=\pm1$, the conditional distribution of the resulting vector $\vec{\lambda}$ is:

$$\rho_{\vec{x}}(\vec{\lambda}|a)=\frac{p_A(a|\vec{x},\vec{\lambda})\cdot\rho_{\vec{x}}(\vec{\lambda})}{p_A(a|\vec{x})}=\frac{1}{\pi}\ \Theta(\vec{\lambda}\cdot\vec{v}_a)\,. \tag{2.13}$$

As in the previous approach, Lemma 2.1 ensures that Bob outputs b in Step 4 according to $p(b|\vec{x},\vec{y},a) = |\langle\vec{v}_a|b\vec{y}\rangle|^2$. All together, the total probability of this procedure becomes $p_C(a,b|\vec{x},\vec{y}) = p_a \cdot p(b|\vec{x},\vec{y},a) = p_a \cdot |\langle\vec{v}_a|b\vec{y}\rangle|^2$. This equals $p_Q(a,b|\vec{x},\vec{y})$ as given in Eq. (2.8). □

Hence, the amount of communication to simulate a qubit pair reduces to an efficient way to sample the distributions $\rho_{\vec{x}}(\vec{\lambda})$. Clearly, the ability to sample each term $\Theta(\vec{\lambda}\cdot\vec{v}_\pm)/\pi$ individually (as in the previous approach [15, 10]) implies the possibility to sample the weighted sum of these two terms $\rho_{\vec{x}}(\vec{\lambda})$. However, in general, this is not necessary, and we find more efficient ways to do that. The improvement comes from the fact, that the two post-measurement states are not independent of each other but satisfy the following relation:

$$p_+\,|\vec{v}_+\rangle\langle\vec{v}_+| + p_-\,|\vec{v}_-\rangle\langle\vec{v}_-| = \mathrm{Tr}_A[\,|\Psi_{AB}\rangle\langle\Psi_{AB}|]\,. \tag{2.14}$$

This follows directly from Eq. (2.6) and $|{+\vec{x}}\rangle\langle{+\vec{x}}| + |{-\vec{x}}\rangle\langle{-\vec{x}}| = \mathbb{1}$. In the Bloch vector representation, this equation becomes:

$$p_+\ \vec{v}_+ + p_-\ \vec{v}_- = (2p-1)\ \vec{z}\,, \tag{2.15}$$

where we define $\vec{z} := (0,0,1)^T$. For instance, if the state is local ($p = 1$), the two post-measurement states are always $\vec{v}_\pm = \vec{z}$, independent of Alice's measurement $\vec{x}$. In that case, the distributions $\rho_{\vec{x}}(\vec{\lambda}) \equiv \Theta(\vec{\lambda}\cdot\vec{z})/\pi$ in Eq. (2.10) are constant and do not require any communication to be implemented. If the state is weakly entangled ($p \lesssim 1$), one post-measurement state $\vec{v}_a$ is still very close to the vector $\vec{z}$. In this way, it turns out that, for every $\vec{x}$, the distribution $\rho_{\vec{x}}(\vec{\lambda})$ is dominated by a constant part proportional to $\Theta(\vec{\lambda}\cdot\vec{z})/\pi$. More formally, we can define

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) :=\rho_{\vec{x}}(\vec{\lambda}) - \frac{(2p-1)}{\pi}\ \Theta(\vec{\lambda}\cdot\vec{z})\,. \tag{2.16}$$

In Appendix 2.C, we prove the following properties of that distribution and give an illustration of them (see Fig. 2.3).

Lemma 2.2. *The distribution $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ defined above is positive, $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \geq 0$ and subnormalized, $\int_{S_2} \tilde{\rho}_{\vec{x}}(\vec{\lambda})\ \mathrm{d}\vec{\lambda} = 2(1-p)$. Additionally, it respects the two upper bounds, $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{\sqrt{p(1-p)}}{\pi}$ and $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{p_\pm}{\pi}|\vec{\lambda}\cdot\vec{v}_\pm|$.*

2.5. One bit protocol for weakly entangled states

In particular, when the state is weakly entangled, the extra term $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ remains small. This allows us to find the following protocol for the range $p \geq 1/2 + \sqrt{3}/4 \approx 0.933$.

Protocol 2.1 ($1/2 + \sqrt{3}/4 \leq p \leq 1$, 1 bit)**.** *Same as Protocol 2.0 with the following 2. Step:*
Alice and Bob share two normalized three-dimensional vectors $\vec{\lambda}_1, \vec{\lambda}_2 \in S_2$ according to the distribution:

$$\rho(\vec{\lambda}_1) = \frac{1}{4\pi}\,, \qquad\qquad \rho(\vec{\lambda}_2) = \frac{1}{\pi}\,\Theta(\vec{\lambda}_2 \cdot \vec{z})\,. \tag{2.17}$$

Alice sets $c = 1$ with probability:

$$p_A(c = 1|\vec{x}, \vec{\lambda}_1) = (4\pi) \cdot \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1) \tag{2.18}$$

and otherwise she sets $c = 2$. She communicates the bit c to Bob. Both set $\vec{\lambda} := \vec{\lambda}_c$ and reject the other vector.

Proof. Whenever Alice chooses the first vector, the resulting distribution of the chosen vector is precisely $p_A(c = 1|\vec{x}, \vec{\lambda}_1) \cdot \rho(\vec{\lambda}_1) = \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1)$. The total probability that she chooses the first vector is $\int_{S_2} p_A(c = 1|\vec{x}, \vec{\lambda}_1) \cdot \rho(\vec{\lambda}_1)\, \mathrm{d}\vec{\lambda}_1 = \int_{S_2} \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1)\, \mathrm{d}\vec{\lambda}_1 = 2(1-p)$ (see Lemma 2.2). In all the remaining cases, with total probability $2p - 1$, she chooses vector $\vec{\lambda}_2$, distributed as $\Theta(\vec{\lambda}_2 \cdot \vec{z})/\pi$. Therefore, the total distribution of the chosen vector $\vec{\lambda} := \vec{\lambda}_c$ becomes the desired distribution

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) + \frac{(2p-1)}{\pi}\,\Theta(\vec{\lambda} \cdot \vec{z}) = \rho_{\vec{x}}(\vec{\lambda})\,. \tag{2.19}$$

In order for the protocol to be well defined, it has to hold that $0 \leq p(c = 1|\vec{x}, \vec{\lambda}_1) \leq 1$, hence $0 \leq \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1) \leq 1/(4\pi)$. As a consequence of Lemma 2.2, this is true whenever $1/2 + \sqrt{3}/4 \leq p \leq 1$. □

Clearly, the simulation of weakly entangled states requires some communication since all pure entangled quantum states violate a Bell inequality [16]. It is surprising that the minimal amount of information (1 bit) is already sufficient to reproduce the correlations for all projective measurements. However, we can even improve that protocol. In Appendix 2.D (Protocol 2.5), we show how to simulate every weakly entangled state with $0.835 \leq p \leq 1$ by communicating only a single bit. Moreover, it turns out that this bit is not necessary in each round. In fact, Alice sends the bit in only a ratio of $N(p)$ of the rounds, where

$$N(p) := \frac{2p(1-p)}{2p-1} \log\left(\frac{p}{1-p}\right) + 2(1-p)\,. \tag{2.20}$$

In the remaining rounds, with probability $1 - N(p)$, they do not communicate with each other. In the limit where p approaches one, the function $N(p)$ approaches zero. Hence,

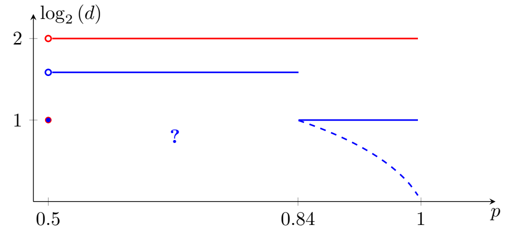


Figure 2.2.: Length of the classical message d required to simulate a qubit pair $|\Psi_{AB}\rangle = \sqrt{p}\,|00\rangle + \sqrt{1-p}\,|11\rangle$ as a function in p. The previous best result, from Toner and Bacon [10], is presented in red. Our novel results are presented in blue. The dashed curve in blue represents the fraction of rounds where Alice needs to send a bit to Bob. The main open question is whether a single bit is sufficient for simulating qubit pairs with $1/2 < p < 0.84$.

if the state is very weakly entangled, a perfect simulation is possible even though they communicate a single bit only in a small fraction of rounds (see dashed curve in Fig. 2.2). It is known that a simulation of a maximally entangled state without communication in some fraction of rounds is impossible. This would contradict the fact that the singlet has no local part [93, 94]. Hence, our result shows that simulating weakly entangled states requires strictly fewer communication resources than simulating a maximally entangled one. Interestingly, we can also use our approach to quantify the local content of any pure entangled two-qubit state (see Appendix 2.E). More precisely, we maximize the fraction of rounds in which no communication is necessary. This provides an independent proof of a result by Portmann et al. [111].

2.6. Trit protocol for arbitrary entangled pairs

It is worth mentioning, that we also recover a one-bit protocol for simulating the maximally entangled state ($p = 1/2$) in our framework. In that case, there is another geometric argument that allows to sample the distributions $\rho_{\vec{x}}(\vec{\lambda})$ efficiently. More precisely, the two post-measurement states are always opposite of each other $\vec{v}_- = -\vec{v}_+$ and it holds that $p_+ = p_- = 1/2$. In this way, the distribution $\rho_{\vec{x}}(\vec{\lambda})$ turns out to be $\rho_{\vec{x}}(\vec{\lambda}) = |\vec{\lambda}\cdot\vec{v}_+|/(2\pi)$. It was already observed, by Degorre et al. [52], that this distribution can be sampled by communicating only a single bit of communication (see Appendix 2.B for details and an independent proof). Here, we connect this with the techniques developed for Protocol 2.1 to present a protocol that simulates all entangled qubit pairs by communicating a classical trit.

Protocol 2.2 ($1/2 \leq p \leq 1$, 1 trit)**.** *Same as Protocol 2.0 with the following 2. Step: Alice and Bob share three normalized three-dimensional vectors $\vec{\lambda}_1, \vec{\lambda}_2, \vec{\lambda}_3 \in S_2$ according to the following distribution:*

$$\rho(\vec{\lambda}_1) = \frac{1}{4\pi}\,, \qquad \rho(\vec{\lambda}_2) = \frac{1}{4\pi}\,, \qquad \rho(\vec{\lambda}_3) = \frac{1}{\pi}\,\Theta(\vec{\lambda}_3\cdot\vec{z})\,. \tag{2.21}$$

If $p_+ \leq 0.5$ she sets $\vec{v} := \vec{v}_+$, otherwise she sets $\vec{v} := \vec{v}_-$. Afterwards, she sets $c = 1$ if $|\vec{v} \cdot \vec{\lambda}_1| \geq |\vec{v} \cdot \vec{\lambda}_2|$ and $c = 2$ otherwise. Finally, with probability

$$p_A(t = c|\vec{x}, \vec{\lambda}_c) = \frac{\tilde{\rho}_{\vec{x}}(\vec{\lambda}_c)}{\frac{1}{2\pi}|\vec{\lambda}_c \cdot \vec{v}|} \tag{2.22}$$

she sets $t = c$ and otherwise, she sets $t = 3$. She communicates the trit t to Bob. Both set $\vec{\lambda} := \vec{\lambda}_t$ and reject the other two vectors.

Proof. We show that the distribution of the shared vector $\vec{\lambda}$ becomes exactly the required $\rho_{\vec{x}}(\vec{\lambda})$. Consider the step before Alice sets $t = c$ or $t = 3$. As a result of Ref. [52] (see Protocol 2.3 in Appendix 2.B for details), the distribution of the vector $\vec{\lambda}_c$ is $\rho(\vec{\lambda}_c) = \frac{1}{2\pi}|\vec{\lambda}_c \cdot \vec{v}|$. Now we use a similar idea as in the protocol for weakly entangled states. Whenever she sets $t = c$, the resulting distribution of the chosen vector is precisely $p_A(t = c|\vec{x}, \vec{\lambda}_c) \cdot \rho(\vec{\lambda}_c) = \tilde{\rho}_{\vec{x}}(\vec{\lambda}_c)$. The total probability that she sets $t = c$ is $\int_{S_2} p_A(t = c|\vec{x}, \vec{\lambda}_c) \cdot \rho(\vec{\lambda}_c) \,\mathrm{d}\vec{\lambda}_c = \int_{S_2} \tilde{\rho}_{\vec{x}}(\vec{\lambda}_c) \,\mathrm{d}\vec{\lambda}_c = 2(1-p)$ (see Lemma 2.2). In all the remaining cases, with total probability $2p - 1$, she chooses vector $\vec{\lambda}_3$, distributed as $\Theta(\vec{\lambda}_3 \cdot \vec{z})/\pi$. Therefore, the total distribution of the chosen vector $\vec{\lambda} := \vec{\lambda}_t$ becomes the desired distribution

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) + \frac{(2p-1)}{\pi}\, \Theta(\vec{\lambda} \cdot \vec{z}) = \rho_{\vec{x}}(\vec{\lambda})\,. \tag{2.23}$$

The fact that $0 \leq p_A(t = c|\vec{x}, \vec{\lambda}_c) \leq 1$ follows from $0 \leq \tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{p_\pm}{\pi}|\vec{\lambda} \cdot \vec{v}_\pm|$ in Lemma 2.2. □

2.7. Conclusion

To conclude, we showed that a classical trit is enough for simulating the outcomes of local projective measurements on any entangled qubit pair. For weakly entangled states, we proved that already a single bit is sufficient. In the latter case, Alice does not need to send the bit in all the rounds, which is impossible for a maximally entangled state [93, 94]. In this way, we show that simulating weakly entangled states is strictly simpler than simulating maximally entangled ones.

The main open question now is whether a single bit is sufficient to simulate every entangled qubit pair, see Fig. 2.2. Recently, numerical evidence has been reported by Sidajaya et al. [112] that a single bit is indeed enough. However, an analytical model is still missing. We remark that our framework is in principle capable of providing such a model. The challenge becomes to find, for each qubit pair, a distribution of two shared random vectors, such that Alice can sample $\rho_{\vec{x}}(\vec{\lambda})$ for every measurement basis $\vec{x}$. In all protocols considered here, the shared vectors are independent of each other, i.e., $\rho(\vec{\lambda}_1, \vec{\lambda}_2) = \rho(\vec{\lambda}_1) \cdot \rho(\vec{\lambda}_2)$. Dropping this constraint may be a way to extend our approach and may lead to a complete solution to this longstanding open question [113, 37, 77].

Acknowledgments

We thank Časlav Brukner, Valerio Scarani, Peter Sidajaya, Armin Tavakoli, Isadora Veeren, and Bai Chu Yu for fruitful discussions. Furthermore, we thank Nicolas Brunner and Nicolas Gisin for pointing out Ref. [111] to us. This research was funded in whole, or in part, by the Austrian Science Fund (FWF) through BeyondC (F7103). For the purpose of open access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801110. It reflects only the authors' view, the EU Agency is not responsible for any use that may be made of the information it contains. ESQ has received funding from the Austrian Federal Ministry of Education, Science and Research (BMBWF).

2.A. Proof of Lemma 2.1

Lemma 2.1. *Bob receives a vector* $\vec{\lambda} \in S_2$ *distributed as* $\rho(\vec{\lambda}) = \Theta(\vec{v} \cdot \vec{\lambda})/\pi$ *and outputs* $b = \operatorname{sgn}(\vec{y} \cdot \vec{\lambda})$. *For every qubit state* $\vec{v} \in S_2$ *and measurement* $\vec{y} \in S_2$ *this reproduces quantum correlations:*

$$p(b = \pm 1|\vec{y}, \vec{v}) = (1 \pm \vec{y} \cdot \vec{v})/2 = |\langle \pm \vec{y}|\vec{v}\rangle|^2 . \tag{2.24}$$

Proof. Bob outputs $b = +1$ if and only if $\vec{y} \cdot \vec{\lambda} \geq 0$. Therefore, the total probability that Bob outputs $b = +1$ becomes:

$$p(b = +1|\vec{y}, \vec{v}) = \int_{S_2} H(\vec{y} \cdot \vec{\lambda}) \cdot \ \rho(\vec{\lambda}) \ \mathrm{d}\vec{\lambda} = \frac{1}{\pi} \int_{S_2} H(\vec{y} \cdot \vec{\lambda}) \cdot \ \Theta(\vec{v} \cdot \vec{\lambda}) \ \mathrm{d}\vec{\lambda} . \tag{2.25}$$

Here, $H(z)$ is the Heaviside function ($H(z) = 1$ if $z \geq 0$ and $H(z) = 0$ if $z < 0$) and $\Theta(z) := H(z) \cdot z$. Exactly the same integral is evaluated in Lemma 1.1 (and in similar forms also in Ref. [50, 15, 52]). Therefore, we obtain:

$$\frac{1}{\pi} \int_{S_2} H(\vec{y} \cdot \vec{\lambda}) \cdot \ \Theta(\vec{v} \cdot \vec{\lambda}) \ \mathrm{d}\vec{\lambda} = \frac{1}{2}(1 + \vec{y} \cdot \vec{v}) . \tag{2.26}$$

Hence, $p(b = +1|\vec{y}, \vec{v}) = (1 + \vec{y} \cdot \vec{v})/2$. Clearly, $p(b = -1|\vec{y}, \vec{v}) = 1 - p(b = +1|\vec{y}, \vec{v}) = (1 - \vec{y} \cdot \vec{v})/2$. □

It appears several times in this work that $\int_{S_2} \Theta(\vec{\lambda} \cdot \vec{v})/\pi \ \mathrm{d}\vec{\lambda} = 1$ for every normalized vector $\vec{v} \in S_2$. The proof follows by a similar calculation as in the above Lemma:

$$\frac{1}{\pi} \int_{S_2} \Theta(\vec{\lambda} \cdot \vec{v}) \ \mathrm{d}\vec{\lambda} = \frac{1}{\pi} \int_{S_2} H(\vec{\lambda} \cdot \vec{v}) \cdot \Theta(\vec{\lambda} \cdot \vec{v}) \ \mathrm{d}\vec{\lambda} = \frac{1}{2}(1 + \vec{v} \cdot \vec{v}) = 1 . \tag{2.27}$$

The introduction of the Heaviside function in the second step clearly does not change the integral since $H(\vec{\lambda} \cdot \vec{v})$ has the same support as $\Theta(\vec{\lambda} \cdot \vec{v})$ or, more formally, $\Theta(\vec{\lambda} \cdot \vec{v}) := H(\vec{\lambda} \cdot \vec{v}) \cdot (\vec{\lambda} \cdot \vec{v}) = H(\vec{\lambda} \cdot \vec{v})^2 \cdot (\vec{\lambda} \cdot \vec{v}) = H(\vec{\lambda} \cdot \vec{v}) \cdot \Theta(\vec{\lambda} \cdot \vec{v})$, where we used that $H(z) = H(z)^2$ for every $z \in \mathbb{R}$.

2.B. Protocol for the maximally entangled qubit pair

As mentioned in the main text, in the case of a maximally entangled state ($p = 1/2$), there is a similar geometric argument that allows sampling the distributions $\rho_{\vec{x}}(\vec{\lambda})$ efficiently. More precisely, the two post-measurement states are always opposite of each other $\vec{v}_- = -\vec{v}_+$ and it holds that $p_+ = p_- = 1/2$. Therefore, the distribution $\rho_{\vec{x}}(\vec{\lambda})$ has, for every choice of Alice's measurement $\vec{x}$, the form (note that $\Theta(z) + \Theta(-z) = |z|$ for every $z \in \mathbb{R}$)

$$\rho_{\vec{x}}(\vec{\lambda}) = \frac{1}{2\pi} \Big(\Theta(\vec{\lambda} \cdot \vec{v}_+) + \ \Theta(\vec{\lambda} \cdot (-\vec{v}_+)) \Big) = \frac{1}{2\pi} |\vec{\lambda} \cdot \vec{v}_+| . \tag{2.28}$$

It was already observed by Degorre et al. [52] ("Theorem 6 (The "choice" method)"), that this distribution can be sampled by communicating only a single bit. This leads directly to a simulation of the maximally entangled state with one bit of communication. This is exactly the version of Degorre et al. [52] for the protocol of Toner and Bacon for the singlet (see also "Theorem 10 (Communication)" of Ref. [52]):

Protocol 2.3 ($p = 1/2$, 1 bit, from Ref. [52])**.** *Same as Protocol 2.0 with the following 2. Step:*
Alice and Bob share two normalized three-dimensional vectors $\vec{\lambda}_1, \vec{\lambda}_2 \in S_2$ that are independent and uniformly distributed on the unit sphere, $\rho(\vec{\lambda}_1) = \rho(\vec{\lambda}_2) = 1/4\pi$. Alice sets $c = 1$ if $|\vec{v}_+ \cdot \vec{\lambda}_1| \geq |\vec{v}_+ \cdot \vec{\lambda}_2|$ and $c = 2$ otherwise. She communicates the bit c to Bob and both set $\vec{\lambda} := \vec{\lambda}_c$.

Proof. The proof can be found in Ref. [52] ("Theorem 6 (The "choice" method)"). However, we want to give an independent proof here. We can focus first on the case where $\vec{v}_+ = \vec{z}$. All other cases are analogous due to the spherical symmetry of the problem. In that case, we can write $\vec{\lambda}_1$ and $\vec{\lambda}_2$ in spherical coordinates, $\vec{\lambda}_i = (\sin\theta_i \cdot \cos\phi_i, \sin\theta_i \cdot \sin\phi_i, \cos\theta_i)$. In this notation, Alice picks $\vec{\lambda}_1$ if and only if $|\vec{\lambda}_1 \cdot \vec{z}| = |\cos\theta_1| \geq |\vec{\lambda}_2 \cdot \vec{z}| = |\cos\theta_2|$. For a given $\vec{\lambda}_1$, this happens with probability

$$\int_{S_2} H(|\vec{\lambda}_1 \cdot \vec{z}| - |\vec{\lambda}_2 \cdot \vec{z}|) \cdot \rho(\vec{\lambda}_2)\ \mathrm{d}\vec{\lambda}_2 = \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} H(|\cos\theta_1| - |\cos\theta_2|) \cdot \sin\theta_2\ \mathrm{d}\theta_2\ \mathrm{d}\phi_2 \tag{2.29}$$

$$= \frac{1}{2} \int_0^{\pi} H(|\cos\theta_1| - |\cos\theta_2|) \cdot \sin\theta_2\ \mathrm{d}\theta_2 \tag{2.30}$$

If $0 \leq \theta_1 \leq \pi/2$ and hence $\cos\theta_1 \geq 0$, the region where $H(|\cos\theta_1| - |\cos\theta_2|) = 1$ becomes exactly $\theta_1 \leq \theta_2 \leq \pi - \theta_1$ and hence the above integral becomes:

$$\frac{1}{2} \int_{\theta_1}^{\pi-\theta_1} \sin\theta_2\ \mathrm{d}\theta_2 = \frac{1}{2}\ [-\cos\theta_2]_{\theta_1}^{\pi-\theta_1} = (-\cos(\pi - \theta_1) + \cos(\theta_1))/2 = \cos\theta_1 = |\vec{z} \cdot \vec{\lambda}_1|\,. \tag{2.31}$$

For $\pi/2 \leq \theta_1 \leq \pi$ and hence $\cos\theta_1 \leq 0$ a similar calculation leads to $-\cos\theta_1 = |\cos\theta_1| = |\vec{z} \cdot \vec{\lambda}_1|$. (One can also observe that the integral depends only on $|\cos\theta_1|$, which leads to the same statement.) Hence, whenever Alice chooses $\vec{\lambda} := \vec{\lambda}_1$ the distribution of that vector becomes $\rho(\vec{\lambda}_1) \cdot |\vec{z} \cdot \vec{\lambda}_1| = |\vec{z} \cdot \vec{\lambda}_1|/(4\pi)$. Analogously, whenever Alice chooses $\vec{\lambda} := \vec{\lambda}_2$ the distribution of that chosen vector is again $|\vec{z} \cdot \vec{\lambda}_2|/(4\pi)$, due to the symmetric roles of $\vec{\lambda}_1$ and $\vec{\lambda}_2$. Hence, the distribution of the chosen vector $\vec{\lambda}$ becomes, in total, the sum of these two terms $\rho(\vec{\lambda}) = |\vec{z} \cdot \vec{\lambda}|/(2\pi)$. For a general vector $\vec{v}_+$, the analog expression $\rho(\vec{\lambda}) = |\vec{v}_+ \cdot \vec{\lambda}|/(2\pi)$ holds, because of the spherical symmetry of the protocol. □

We also want to remark here, that in the case of a maximally entangled state, Alice's response function in the third step Eq. (2.11) can be, due to Eq. (2.28), rewritten into $p_A(a = \pm 1|\vec{x}, \vec{\lambda}) = H(\vec{\lambda} \cdot \vec{v}_+)$, or, equivalently, $a = \mathrm{sgn}(\vec{\lambda} \cdot \vec{v}_+)$.

2.B.1. "Classical teleportation" protocol

With this observation, we can also understand the classical teleportation protocol from the main text. To avoid confusion, this protocol is *not* of the form given in Protocol 2.0:

Protocol 2.4. *The following protocol simulates a qubit in a prepare-and-measure scenario:*

1. *Alice chooses the quantum state $|v\rangle\langle v| = (\mathbb{1} + \vec{v}\cdot\vec{\sigma})/2$ she wants to send to Bob.*

2. *Alice and Bob share two normalized three-dimensional vectors $\vec{\lambda}_1, \vec{\lambda}_2 \in S_2$ that are independent and uniformly distributed on the unit sphere, $\rho(\vec{\lambda}_1) = \rho(\vec{\lambda}_2) = 1/4\pi$. Alice sets $c_1 = 1$ if $|\vec{v}\cdot\vec{\lambda}_1| \geq |\vec{v}\cdot\vec{\lambda}_2|$ and $c_1 = 2$ otherwise. In addition, Alice defines a second bit $c_2 = \mathrm{sgn}(\vec{\lambda}_{c_1}\cdot\vec{v})$. She communicates the two bits c_1 and c_2 to Bob and both set $\vec{\lambda} := c_2\,\vec{\lambda}_{c_1}$.*

3. *Bob outputs $b = \mathrm{sgn}(\vec{y}\cdot\vec{\lambda})$.*

Proof. As a result of the above (Protocol 2.3), the distribution of the vector $\vec{\lambda}_{c_1}$ is $\rho(\vec{\lambda}_{c_1}) = |\vec{\lambda}_{c_1}\cdot\vec{v}|/(2\pi)$. When he defines $\vec{\lambda} := c_2\,\vec{\lambda}_{c_1}$, he exactly flips the vector $\vec{\lambda}_{c_1}$ if and only if $\vec{\lambda}_{c_1}\cdot\vec{v} < 0$. With the additional flip, he obtains the distribution:

$$\frac{1}{2\pi}|\vec{\lambda}_{c_1}\cdot\vec{v}| = \frac{1}{2\pi}\left(\Theta(\vec{\lambda}_{c_1}\cdot\vec{v}) + \ \Theta(\vec{\lambda}_{c_1}\cdot(-\vec{v}))\right) \tag{2.32}$$

$$\xrightarrow{flip} \frac{1}{2\pi}\left(\Theta(\vec{\lambda}\cdot\vec{v}) + \ \Theta((-\vec{\lambda})\cdot(-\vec{v}))\right) = \frac{1}{\pi}\Theta(\vec{\lambda}\cdot\vec{v})\,. \tag{2.33}$$

Therefore, the distribution of the vector $\vec{\lambda}$ becomes $\Theta(\vec{v}\cdot\vec{\lambda})/\pi$. In this way, Alice managed to send exactly a classical description of the state $|\vec{v}\rangle\langle\vec{v}|$ to Bob. More precisely, Lemma 2.1 ensures that Bob outputs according to $p(b = \pm 1|\vec{v},\vec{y}) = (1 \pm \vec{y}\cdot\vec{v})/2$, as required by quantum mechanics. □

Note that, in the main text, the second step is formulated as follows: "Alice and Bob share four normalized three-dimensional vectors $\vec{\lambda}_1, \vec{\lambda}_2, \vec{\lambda}_3, \vec{\lambda}_4 \in S_2$. The first two $\vec{\lambda}_1$ and $\vec{\lambda}_2$ are uniformly and independently distributed on the sphere, whereas $\vec{\lambda}_3 = -\vec{\lambda}_1$ and $\vec{\lambda}_4 = -\vec{\lambda}_2$. From these four vectors, Alice chooses the one that maximizes $\vec{\lambda}_i\cdot\vec{v}$ and communicates the result to Bob and both set $\vec{\lambda} := \vec{\lambda}_i$."

This is just a reformulation of the second step in Protocol 2.4 and both versions are equivalent. To see this, fix $\vec{\lambda}_1$ and $\vec{\lambda}_2$. If $|\vec{v}\cdot\vec{\lambda}_1| \geq |\vec{v}\cdot\vec{\lambda}_2|$ and $\vec{v}\cdot\vec{\lambda}_1 \geq 0$, Alice will send $c_1 = 1$ and $c_2 = +1$ and both set $\vec{\lambda} := c_2\,\vec{\lambda}_{c_1} = \vec{\lambda}_1$ in step two of Protocol 2.4. In the reformulation, it turns out that the vector that maximizes $\vec{v}\cdot\vec{\lambda}_i$ is precisely $\vec{\lambda}_1$ since $|\vec{v}\cdot\vec{\lambda}_1| \geq |\vec{v}\cdot\vec{\lambda}_2|$ and $\vec{v}\cdot\vec{\lambda}_1 \geq 0$ imply that $\vec{v}\cdot\vec{\lambda}_1 \geq \vec{v}\cdot\vec{\lambda}_2$; $\vec{v}\cdot\vec{\lambda}_1 \geq \vec{v}\cdot\vec{\lambda}_3 = -\vec{v}\cdot\vec{\lambda}_1$ as well as $\vec{v}\cdot\vec{\lambda}_1 \geq \vec{v}\cdot\vec{\lambda}_4 = -\vec{v}\cdot\vec{\lambda}_2$. With similar arguments, one can check that, for a fixed $\vec{\lambda}_1$ and $\vec{\lambda}_2$, they always choose the same vector $\vec{\lambda}$ in both versions.

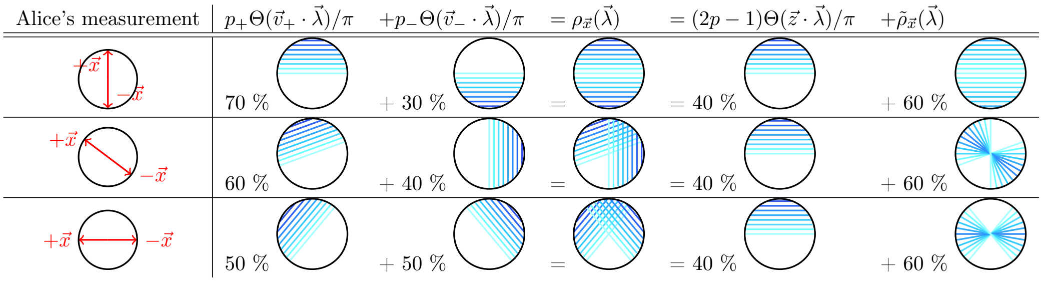

Alice's measurement
$p_+\Theta(\vec{v}_+\cdot\vec{\lambda})/\pi$
$+p_-\Theta(\vec{v}_-\cdot\vec{\lambda})/\pi$
$=\rho_{\vec{x}}(\vec{\lambda})$
$=(2p-1)\Theta(\vec{z}\cdot\vec{\lambda})/\pi$
$+\tilde{\rho}_{\vec{x}}(\vec{\lambda})$
$+\vec{x}$
$-\vec{x}$
70 %
+ 30 %
=
= 40 %
+ 60 %
$+\vec{x}$
$-\vec{x}$
60 %
+ 40 %
=
= 40 %
+ 60 %
$+\vec{x}$
$-\vec{x}$
50 %
+ 50 %
=
= 40 %
+ 60 %

Figure 2.3.: A sketch of the relevant distributions for the state $|\Psi_{AB}\rangle = \sqrt{0.7}\,|00\rangle + \sqrt{0.3}\,|11\rangle$: In the previous approach, Alice tosses a biased coin according to the marginals of her measurement. Then she uses the classical teleportation protocol to create on Bob's side a vector according to the distribution $\Theta(\vec{v}_a\cdot\vec{\lambda})/\pi$, from which Bob can reproduce quantum statistics (see Lemma 2.1). However, it turns out that it is enough to sample only the sum of these two distributions $\rho_{\vec{x}}(\vec{\lambda})$. Since all of these distributions (for every $\vec{x}$) can be rewritten into a constant part and the extra term $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$, one can sample all of these distributions by communicating only a classical trit. If the state is very weakly entangled ($p \to 1$), it turns out that the constant part with weight $2p-1$ dominates and already one bit is sufficient to sample all the distributions $\rho_{\vec{x}}(\vec{\lambda})$.

2.C. Properties of $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$

In this section, we prove several properties of the distribution $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ that are crucial for our protocols. Let us recall that,

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) := \frac{1}{\pi}\Big(p_+\ \Theta(\vec{\lambda}\cdot\vec{v}_+) + p_-\ \Theta(\vec{\lambda}\cdot\vec{v}_-) - (2p-1)\ \Theta(\vec{\lambda}\cdot\vec{z})\Big) \tag{2.34}$$

where, $0 \le p_\pm \le 1$, $p_+ + p_- = 1$ and $p \ge 0.5$ (therefore $0 \le (2p-1) \le 1$) and all vectors are normalized vectors on the Bloch sphere $\vec{\lambda}, \vec{v}_+, \vec{v}_-, \vec{z} \in S_2$. The only relevant equation that we need for the proof is Eq. (2.15), which reads as

$$p_+\ \vec{v}_+ + p_-\ \vec{v}_- = (2p-1)\ \vec{z}, \tag{2.35}$$

and the definition of $\Theta(z)$:

$$\Theta(z) := \begin{cases} z & \text{if } z \ge 0 \\ 0 & \text{if } z < 0 \end{cases}. \tag{2.36}$$

Lemma 2.2. *The Distribution $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ defined above satisfies the following properties:*

(i) positive: $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \ge 0$

(ii) symmetric: $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) = \tilde{\rho}_{\vec{x}}(-\vec{\lambda})$

(iii) area: $\int_{S_2} \tilde{\rho}_{\vec{x}}(\vec{\lambda})\ \mathrm{d}\vec{\lambda} = 2(1-p)$

(iv) 1st bound: $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{p_\pm}{\pi} |\vec{\lambda} \cdot \vec{v}_\pm|$

(v) 2nd bound: $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{1}{2\pi} \frac{1-C^2}{\sqrt{1-C^2 \sin^2(\theta)+C|\cos(\theta)|}}$ *with* $C := 2p-1$ *and* $\cos(\theta) = \vec{\lambda} \cdot \vec{z}$

(vi) 3rd bound: $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{\sqrt{p(1-p)}}{\pi}$

Proof. Most of these properties follow directly from the fact that the function $\Theta(a)$ is convex and satisfies:

$$\forall a, b \in \mathbb{R}: \ \Theta(a) + \Theta(b) \geq \Theta(a+b)\,. \tag{2.37}$$

Furthermore, we use the following property frequently:

$$\forall a, b \in \mathbb{R} \text{ with } a \geq 0: \ \Theta(a \cdot b) = a \cdot \Theta(b)\,. \tag{2.38}$$

Furthermore, $\Theta(a) + \Theta(-a) = |a|$ as well as $\Theta(a) - \Theta(-a) = a$ for all $a \in \mathbb{R}$. All of these properties follow directly from the definition of $\Theta(a)$.

(i) positive:
We can use $\Theta(a) + \Theta(b) \geq \Theta(a+b)$ with $a = p_+\ \vec{\lambda} \cdot \vec{v}_+$ and $b = p_-\ \vec{\lambda} \cdot \vec{v}_-$. As a consequence of $p_+\ \vec{v}_+ + p_-\ \vec{v}_- = (2p-1)\ \vec{z}$ we obtain $a + b = p_+\ \vec{\lambda} \cdot \vec{v}_+ + p_-\ \vec{\lambda} \cdot \vec{v}_- = \vec{\lambda} \cdot (p_+\ \vec{v}_+ + p_-\ \vec{v}_-) = (2p-1)\ \vec{\lambda} \cdot \vec{z}$ and therefore:

$$\Theta(p_+\ \vec{\lambda} \cdot \vec{v}_+) + \Theta(p_-\ \vec{\lambda} \cdot \vec{v}_-) \geq \Theta((2p-1)\ \vec{\lambda} \cdot \vec{z}) \tag{2.39}$$

$$p_+\ \Theta(\vec{\lambda} \cdot \vec{v}_+) + p_-\ \Theta(\vec{\lambda} \cdot \vec{v}_-) \geq (2p-1)\ \Theta(\vec{\lambda} \cdot \vec{z}) \tag{2.40}$$

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \geq 0\,. \tag{2.41}$$

Note that we have used $p_+, p_- \geq 0$ and $(2p-1) \geq 0$.

(ii) symmetric:
From $\Theta(a) - \Theta(-a) = a$ we conclude:

$$a + b = (a+b) \tag{2.42}$$

$$\Theta(a) - \Theta(-a) + \Theta(b) - \Theta(-b) = \Theta(a+b) - \Theta(-a-b) \tag{2.43}$$

$$\Theta(a) + \Theta(b) - \Theta(a+b) = \Theta(-a) + \Theta(-b) - \Theta(-a-b)\,. \tag{2.44}$$

Now we can choose $a = p_+\ \vec{\lambda} \cdot \vec{v}_+$ and $b = p_-\ \vec{\lambda} \cdot \vec{v}_-$ such that $a + b = p_+\ \vec{\lambda} \cdot \vec{v}_+ + p_-\ \vec{\lambda} \cdot \vec{v}_- = (2p-1)\ \vec{\lambda} \cdot \vec{z}$. We obtain directly:

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) = \frac{1}{\pi}(\Theta(a) + \Theta(b) - \Theta(a+b)) = \frac{1}{\pi}(\Theta(-a) + \Theta(-b) - \Theta(-a-b)) = \tilde{\rho}_{\vec{x}}(-\vec{\lambda})\,. \tag{2.45}$$

(iii) area:
Since $\int_{S_2} \Theta(\vec{\lambda}\cdot\vec{v})/\pi \ \mathrm{d}\vec{\lambda} = 1$ (see Eq. (2.27)) we obtain by linearity:

$$\int_{S_2} \tilde{\rho}_{\vec{x}}(\vec{\lambda})\ \mathrm{d}\vec{\lambda} = \int_{S_2} \frac{p_+}{\pi}\ \Theta(\vec{\lambda}\cdot\vec{v}_+)\ \mathrm{d}\vec{\lambda} + \int_{S_2} \frac{p_-}{\pi}\ \Theta(\vec{\lambda}\cdot\vec{v}_-)\ \mathrm{d}\vec{\lambda} - \int_{S_2} \frac{(2p-1)}{\pi}\ \Theta(\vec{\lambda}\cdot\vec{z})\ \mathrm{d}\vec{\lambda} \tag{2.46}$$

$$= p_+ + p_- - (2p-1) = 1 - (2p-1) = 2(1-p)\,. \tag{2.47}$$

(iv) 1st bound:
We can use $\Theta(a) + \Theta(b) \geq \Theta(a+b)$ with $a = p_+\ \vec{\lambda}\cdot\vec{v}_+ + p_-\ \vec{\lambda}\cdot\vec{v}_- = (2p-1)\ \vec{\lambda}\cdot\vec{z}$ and $b = -p_-\ \vec{\lambda}\cdot\vec{v}_-$. We obtain $a + b = p_+\ \vec{\lambda}\cdot\vec{v}_+$ and therefore:

$$\Theta((2p-1)\ \vec{\lambda}\cdot\vec{z}) + \Theta(-p_-\ \vec{\lambda}\cdot\vec{v}_-) \geq \Theta(p_+\ \vec{\lambda}\cdot\vec{v}_+) \tag{2.48}$$

$$\Theta((2p-1)\ \vec{\lambda}\cdot\vec{z}) + \Theta(-p_-\ \vec{\lambda}\cdot\vec{v}_-) + \Theta(p_-\ \vec{\lambda}\cdot\vec{v}_-) \geq \Theta(p_+\ \vec{\lambda}\cdot\vec{v}_+) + \Theta(p_-\ \vec{\lambda}\cdot\vec{v}_-) \tag{2.49}$$

$$p_-\ \Theta(-\vec{\lambda}\cdot\vec{v}_-) + p_-\ \Theta(\vec{\lambda}\cdot\vec{v}_-) \geq p_+\ \Theta(\vec{\lambda}\cdot\vec{v}_+) + p_-\ \Theta(\vec{\lambda}\cdot\vec{v}_-) \tag{2.50}$$

$$- (2p-1)\ \Theta(\vec{\lambda}\cdot\vec{z}) \tag{2.51}$$

$$p_-\ |\vec{\lambda}\cdot\vec{v}_-| \geq \pi\cdot\tilde{\rho}_{\vec{x}}(\vec{\lambda})\,. \tag{2.52}$$

If we choose $b = -p_+\ \vec{\lambda}\cdot\vec{v}_+$ instead, we obtain $p_+\ |\vec{\lambda}\cdot\vec{v}_+| \geq \pi\cdot\tilde{\rho}_{\vec{x}}(\vec{\lambda})$.

(v) 2nd bound:
Here we prove:

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{1}{2\pi}\frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)} + C|\cos(\theta)|} \tag{2.53}$$

where $C := 2p-1$ and $\cos(\theta) = \vec{\lambda}\cdot\vec{z}$.

We prove it in the following way: For a given vector, $\vec{\lambda} \in S_2$ we want to find the distribution $\tilde{\rho}_{\vec{x}}$ for which $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ is maximal. First, we focus on a vector $\vec{\lambda}$ in the lower hemisphere ($\vec{\lambda}\cdot\vec{z} \leq 0$). In that region, it turns out that $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) = \frac{1}{\pi}(p_+\Theta(\vec{v}_+\cdot\vec{\lambda}) + p_-\Theta(\vec{v}_-\cdot\vec{\lambda})) = \rho_{\vec{x}}(\vec{\lambda})$ and furthermore, only one of the two terms ($p_+\vec{v}_+\cdot\vec{\lambda}$ or $p_-\vec{v}_-\cdot\vec{\lambda}$) is positive (if both are positive, we have $p_+\vec{v}_+\cdot\vec{\lambda} + p_-\vec{v}_-\cdot\vec{\lambda} = C\ \vec{z}\cdot\vec{\lambda} > 0$ which is a contradiction). Therefore, we want to find, for a given vector $\vec{\lambda}$, $\vec{v}_+$ and p_+ such that $\rho_{\vec{x}}(\vec{\lambda}) = \frac{1}{\pi}(p_+\vec{v}_+\cdot\vec{\lambda})$ is maximal (we choose "+" w.l.o.g.).

For what follows, we choose the following parametrization: $\vec{v}_\pm = (\sin(\alpha_\pm), 0, \cos(\alpha_\pm))^T$, $\vec{\lambda} = (\sin(\theta), 0, \cos(\theta))^T$. Note that the maximum is the same for two different $\vec{\lambda}$ with the same z-component due to the rotational symmetry around the z-axis. This allows us to focus only on the particular choice of $\vec{\lambda} = (\sin(\theta), 0, \cos(\theta))^T$ where we set the y-component to zero. Then the vector $\vec{v}_+$ that achieves the maximum will have zero y-component as well since we want to maximize the inner product between these two vectors. Solving the equation $p_+\vec{v}_+ + p_-\vec{v}_- = C\ \vec{z}$ together with $p_+ + p_- = 1$ leads to:

$$p_+ = \frac{1-C^2}{2-2C\cos(\alpha_+)}, \tag{2.54}$$

$$\sin(\alpha_-) = \frac{(1-C^2)\sin(\alpha_+)}{2C\cos(\alpha_+)-(1+C^2)}, \tag{2.55}$$

$$\cos(\alpha_-) = \frac{(1+C^2)\cos(\alpha_+)-2C}{2C\cos(\alpha_+)-(1+C^2)}. \tag{2.56}$$

Here, $\sin(\alpha_-)$ and $\cos(\alpha_-)$ are stated merely for completeness and are not necessary for what follows. In order to maximize $\frac{1}{\pi}(p_+\vec{v}_+ \cdot \vec{\lambda})$, we have to find the maximal α_+ for the function:

$$\frac{1}{\pi}(p_+\vec{v}_+ \cdot \vec{\lambda}) = \frac{(1-C^2)\cos(\theta-\alpha_+)}{2\pi(1-C\cos(\alpha_+))}. \tag{2.57}$$

Maximizing over α_+ leads to the condition $C\sin(\theta) = \sin(\theta-\alpha_+)$ or $\alpha_+ = \theta - \arcsin(C\sin(\theta))$. For the two expressions in the above function that contain α_+, we obtain $\cos(\alpha_+) = \cos(\theta)\sqrt{1-C^2\sin^2(\theta)} + C\sin^2(\theta)$ and $\cos(\theta-\alpha_+) = \sqrt{1-C^2\sin^2(\theta)}$. This leads to the following bound:

$$\frac{1}{\pi}(p_+\vec{v}_+ \cdot \vec{\lambda}) \leq \frac{1}{2\pi}\frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)}-C\cos(\theta)} = \frac{1}{2\pi}\frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)}+C|\cos(\theta)|}. \tag{2.58}$$

In the second step, we used that $\cos(\theta) \leq 0$ and therefore $\cos(\theta) = -|\cos(\theta)|$. For a vector in the upper hemisphere, we simply observe that the function $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ is symmetric $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) = \tilde{\rho}_{\vec{x}}(-\vec{\lambda})$. Since $-\vec{\lambda} = (-\sin(\theta), 0, -\cos(\theta))^T$ this leads directly to

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{1}{2\pi}\frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)}+C|\cos(\theta)|}, \tag{2.59}$$

since the bound is invariant under changing the sign of $\sin(\theta)$ and $\cos(\theta)$.

(vi) 3rd bound:
We maximize the 2nd bound over θ. The maximum is reached when $\theta = \pi/2$, which leads to:

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \frac{\sqrt{1-C^2}}{2\pi} = \frac{\sqrt{p(1-p)}}{\pi}. \tag{2.60}$$

Note, that this bound is strictly weaker than the second bound but easier to state and useful for pedagogical reasons. □

2.D. Improved one bit protocol

We can improve the protocol from the main text in two independent ways. The first improvement comes from the fact that $p_A(c = 1|\vec{x}, \vec{\lambda}_1) = (4\pi) \cdot \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1) \leq 4\sqrt{p(1-p)}$, where we used the bound $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \sqrt{p(1-p)}/\pi$. Hence, if the state is very weakly entangled ($p \lesssim 1$), the probability that Alice sends the bit $c = 1$ is always small. Intuitively speaking, this allows us to rewrite the protocol into a form where Alice and Bob only communicate in a fraction of rounds, but in those rounds with a higher (rescaled) probability $p_A(c = 1|\vec{x}, \vec{\lambda}_1) \propto \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1)$. In the limit where p approaches one (the separable state $|00\rangle\langle 00|$), the fraction of rounds in which they have to communicate at all approaches even zero. The second improvement comes from using a better bound for the function $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$. Indeed, the bound $\tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \sqrt{p(1-p)}/\pi$ is easy to state but not optimal. More precisely, we have proven in Appendix 2.C:

$$0 \leq \tilde{\rho}_{\vec{x}}(\vec{\lambda}) \leq \tilde{\rho}_{max}(\vec{\lambda}) := \frac{1}{2\pi} \frac{1 - C^2}{\sqrt{1 - C^2 \sin^2(\theta)} + C|\cos(\theta)|} \,. \tag{2.61}$$

Here, $C := 2p - 1$ and $\cos(\theta) = \vec{\lambda} \cdot \vec{z}$ is the z component of $\vec{\lambda}$ in spherical coordinates. Note that, neither $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ nor $\tilde{\rho}_{max}(\vec{\lambda})$ are normalized, and we define the function $N(p)$ as the normalization of that function $\tilde{\rho}_{max}(\vec{\lambda})$:

$$N(p) := \int_{S_2} \tilde{\rho}_{max}(\vec{\lambda}) \, \mathrm{d}\vec{\lambda} \,. \tag{2.62}$$

To not scare off the reader, we evaluate the integral after we present the protocol.

Protocol 2.5 ($0.835 \leq p \leq 1$, 1 bit in the worst case, $N(p)$ bits on average)**.** *Same as Protocol 2.0 with the following 2. Step:*
Alice and Bob share two random vectors $\vec{\lambda}_1, \vec{\lambda}_2 \in S_2$ according to the distribution:

$$\rho(\vec{\lambda}_1) = \frac{1}{N(p)} \cdot \tilde{\rho}_{max}(\vec{\lambda}_1) \,, \qquad \rho(\vec{\lambda}_2) = \frac{1}{\pi} \, \Theta(\vec{\lambda}_2 \cdot \vec{z}) \,. \tag{2.63}$$

In addition, Alice and Bob share a random bit r distributed according to $p(r = 0) = 1 - N(p)$ and $p(r = 1) = N(p)$. If $r = 0$, Alice and Bob do not communicate and both set $\vec{\lambda} := \vec{\lambda}_2$. If $r = 1$, Alice sets $c = 1$ with probability:

$$p(c = 1|\vec{x}, \vec{\lambda}_1) = \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1)/\tilde{\rho}_{max}(\vec{\lambda}_1) \tag{2.64}$$

and otherwise she sets $c = 2$. She communicates the bit c to Bob. Both set $\vec{\lambda} := \vec{\lambda}_c$ and reject the other vector.

Proof. We show that the distribution of the shared vector $\vec{\lambda}$ becomes exactly the required $\rho_{\vec{x}}(\vec{\lambda})$. To see that, consider all the cases where Alice chooses the first vector. This happens only when $r = 1$ and when she sets the bit c to 1. This samples the distribution:

$$p(r = 1) \cdot p(c = 1|\vec{x}, \vec{\lambda}_1) \cdot \rho(\vec{\lambda}_1) = \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1) \,. \tag{2.65}$$

The total probability that she is choosing the first vector is $\int_{S_2} p(r=1)\cdot p(c=1|\vec{x},\vec{\lambda}_1)\cdot \rho(\vec{\lambda}_1)\,\mathrm{d}\vec{\lambda}_1 = \int_{S_2} \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1)\,\mathrm{d}\vec{\lambda}_1 = 2(1-p)$. In all the remaining cases, with total probability $2p-1$, she is choosing vector $\vec{\lambda} := \vec{\lambda}_2$. Therefore, the total distribution of the chosen vector $\vec{\lambda}$ becomes the desired distribution

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) + \frac{(2p-1)}{\pi}\,\Theta(\vec{\lambda}\cdot\vec{z}) = \rho_{\vec{x}}(\vec{\lambda})\,. \tag{2.66}$$

Here, the first term $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ in the sum corresponds to all the instances where Alice chooses $\vec{\lambda} := \vec{\lambda}_1$ and the second term to all the instances, where she chooses $\vec{\lambda}_2$. In order for the protocol to be well defined, it has to hold that $0 \le p(c=1|\vec{x},\vec{\lambda}_1) \le 1$ and $0 \le N(p) \le 1$. The first bound is true since $0 \le \tilde{\rho}_{\vec{x}}(\vec{\lambda}_1) \le \tilde{\rho}_{max}(\vec{\lambda}_1)$ and the second bound $0 \le N(p) \le 1$ holds whenever $0.835 \le p \le 1$. □

We can explicitly solve that integral for $N(p)$ by using spherical coordinates $\vec{\lambda} = (\sin\theta\cdot\cos\phi, \sin\theta\cdot\sin\phi, \cos\theta)$:

$$\begin{aligned}
N(p) = \int_{S_2} \tilde{\rho}_{max}(\vec{\lambda})\,\mathrm{d}\vec{\lambda} &= \int_0^{2\pi}\int_0^{\pi} \tilde{\rho}_{max}(\theta,\phi)\cdot\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi && (2.67)\\
&= \int_0^{2\pi}\int_0^{\pi} \frac{1}{2\pi}\frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)}+C|\cos(\theta)|}\cdot\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi && (2.68)\\
&= \int_0^{\pi} \frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)}+C|\cos(\theta)|}\cdot\sin\theta\,\mathrm{d}\theta && (2.69)\\
&=2\int_0^{\pi/2} \frac{1-C^2}{\sqrt{1-C^2\sin^2(\theta)}+C\cos(\theta)}\cdot\sin\theta\,\mathrm{d}\theta\,. && (2.70)
\end{aligned}$$

Substituting $x = C\cdot\cos\theta$ leads to:

$$\begin{aligned}
N(p) =&\frac{2(1-C^2)}{C}\int_0^C \frac{1}{\sqrt{(1-C^2)+x^2}+x}\,\mathrm{d}x && (2.71)\\
=&\frac{2(1-C^2)}{C}\left[\frac{1}{2}\log\left(\sqrt{(1-C^2)+x^2}+x\right) - \frac{1-C^2}{4\left(\sqrt{(1-C^2)+x^2}+x\right)^2}\right]_0^C && (2.72)\\
=&\frac{1-C^2}{2C}\left(\log\left(\frac{1+C}{1-C}\right)+\frac{2C}{1+C}\right) && (2.73)\\
=&\frac{1-C^2}{2C}\log\left(\frac{1+C}{1-C}\right)+(1-C) && (2.74)\\
=&\frac{2p(1-p)}{2p-1}\log\left(\frac{p}{1-p}\right)+2(1-p)\,. && (2.75)
\end{aligned}$$

Here we used $C = 2p-1$ in the last step.

2.E. Maximal local content

In Appendix 2.D, we showed that it is possible to simulate weakly entangled states without communication in some fraction of rounds. One can even go one step further and maximize the fraction of rounds in which no communication is required. This is called the local content of the state $|\Psi_{AB}\rangle$. More formally, a simulation of an entangled qubit pair can be decomposed into a local part $p_L(a,b|\vec{x},\vec{y})$ that can be implemented by Alice and Bob without communication and the remaining non-local content, denoted as $p_{NL}(a,b|\vec{x},\vec{y})$:

$$p_Q(a,b|\vec{x},\vec{y}) = p_L \cdot p_L(a,b|\vec{x},\vec{y}) + (1-p_L)\cdot p_{NL}(a,b|\vec{x},\vec{y})\,. \tag{2.76}$$

The problem consists in finding the maximal value of p_L for a given state $|\Psi_{AB}\rangle = \sqrt{p}\,|00\rangle + \sqrt{1-p}\,|00\rangle$, that we denote here as $p_L^{max}(p)$. For a maximally entangled state, Elitzur, Popescu, and Rohrlich showed that the local content is necessarily zero (hence $p_L^{max}(p=1/2)=0$), also known as the EPR2 decomposition [93] (see also Barrett et al. [94]). For general entangled qubit pairs, it was shown by Scarani [114] that the local content is upper bounded by $2p-1$, hence $p_L^{max}(p) \leq 2p-1$. At the same time, subsequently better lower bounds were found [93, 114, 115, 111]. Finally, Portmann et al. [111] found an explicit decomposition with a local content of $p_L(p) = 2p-1$, hence proving that the upper and lower bound coincide and, therefore, $p_L^{max}(p) = 2p-1$. Here, we give an independent proof of the result by Portmann et al. [111]. More precisely, we provide a protocol that simulates any pure entangled two-qubit state of the general form $|\Psi_{AB}\rangle = \sqrt{p}\,|00\rangle + \sqrt{1-p}\,|00\rangle$ with a local content of $p_L = 2p-1$.

We remark that often in the literature, the state $|\Psi_{AB}\rangle$ is written as $|\Psi_{AB}\rangle = \cos\theta\,|00\rangle + \sin\theta\,|00\rangle$ where $\cos\theta \geq \sin\theta$. These two notations are related through the following expressions: $\cos 2\theta = 2p-1$ which follows from $\cos\theta = \sqrt{p}$ and $\sin\theta = \sqrt{1-p}$ together with $\cos 2\theta = \cos^2\theta - \sin^2\theta = p-(1-p) = 2p-1$.

Protocol 2.6 ($1/2 \leq p \leq 1$, maximal local content of $p_L = 2p-1$)**.** *Same as Protocol 2.0 with the following 2. Step:*
Alice and Bob share a random vector $\vec{\lambda}_1 \in S_2$ according to the distribution ($\vec{z} := (0,0,1)^T$):

$$\rho(\vec{\lambda}_1) = \frac{1}{\pi}\,\Theta(\vec{\lambda}_1\cdot\vec{z})\,. \tag{2.77}$$

In addition, Alice and Bob share a random bit r distributed according to $p(r=0) = 2p-1$ and $p(r=1) = 2(1-p)$. If $r=0$, Alice and Bob do not communicate and both set $\vec{\lambda} := \vec{\lambda}_1$. If $r=1$, Alice samples $\vec{\lambda} \in S_2$ according to the distribution $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$ and communicates that $\vec{\lambda}$ to Bob. (Alice can for example encode the three coordinates of $\vec{\lambda}$ and sends this information to Bob.)

Proof. Whenever $r=1$, the resulting distribution of $\vec{\lambda}$ is, by construction, $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$. Whenever, $r=0$ (with probability $2p-1$) the distribution of $\vec{\lambda}$ is $\Theta(\vec{\lambda}\cdot\vec{z})/\pi$. Therefore, the total distribution for the chosen vector $\vec{\lambda}$ becomes the desired distribution

$$\tilde{\rho}_{\vec{x}}(\vec{\lambda}) + \frac{(2p-1)}{\pi}\,\Theta(\vec{\lambda}\cdot\vec{z}) = \rho_{\vec{x}}(\vec{\lambda})\,. \tag{2.78}$$

Hence, by the proof of Protocol 2.0, this exactly reproduces quantum correlations. □

This directly provides a decomposition of the above form (Eq. (2.76)) for $p_L = 2p-1$. More precisely, whenever $r=0$ (with total probability $2p-1$), they do not communicate and, hence, implement a local strategy $p_L(a,b|\vec{x},\vec{y})$. On the other hand, whenever $r=0$ they communicate and, therefore, implement a non-local behavior $p_{NL}(a,b|\vec{x},\vec{y})$. That protocol requires an unbounded amount of communication in the rounds where $r=1$ and there are more efficient ways to sample the distribution $\tilde{\rho}_{\vec{x}}(\vec{\lambda})$. However, this is not an issue for determining $p_L^{max}(p)$ since we only maximize the local content, hence, the number of rounds in which Alice and Bob do not have to communicate. At the same time, we are not concerned with the amount of communication in the remaining rounds. One can also ask, what is the maximal local content under the restriction that Alice and Bob only communicate a single bit in the remaining rounds. We found in Appendix 2.D such a decomposition. However, in general, it seems unlikely that a decomposition that attains the maximal local content of $p_L^{max}(p) = 2p-1$ can be achieved if the two parties communicate only a single bit in the remaining rounds.

3. Compatibility of generalized noisy qubit measurements

This chapter is based on the article:

Martin J. Renner "Compatibility of Generalized Noisy Qubit Measurements," Physical Review Letters 132 (25), 250202 (2024)
(Received: 25 October 2023, Revised: 12 March 2024, Accepted: 11 April 2024, Published: 17 June 2024)

Contributions: The results were found and the manuscript was written by myself. All technical proofs were carried out by myself.

Abstract

It is a crucial feature of quantum mechanics that not all measurements are compatible with each other. However, if measurements suffer from noise they may lose their incompatibility. Here, we consider the effect of white noise and determine the critical visibility such that all qubit measurements, i.e. all positive operator-valued measures (POVMs), become compatible, i.e. jointly measurable. In addition, we apply our methods to quantum steering and Bell nonlocality. We obtain a tight local hidden state model for two-qubit Werner states of visibility $1/2$. This determines the exact steering bound for two-qubit Werner states and also provides a local hidden variable model that improves on previously known ones. Interestingly, this proves that POVMs are not more powerful than projective measurements to demonstrate quantum steering for these states.

3.1. Introduction

Quantum mechanics provides a remarkably accurate framework for predicting the outcomes of experiments and has led to the development of numerous technological advancements. Despite its successes, it presents us with puzzling and counterintuitive phenomena that challenge our classical notions of reality. One of the key aspects that set quantum mechanics apart from classical physics is the concept of measurement incompatibility. In classical physics, measuring one property of a system need not affect the measurement of another property. In quantum mechanics, however, the situation is radically different. The uncertainty principle, formulated by Werner Heisenberg, establishes a fundamental limit to the precision with which certain pairs of properties can be simultaneously known [1].

A simple and well-known example is the fact that we cannot simultaneously measure the spin of a particle in two orthogonal directions. It is known that incompatible measurements are at the core of many quantum information tasks. For example, they are necessary to violate Bell inequalities [116, 117, 118] and necessary to provide an advantage in quantum communication [119, 83, 120] or state discrimination tasks [121, 122, 123] (see also the reviews [124, 125]).

However, measurement devices always suffer from imprecision. Therefore, an apparatus measures in practice only a noisy version of the measurements. If the noise gets too large, these noisy measurements can become compatible even though they are incompatible in the noiseless limit [20]. In that case, the statistics of these noisy measurements can be obtained from the statistics of just a single measurement, and we say that these noisy measurements are jointly measurable. However, a detector that can only perform compatible measurements has limited power. Most importantly, it cannot be used for many quantum information processing tasks like demonstrating Bell-nonlocality since these require incompatible measurements. It is therefore important to ask, how much noise can be tolerated before all measurements become jointly measurable.

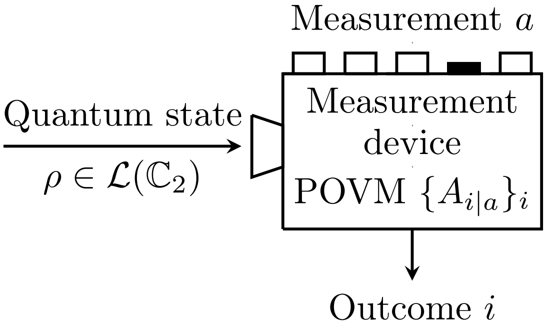


Figure 3.1.: A measurement device can perform different measurements (labeled with a) that produce an outcome i. If the measurements are too noisy they can be simulated by a device that just performs a single measurement. In this work, we address the question of how much white noise can be tolerated before all qubit measurements become jointly measurable.

In this work, we study the effect of white noise and show that all qubit measurements become jointly measurable at a critical visibility of 1/2. This result has direct implications for related fields of quantum information, in particular, Bell nonlocality [4, 77] and quantum steering [3, 18, 126, 127, 128, 19, 129]. More precisely, we use the close connection between joint measurability and quantum steering [130, 131, 132] to show that the two-qubit Werner state [17]

$$\rho_W^\eta = \eta \, |\Psi^-\rangle\langle\Psi^-| + (1-\eta)\, \mathbb{1}/4 \tag{3.1}$$

cannot demonstrate EPR-steering if $\eta \leq 1/2$. Here, $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ denotes the two-qubit singlet state. This also implies that the same state does not violate any Bell inequality for arbitrary positive operator-valued measurements (POVM) applied on both sides whenever $\eta \leq 1/2$.

3.2. Notation and joint measurability

Before we introduce the problem, we introduce the necessary notation. Qubit states are described by positive semidefinite 2×2 complex operators $\rho \in \mathcal{L}(\mathbb{C}_2)$, $\rho \geq 0$ with unit trace $\mathrm{tr}[\rho] = 1$. They can be represented as $\rho = (\mathbb{1} + \vec{x} \cdot \vec{\sigma})/2$, where $\vec{x} \in \mathbb{R}^3$ is a three-dimensional real vector such that $|\vec{x}| \leq 1$, and $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ are the standard Pauli matrices. In this notation, $\vec{x}$ is the corresponding Bloch vector of the qubit state. General qubit measurements are described by a positive operator-valued measure (POVM), which is a set of positive semidefinite operators $A_{i|a} \geq 0$ that sum to the identity, $\sum_i A_{i|a} = \mathbb{1}$. Here, we use the label "a" to distinguish between different measurements, while "i" denotes the outcome of a given POVM (see also Fig. 3.1). In quantum theory, the probability of outcome i when performing the POVM with elements $A_{i|a}$ on the state ρ is given by Born's rule,

$$p(i|a,\rho) = \mathrm{tr}\big[A_{i|a}\ \rho\big]\,. \tag{3.2}$$

Because every qubit POVM can be written as a coarse-graining of rank-1 projectors [85], we may restrict ourselves to POVMs proportional to rank-1 projectors. (We could also restrict ourselves to POVMs with at most four outcomes [59] but this is not necessary in what follows.) Thus, we write the measurements as $A_{i|a} = p_i\,|\vec{a}_i\rangle\langle\vec{a}_i|$, where $p_i \geq 0$ and $|\vec{a}_i\rangle\langle\vec{a}_i| = \big(\mathbb{1} + \vec{a}_i \cdot \vec{\sigma}\big)/2$ for some normalized vector $\vec{a}_i \in \mathbb{R}^3$ ($|\vec{a}_i| = 1$). As a consequence of $\sum_i A_{i|a} = \mathbb{1}$ we obtain $\sum_i p_i = 2$ and $\sum_i p_i\ \vec{a}_i = \vec{0}$.

These expressions are valid if all measurements are perfectly implemented. However, noise is usually unavoidable in experiments. In this work, we study the effect of white noise, where η denotes the visibility. More formally, we define the noisy measurements as:

$$A^{\eta}_{i|a} = \eta\ A_{i|a} + (1-\eta)\ \ \mathrm{tr}\big[A_{i|a}\big]\ \mathbb{1}/2\,. \tag{3.3}$$

With the notation introduced above the POVM elements become $A^{\eta}_{i|a} = p_i\big(\mathbb{1} + \eta\ \vec{a}_i \cdot \vec{\sigma}\big)/2$. The goal of this work is to determine the critical value of η such that all qubit POVMs become jointly measurable.

A set of measurements $\{A_{i|a}\}_{i,a}$ is jointly measurable if there exists a single measurement (so-called parent POVM) $\{G_\lambda\}_\lambda$ such that the statistics of all measurements in the set can be obtained by classical post-processing of the data of that single parent measurement. More precisely, if for every POVM in the set there exist conditional probabilities $p(i|a,\lambda)$ (with $0 \leq p(i|a,\lambda) \leq 1$ and $\sum_i p(i|a,\lambda) = 1$) such that:

$$A_{i|a} = \sum_\lambda p(i|a,\lambda)\ G_\lambda\,. \tag{3.4}$$

If this is satisfied, all measurements in the set can be simulated by the single parent POVM with operators G_λ: First, the parent POVM is measured on the quantum state ρ in which outcome λ occurs with probability $p(\lambda|\rho) = \mathrm{tr}[G_\lambda\ \rho]$. Second, given the POVM labeled by "a" that we want to simulate, the outcome i is produced with probability $p(i|a, \lambda)$. In total, the probability of outcome i becomes:

$$\sum_\lambda p(i|a,\lambda)p(\lambda|\rho) = \sum_\lambda p(i|a,\lambda)\,\mathrm{tr}[G_\lambda\ \rho] = \mathrm{tr}\big[A_{i|a}\ \rho\big]\,. \tag{3.5}$$

Here, we used the linearity of the trace. This perfectly simulates a given POVM with elements $\{A_{i|a}\}_i$, since this is the same expression as if the measurement was directly performed on the quantum state ρ given in Eq. (3.2).

The most prominent example are the two noisy spin-measurements $A^\eta_{\pm|x} = (\mathbb{1} \pm 1/\sqrt{2}\ \sigma_x)/2$ and $A^\eta_{\pm|z} = (\mathbb{1} \pm 1/\sqrt{2}\ \sigma_z)/2$ where $\eta = 1/\sqrt{2}$. We can consider the following measurement with four outcomes $\lambda = (i, j)$ where $i, j \in \{+1, -1\}$:

$$G_{(i,j)} = \frac{1}{4}\left(\mathbb{1} + \frac{i}{\sqrt{2}}\sigma_x + \frac{j}{\sqrt{2}}\sigma_z\right)\,. \tag{3.6}$$

One can check that this is a valid POVM and that $A^\eta_{i|x} = \sum_j G_{(i,j)}$ as well as $A^\eta_{j|z} = \sum_i G_{(i,j)}$. Therefore, the statistics of both measurements $\{A^\eta_{i|x}\}_i$ and $\{A^\eta_{j|z}\}_j$ can be obtained from the statistics of just a single parent measurement. Now we consider not only two but the set of all noisy qubit POVMs $\{A^\eta_{i|a}\}_{i,a}$ and show that for $\eta \leq 1/2$ this set becomes jointly measurable.

3.3. The protocol

First, we define two functions. The first one is the sign function, which is defined as $\mathrm{sgn}\,(x) := +1$ if $x \geq 0$ and $\mathrm{sgn}\,(x) := -1$ if $x < 0$. Similarly, the function $\Theta(x)$ is defined as $\Theta(x) := x$ if $x \geq 0$ and $\Theta(x) := 0$ if $x < 0$ (or, $\Theta(x) := (|x| + x)/2$).

The parent POVM $\{G_{\vec{\lambda}}\}_{\vec{\lambda}}$ is the measurement with elements

$$G_{\vec{\lambda}} = \frac{1}{4\pi}(\mathbb{1} + \vec{\lambda}\cdot\vec{\sigma})\,. \tag{3.7}$$

Here, $\vec{\lambda} \in \mathbb{R}^3$ is a normalized vector uniformly distributed on the unit radius sphere S_2. Physically, this corresponds to a (sharp) projective measurement with outcome $\vec{\lambda}$, where the measurement direction is chosen Haar-random on the Bloch sphere.[1]

For a given POVM with operators $A^{1/2}_{i|a} = p_i\big(\mathbb{1} + 1/2\ \vec{a}_i\cdot\vec{\sigma}\big)/2$ where $\sum_i p_i = 2$, $|\vec{a}_i| = 1$, and $\sum_i p_i\ \vec{a}_i = \vec{0}$, we define the following function that associates a real-valued number to each point in $\vec{x} \in \mathbb{R}^3$:

$$f_a : \mathbb{R}^3 \to \mathbb{R} :\ f_a(\vec{x}) := \sum_i p_i\ \Theta(\vec{x}\cdot\vec{a}_i)\,. \tag{3.8}$$

[1] Note that, in contrary to all other POVMs in this work, the parent POVM has a continuous set of outcomes $\vec{\lambda}$ and needs to satisfy $\int_{S_2} \mathrm{d}\vec{\lambda}\ G_{\vec{\lambda}} = \mathbb{1}$ as well as $G_{\vec{\lambda}} \geq 0$.

Now, we choose an orthonormal coordinate frame of the Bloch sphere, defined by the three pairwise orthogonal unit vectors $\vec{x}', \vec{y}', \vec{z}' \in S_2$. In addition, we define the eight vectors $\vec{v}_{s_x s_y s_z} := s_x \vec{x}' + s_y \vec{y}' + s_z \vec{z}'$ where $s_x, s_y, s_z \in \{+1, -1\}$. This frame shall be chosen such that $f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ for all of these eight vectors and we show below that one can always find such a coordinate frame. Note that the vectors $\vec{v}_{s_x s_y s_z}$ are the vertices of a cube with sidelength two that is centered at the origin of the Bloch sphere.

After choosing a suitable frame, we can define the conditional probabilities:

$$p(i|a, \vec{\lambda}) = p_i \; \Theta(\vec{a}_i \cdot \vec{v}_{s_x s_y s_z}) + \frac{(1 - f_a(\vec{v}_{s_x s_y s_z}))\alpha_i}{\sum_i \alpha_i} \,. \tag{3.9}$$

Here, $\vec{v}_{s_x s_y s_z}$ is the vector with indices $s_k = \operatorname{sgn}(\vec{\lambda} \cdot \vec{k}')$ for $k \in \{x, y, z\}$. Hence, the three signs s_k denote the octant of $\vec{\lambda}$ in the rotated frame defined by $\vec{x}', \vec{y}', \vec{z}'$. (Equivalently, $\vec{v}_{s_x s_y s_z}$ is the vertex of the cube closest to $\vec{\lambda}$.) In addition, α_i is defined as:

$$\alpha_i := \frac{p_i}{2} \left(1 - \frac{1}{4} \sum_{s_x, s_y, s_z = \pm 1} \Theta(\vec{a}_i \cdot \vec{v}_{s_x s_y s_z}) \right) \,. \tag{3.10}$$

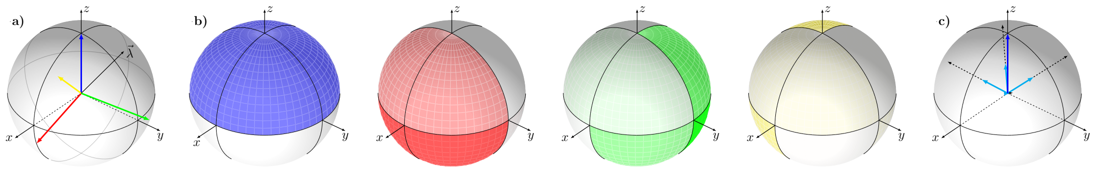


Figure 3.2.: An illustration for a SIC-POVM [133]: **a)** The different outcomes i are represented with different colors and the colored vectors represent $\vec{a}_i$ (note also $p_i = 1/2$ for $i = 1, 2, 3, 4$). **b)** The opacity of the colors represents the probability to output i given that $\vec{\lambda}$ lies in that region of the sphere, hence $p(i|a, \vec{\lambda})$. This function is constant in each octant of the chosen frame, which is simply the standard coordinate frame in this case. For the $\vec{\lambda}$ shown in the left sphere ($s_x = -1$, $s_y = s_z = +1$), the outcome is most likely blue (50%) or green (49%). **c)** Collecting all results $\vec{\lambda}$ from one octant behaves like the operator $G_{s_x s_y s_z}$ represented by the cyan arrows for the blue outcome. The sum of these operators simulates the desired (blue) operator $A_{1|a}^{1/2}$. (more details in Appendix 3.E.2)

3.4. Idea of the protocol

Suppose for now that it is possible to find a suitable frame in which $f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ for all eight vectors $\vec{v}_{s_x s_y s_z}$. Since this part is more technical, we discuss it at the end of this section. We can check first that the conditional probabilities are indeed well-defined.

Namely, they are positive and sum to one. Positivity follows from the fact that $p_i \geq 0$ and $\Theta(x) \geq 0$ (for all $x \in \mathbb{R}$). In addition, $f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ and the proof that $\alpha_i \geq 0$ is given in the Appendix 3.B (see Lemma 3.1 (2)). A quick calculation also shows that the probabilities sum to one:

$$\sum_i p(i|a,\vec{\lambda}) = f_a(\vec{v}_{s_x s_y s_z}) + (1 - f_a(\vec{v}_{s_x s_y s_z})) = 1\,. \tag{3.11}$$

Now we are in a position to show that

$$A_{i|a}^{1/2} = \int_{S_2} \mathrm{d}\vec{\lambda}\; p(i|a,\vec{\lambda})\; G_{\vec{\lambda}}\,. \tag{3.12}$$

We give the detailed proof in Appendix 3.D but sketch the main idea here. It is important to recognize that, the function $p(i|a,\vec{\lambda})$ is the same for two different $\vec{\lambda}$ that lie in the same octant of the rotated frame $\vec{x}'$, $\vec{y}'$, $\vec{z}'$. Intuitively speaking, this leads to a coarse-graining of the measurement outcomes $\vec{\lambda}$ in each of these octants. These coarse-grained operators $G_{s_x s_y s_z}$ behave like a noisy measurement in the direction of the corresponding vector $\vec{v}_{s_x s_y s_z}$. More precisely, we calculate in Appendix 3.D.1 that:

$$G_{s_x s_y s_z} := \int_{S_2 |\, \mathrm{sgn}\,(\vec{\lambda}\cdot\vec{k}') = s_k} \mathrm{d}\vec{\lambda}\; G_{\vec{\lambda}} = \frac{\mathbb{1}}{8} + \frac{\vec{v}_{s_x s_y s_z}\cdot\vec{\sigma}}{16}\,. \tag{3.13}$$

With this definition, Eq. (3.12) becomes:

$$A_{i|a}^{1/2} = \sum_{s_x,s_y,s_z=\pm 1} p(i|a,\vec{\lambda})\; G_{s_x s_y s_z}\,. \tag{3.14}$$

Using the definition of $p(i|a,\vec{\lambda})$ in Eq. (3.9) and some algebra (details in Appendix 3.D) this reduces to:

$$A_{i|a}^{1/2} = \sum_{s_x,s_y,s_z=\pm 1} p_i\; \Theta(\vec{a}_i\cdot\vec{v}_{s_x s_y s_z})\; G_{s_x s_y s_z} + \alpha_i \mathbb{1}\,. \tag{3.15}$$

In the end, we prove this identity by using a closely related geometric formula that decomposes $\vec{a}_i$ into the vectors $\vec{v}_{s_x s_y s_z}$ (see Appendix 3.B):

$$\sum_{s_x,s_y,s_z=\pm 1} \Theta(\vec{a}_i\cdot\vec{v}_{s_x s_y s_z})\; \vec{v}_{s_x s_y s_z} = 4\,\vec{a}_i\,. \tag{3.16}$$

The identity in Eq. (3.15) can be seen as the main idea of the protocol. We want to find a set of coarse-grained operators $G_{s_x s_y s_z}$ that can be used to decompose all the POVM elements $A_{i|a}^{1/2}$. The conditional probabilities $p(i|a,\vec{\lambda})$ are then constructed according to this decomposition. The first term in $p(i|a,\vec{\lambda})$, namely $p_i\;\Theta(\vec{v}_{s_x s_y s_z}\cdot\vec{a}_i)$ is the coefficient that comes from the decomposition of $A_{i|a}^{1/2}$ in terms of $G_{s_x s_y s_z}$. The second term in $p(i|a,\vec{\lambda})$ is constructed to add the noise term $\alpha_i\mathbb{1}$.

To give an example, consider the blue vector in Fig. 3.2 for which $\vec{a}_1 = (0,0,1)^T$ and $p_1 = 1/2$, hence $A^{1/2}_{1|a} = p_1\big(\mathbb{1} + 1/2\ \vec{a}_1 \cdot \vec{\sigma}\big)/2 = \mathbb{1}/4 + \sigma_z/8$. It turns out that we can use the standard coordinate frame in which the cube vertices are simply $\vec{v}_{\pm\pm\pm} := (\pm 1, \pm 1, \pm 1)^T$. Direct calculation shows that $p_1 \cdot \Theta(\vec{a}_1 \cdot \vec{v}_{s_x s_y s_z}) = 1/2$ if $s_z = +1$ (and zero if $s_z = -1$) as well as $\alpha_1 = 0$. In addition, the coarse-grained operators become $G_{s_x s_y s_z} = \mathbb{1}/8 + (s_x \sigma_X + s_y \sigma_Y + s_z \sigma_Z)/16$. It is then easy to check that $1/2(G_{+++} + G_{+-+} + G_{-++} + G_{--+}) = A^{1/2}_{1|a}$.

However, while the identity in Eq. (3.15) holds for any orthonormal frame, it can be translated into a protocol with well-defined probabilities only if $\sum_i p_i \cdot \Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a}_i) = f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ for all eight vertices of the cube. We show now that such a frame always exists. The proof has two steps. First, we show that for any such cube, it holds that:

$$\sum_{s_x, s_y, s_z = \pm 1} f_a(\vec{v}_{s_x s_y s_z}) \leq 8\,. \tag{3.17}$$

The second part of the proof uses a theorem by Hausel, Makai, and Szűcs [134] (see Theorem 1 in that reference) that applies to continuous real-valued functions on S_2 that have the additional property that $f(\vec{x}) = f(-\vec{x})$. By using similar techniques as in Ref. [103], we show in Appendix 3.C that these conditions are indeed fulfilled. In their theorem, they show that there always exists a rotation of the cube such that the functional values coincide at all eight vertices of that cube. Hence, choosing the orthonormal frame according to that rotation, we obtain $f_a(\vec{v}_{s_x s_y s_z}) = C$ for all $s_x, s_y, s_z \in \{+1, -1\}$. Combining this with the above bound in Eq. (3.17), we get $8C \leq 8$ and therefore $f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ for that specific cube (see Appendix 3.C for more details).

The theorem in Ref. [134] is a special case of a family of so-called Knaster-type theorems. They state that for a given continuous real-valued function on the sphere, a certain configuration of points can always be rotated such that the functional values coincide at each of these points. Other interesting related results concerning S_2 are due to Dyson [135], Livesay [136], Floyd [137]. Also, the well-known Borsuk–Ulam theorem is of this type [138].

We want to remark that we do not necessarily have to choose a cube in which all of these eight values coincide. It is only required that all of these eight values are smaller than one. Note that we do not give an explicit way to construct such a coordinate frame. However, in many cases, for instance, for POVMs with two or three outcomes, it turns out that an explicit construction can be found. We discuss this further in Appendix 3.A and Appendix 3.E (see also there for further examples and more illustrations).

3.5. Local models for entangled quantum states

Now we apply the developed techniques to Bell nonlocality and quantum steering. Suppose Alice and Bob share a two-qubit Werner state [17]:

$$\rho_W^\eta = \eta \left|\Psi^-\middle\rangle\!\middle\langle\Psi^-\right| + (1-\eta)\ \mathbb{1}/4\,, \tag{3.18}$$

where $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ denotes the two-qubit singlet. They can apply arbitrary local POVMs on their qubit. As before, we denote Alice's measurement operators with $A_{i|a} = p_i\,|\vec{a}_i\rangle\langle\vec{a}_i| = \big(\mathbb{1} + \vec{a}_i \cdot \vec{\sigma}\big)/2$ (where $p_i \geq 0$, $|\vec{a}_i| = 1$, $\sum_i p_i = 2$, and $\sum_i p_i\ \vec{a}_i = \vec{0}$). Similarly, Bob can perform an arbitrary POVM with elements $B_{j|b}$ that are defined analogously. Note, that Alice's and Bob's measurements are now completely arbitrary, i.e. they are not noisy. Instead, the entangled state is not pure but has a certain amount of white noise. The correlations when Alice and Bob apply local POVMs to this state become:

$$p(i,j|a,b) = \mathrm{tr}\big[(A_{i|a} \otimes B_{j|b})\ \rho_W^\eta\big]\,. \tag{3.19}$$

If these correlations $p(i,j|a,b)$ admit a local hidden variable model

$$p(i,j|a,b) = \int \mathrm{d}\lambda\ q(\lambda)\ p(i|a,\lambda)\ p(j|b,\lambda)\,, \tag{3.20}$$

we say that the state ρ_W^η is local. Here, λ is the hidden variable (distributed according to $q(\lambda)$), and $p(i|a,\lambda)$ $(p(j|b,\lambda))$ are the conditional probabilities for Alice (Bob) to output i (j) given their measurement and the value of the hidden variable λ. If the state ρ_W^η violates a Bell inequality, such a local hidden variable description cannot exist and we say that the state is nonlocal. It is a fundamental question in Bell nonlocality, for which η these correlations can violate a Bell inequality or admit a local hidden variable model. It is known, that two-qubit Werner states violate the CHSH inequality [139] for $\eta > 1/\sqrt{2} \approx 0.7071$. Vertesi showed that they violate another Bell inequality whenever $\eta > 0.7056$ [140].

On the other hand, Werner constructed in his seminal paper from 1989 a local model for all bipartite projective measurements if $\eta \leq 1/2$ albeit these states are entangled if $\eta > 1/3$ [17]. Later, this bound was improved by Acin, Toner, and Gisin, who showed that the state is local whenever $\eta \leq 1/K_G(3)$ [141]. Here, $K_G(3)$ is the so-called Grothendieck constant of order three and the best current bound is by Designolle et al. $1.4367 \leq K_G(3) \leq 1.4546$ [142]. This implies that ρ_W^η is local if $\eta \leq 0.6875$ and violates a Bell inequality if $\eta \geq 0.6961$. However, these local models only apply to projective measurements (where $p_1 = p_2 = 1$ and $\vec{a}_2 = -\vec{a}_1$).

Considering general POVMs, Barrett found a local model for all POVMs whenever $\eta \leq 5/12$ [85]. Using a technique developed in Ref. [143, 144], the best bound is again by Ref. [142] which shows that ρ_W^η is local for all POVMs if $\eta \leq 0.4583$. Based on the connections made in Ref. [130, 131, 132], we can now show that whenever $\eta \leq 1/2$ we cannot violate any Bell inequality since all correlations can be described by the following local model.

Suppose Alice performs her measurement $\{A_{i|a}\}_i$ on the Werner state with $\eta = 1/2$ and obtains outcome i. After doing so, Bob's qubit is precisely in the (unnormalized) post-measurement state:

$$\rho_B(i|a) = \mathrm{tr}_A[(A_{i|a} \otimes \mathbb{1})\rho_W^{1/2}] = p_i\big(\mathbb{1} - \vec{a}_i \cdot \vec{\sigma}/2\big)/4\,. \tag{3.21}$$

It is now important to recognize that this state can be simulated with the same techniques as before, due to the duality of states and measurements. More precisely, consider the following protocol:

1. $\vec{\lambda} \in \mathbb{R}^3$ is a normalized vector, drawn randomly from the unit radius sphere S_2 (according to the Haar measure). Alice knows the vector $\vec{\lambda}$. Bob's system is in the pure qubit state $\rho_{\vec{\lambda}} = \big(\mathbb{1} + \vec{\lambda} \cdot \vec{\sigma}\big)/2$.

2. Alice chooses her POVM with operators $A_{i|a} = p_i(\mathbb{1} + \vec{a}_i \cdot \vec{\sigma})/2$. Now, she applies precisely the same steps as in the previous protocol for the given values of p_i, vectors $-\vec{a}_i$ ("-" to account for the anticorrelations in the singlet), and $\vec{\lambda}$. Namely, she chooses a suitable frame and produces her outcome i according to the conditional probabilities $p(i| - a, \vec{\lambda})$ in Eq. (3.9). (The "-" in $-a$ accounts for the "-" in $-\vec{a}_i$.)

3. Bob chooses his POVM with elements $B_{j|b}$ and performs a quantum measurement on his state $\rho_{\vec{\lambda}}$.

The distribution of the state $\rho_{\vec{\lambda}}$, namely $\frac{1}{8\pi}\big(\mathbb{1} + \vec{\lambda} \cdot \vec{\sigma}\big)$, is the same expression as the one for the parent POVM in Eq. (3.7) (up to a factor of 2 since states and measurements are normalized differently). Hence, if we sum over all the states where Alice outputs i, she samples precisely the state $p_i\big(\mathbb{1} - \vec{a}_i \cdot \vec{\sigma}/2\big)/4$ (analog to $A_{i|a}^{1/2} = p_i(\mathbb{1} + \vec{a}_i \cdot \vec{\sigma}/2)/2$ before). This matches exactly the expression in Eq. (3.21). Intuitively speaking, there is no difference for Bob's qubit if Alice performs the protocol above or performs the measurement on the actual Werner state for $\eta = 1/2$. Therefore, when Bob applies his POVM, the resulting statistics are the same in both cases. Hence, the protocol above simulates the statistics of arbitrary POVMs applied to the state $\rho_W^{1/2}$ in a local way:[2]

$$\operatorname{tr}\Big[(A_{i|a} \otimes B_{j|b})\ \rho_W^{1/2}\Big] = \frac{1}{4\pi} \int_{S_2} \mathrm{d}\vec{\lambda}\ p(i| - a, \vec{\lambda}) \operatorname{tr}\big[B_{j|b}\ \rho_{\vec{\lambda}}\big]\,. \tag{3.22}$$

This model is even a so-called local hidden state model which implies that the state $\rho_W^{1/2}$ is not steerable [3, 18, 19]. In the most fundamental steering scenario, we consider two parties, Alice and Bob, that share an entangled quantum state. The question is, whether Alice can steer Bob's state by applying a measurement on her side. However, Bob wants to exclude the possibility that his system is prepared in a well-defined state that is known to Alice. Then, Alice could just use her knowledge of the "hidden state" to pretend to Bob that she can steer his state. However, in reality, they do not share any entanglement at all. This is precisely the case in the above protocol, proving that the state ρ_W^{η} cannot demonstrate quantum steering whenever $\eta \leq 1/2$. This was known before for the restricted case of projective measurements $A_{\pm|a} = (\mathbb{1} \pm \vec{a} \cdot \vec{\sigma})/2$ [18]. When general POVMs are considered, the best model so far is the one from Barrett [85], which was shown to be a local hidden state model by Quintino et al. [145]. That model shows

[2] In the protocol, it seems that we need quantum resources to simulate the statistics. However, we can also assume that $\vec{\lambda}$ is known to both and then Bob can output j with probability $p(j|b, \vec{\lambda}) = \operatorname{tr}\big[B_{j|b}\ \rho_{\vec{\lambda}}\big]$ using his knowledge of $\vec{\lambda}$ and his measurement operators $B_{j|b}$.

that ρ_W^η cannot demonstrate steering if $\eta \leq 5/12$. Numerical evidence suggested that the same holds for all $\eta \leq 1/2$ [146, 147, 148]. Our model shows, that this is indeed the case.

On the other hand, if such a local hidden state model cannot exist we say that the state is steerable. It is known that the two-qubit Werner state can demonstrate steering whenever $\eta > 1/2$ [18]. Therefore, the bound of $\eta = 1/2$ is tight. Due to the connection between steering and joint measureability [131, 130, 132], $\eta = 1/2$ is also tight for the joint measureability problem, ensuring the optimality of our construction.

.

3.6. Conclusion

In this work, we provided tight bounds on how much white noise a measurement device can tolerate before all qubit measurements become jointly measurable. We considered the most general set of measurements (POVMs) and applied our techniques to quantum steering and Bell nonlocality. Exploiting the connection between joint measurability and steering [131, 130, 132], we found a tight local hidden state model for two-qubit Werner states of visibility $\eta = 1/2$. This solves Problem 39 on the page of Open quantum problems [149] (see also Ref. [150]) and Conjecture 1 of Ref. [147]. An important direction for further research is the generalization to higher dimensional systems [151, 152].

Acknowledgments

Most importantly, I acknowledge Marco Túlio Quintino for important discussions and for introducing the problem to me. Furthermore, I acknowledge Haggai Nuchi for correspondence about the Knaster-type theorems and an anonymous referee for very useful suggestions. This research was funded in whole, or in part, by the Austrian Science Fund (FWF) through BeyondC with Grant-DOI 10.55776/F71.

Note added: At the very last stage of this work, we became aware of the work by Yujie Zhang and Eric Chitambar [153] that proves the same results with a different approach.

3.A. A note on the non-constructive nature of the protocol

Here, we would like to provide some additional information about the non-constructive nature of our protocol. We stress again that the theorem of Hausel, Makai, and Szűcs [134] only implies that a suitable coordinate frame exists but does not imply how to find one. However, in some cases, we can explicitly find a frame.

Consider for instance the important special case of a POVM with only two outcomes which corresponds to a projective measurement. In that case, we have $p_1 = p_2 = 1$ and $\vec{a}_2 = -\vec{a}_1$, hence $A_{1|a}^{1/2} = (\mathbb{1} + \vec{a}_1 \cdot \vec{\sigma}/2)/2$ and $A_{2|a}^{1/2} = (\mathbb{1} - \vec{a}_1 \cdot \vec{\sigma}/2)/2$. We can express the function $f_a(\vec{x})$ as $f_a(\vec{x}) = \Theta(\vec{x} \cdot \vec{a}_1) + \Theta(-\vec{x} \cdot \vec{a}_1) = |\vec{x} \cdot \vec{a}_1|$. To find a suitable frame, we can choose the x'-axis to be aligned with $\vec{a}_{1/2}$, while the y'- and z'-axis are orthogonal to $\vec{a}_1$. In this way, $\vec{x}' = \vec{a}_1$ and direct calculation shows that $f_a(\vec{v}_{s_x s_y s_z}) = 1$ for all eight vertices $\vec{v}_{s_x s_y s_z} = s_x \vec{x}' + s_y \vec{y}' + s_z \vec{z}'$ as required. In addition, note that $\Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a}_1) = 1$ if $s_x = +1$ and $\Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a}_1) = 0$ if $s_x = -1$. Therefore, $\alpha_1 = \alpha_2 = 0$ and the conditional probabilities translate precisely to $p(1|a, \vec{\lambda}) = 1$ if $\vec{\lambda} \cdot \vec{a}_1 \geq 0$ and $p(1|a, \vec{\lambda}) = 0$ if $\vec{\lambda} \cdot \vec{a}_1 < 0$ (and the analog expression for $i = 2$). See Fig. 3.3 for an illustration.

The choice of the frame is unique up to an arbitrary rotation around the x'-axis (and a relabelling of the axes). To see this, note that the angle α between $\vec{a}_{1/2}$ and each cube vertex $\vec{v}_{s_x s_y s_z}$ must be at least $\alpha \geq \cos^{-1}(1/\sqrt{3})$ since $|\vec{a}_{1/2}| = 1$, $|\vec{v}_{s_x s_y s_z}| = \sqrt{3}$ and $f_a(\vec{v}_{s_x s_y s_z}) = |\vec{v}_{s_x s_y s_z} \cdot \vec{a}_1| = |\vec{v}_{s_x s_y s_z}| \cdot |\vec{a}_1| \cdot \cos(\alpha)$. Geometrically, this defines the cube uniquely up to a rotation around $\vec{a}_1$. Note, however, that such a rotation would not change the conditional probabilities $p(i|a, \vec{\lambda})$ in the end. It is worth pointing out, that this construction becomes equivalent to the one of Werner [17], which is known to be a tight local hidden state model for projective measurements [18] (and therefore also tight for the problem of joint measurability due to the close connection of these two fields [130, 131, 132]).

It turns out that for the case of three outcome POVMs, we can also construct a suitable coordinate frame without relying on the theorem of Hausel, Makai, and Szűcs [134] but only on the intermediate value theorem for continuous functions (see Appendix 3.E). For general POVMs, we want to point out that a suitable frame is computationally easy to find for many cases. For instance, we can parametrize a rotation by its three Euler angles. When we discretize the three angles into equally spaced values, we can search through many possible rotations and calculate the functional values for the corresponding cube. If we find a cube, for which all of these eight values are smaller than one, we have found a suitable frame. We provide a MATLAB code for this simple algorithm via GITHUB [154]. It turns out that even this brute-force method finds a suitable frame for most POVMs almost immediately. (However, more sophisticated algorithms, are likely to perform even better.)

We did some numerical simulations with random POVMs. For that, we generate random points on the sphere $\vec{a}_i \in S_2$ and find p_i by solving $\sum_i p_i \cdot \vec{a}_i = \vec{0}$ and $\sum_i p_i = 2$. Then we use our algorithm to find a suitable frame. These numerical simulations strongly suggest that it is the hardest to find a frame if all directions are almost collinear ($|\vec{a}_i \cdot \vec{a}_j| \approx |\vec{a}_i| \cdot |\vec{a}_j|$ for all pairs i, j). Note that, the two outcome POVMs from above are precisely of that

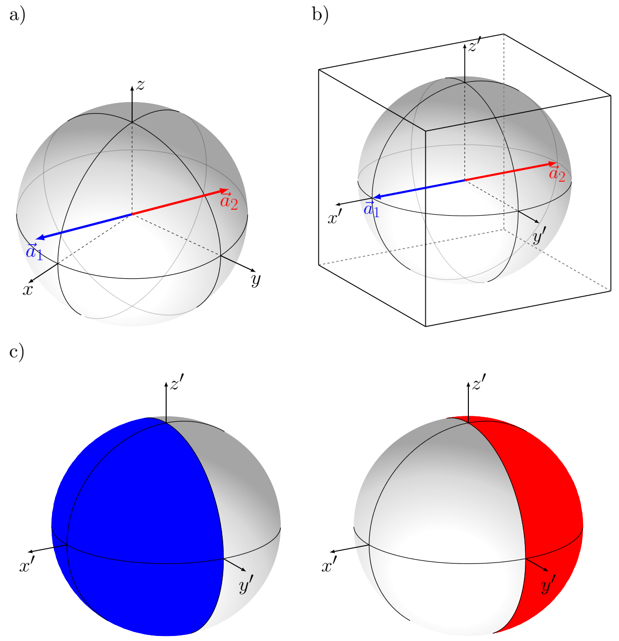


Figure 3.3.: Construction for the two-outcome POVM with operators $A^{1/2}_{i|a} = (\mathbb{1} + \vec{a}_i \cdot \vec{\sigma}/2)/2$ (with $\vec{a}_2 = -\vec{a}_1$): **a)** Here, $\vec{a}_1$ can be an arbitrary direction in the Bloch sphere. **b)** We can choose the rotated frame such that the x'-axis is aligned with $\vec{a}_1$. We also show the corresponding cube here. **c)** The conditional probabilities $p(i|a, \vec{\lambda})$ reduce precisely to $p(1|a, \vec{\lambda}) = 1$ if $\vec{\lambda} \cdot \vec{a}_1 \geq 0$ and $p(1|a, \vec{\lambda}) = 0$ if $\vec{\lambda} \cdot \vec{a}_1 < 0$ as indicated with the two colors. Hence, if the outcome $\vec{\lambda}$ of the parent POVM lies in the hemisphere centered around $\vec{a}_1$ (blue region) the outcome is always $i = 1$ and if it lies in the hemisphere centered around $\vec{a}_2$ (red region), the outcome will be $i = 2$.

form. However, even in these cases, a frame was always found in which the largest of the eight values $f(\vec{v}_{s_x s_y s_z})$ is only slightly larger than 1 and this value can be further decreased by further discretizing the Euler angles. There is an intuitive explanation for this effect. For the simulation of a given POVM, it is always advantageous if a given $\vec{\lambda}$ is mapped to an outcome i that is close, meaning that the angle between $\vec{\lambda}$ and $\vec{a}_i$ is small. Consider for instance the case of a projective measurement discussed before (or a POVM with almost collinear vectors $\vec{a}_i$). If $\vec{\lambda}$ is (almost) orthogonal to $\vec{a}_{1/2}$, the outcome of the parent measurement $\vec{\lambda}$ is (almost) uncorrelated to the $\vec{a}_{1/2}$ measurement but it still has to be mapped to either $\vec{a}_1$ or $\vec{a}_2$.

Contrary to that, for a POVM with more outcomes $\vec{a}_i$ spread over the Bloch sphere (like the symmetric, informationally complete (SIC) POVM in Fig. 3.2), there are more options a given $\vec{\lambda}$ can be mapped to. Roughly speaking it is then more likely to find a measurement outcome $\vec{a}_i$ that is highly correlated with the actual measurement outcome of the parent POVM $\vec{\lambda}$. Based on this intuition, it is reasonable to expect that these POVMs are easier to simulate. In our construction, this expresses itself in the fact that for these POVMs many different coordinate frames are suitable, and therefore several different simulations for such a POVM and $\eta = 1/2$ exist. We can even prove, that for the case of the four-outcome SIC-POVM [133], any rotation can be chosen (see Appendix 3.E). On the contrary, $\eta = 1/2$ is known to be tight for the special case of two-outcome POVMs [18], and therefore only very particular coordinate frames are possible (similar for collinear POVMs).

Note also, that we do not exclude the possibility that for certain POVMs better constructions with $\eta > 1/2$ exist. For instance, SIC-POVMs are by definition very symmetric and one would expect that a symmetric model gives an even better bound $\eta > 1/2$ (e.g., one can map $\vec{\lambda}$ to the closest outcome $\vec{a}_i$ of the SIC-POVM). However, in this work, we are merely concerned with finding one construction that works for all POVMs and $\eta = 1/2$. Hence, it is more important for our approach to recover the hemisphere construction of Fig. 3.3 (which is known to be tight for projective measurements) than to maintain the symmetry of a given POVM.

3.B. A helpful lemma

Lemma 3.1. *Given the eight vectors $\vec{v}_{s_x s_y s_z}$ forming a cube of sidelength two centered at the origin of the Bloch sphere and an arbitrary vector $\vec{a} \in \mathbb{R}^3$. In addition, the function $\Theta(x)$ (for $x \in \mathbb{R}$) is defined as $\Theta(x) := x$ if $x \geq 0$ and $\Theta(x) := 0$ if $x < 0$ (equivalent: $\Theta(x) := (|x| + x)/2$). We prove the following properties:*

$$(1) \quad \sum_{s_x,s_y,s_z=\pm 1} |\vec{v}_{s_x s_y s_z} \cdot \vec{a}| \leq 8 \cdot |\vec{a}| \tag{3.23}$$

$$(2) \quad \sum_{s_x,s_y,s_z=\pm 1} \Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a}) \leq 4 \cdot |\vec{a}| \tag{3.24}$$

$$(3) \quad \sum_{s_x,s_y,s_z=\pm 1} (\vec{v}_{s_x s_y s_z} \cdot \vec{a})\ \vec{v}_{s_x s_y s_z} = 8 \cdot \vec{a} \tag{3.25}$$

$$(4) \quad \sum_{s_x,s_y,s_z=\pm 1} \Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a})\ \vec{v}_{s_x s_y s_z} = 4 \cdot \vec{a} \tag{3.26}$$

Proof. Note that all four statements remain the same under a rotation of the coordinate system. Hence, it is sufficient to prove them in the rotated frame $\vec{x}'$, $\vec{y}'$, $\vec{z}'$ in which $\vec{v}_{s_x s_y s_z}$ has the coordinates $\vec{v}'_{s_x s_y s_z} = (s_x, s_y, s_z)^T$ where $s_x, s_y, s_z \in \{+1, -1\}$. For ease of notation, we write all vectors in the proof without the prime.

(1) We apply the Cauchy-Schwarz inequality to the following two eight-dimensional

vectors:

$$\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}\cdot\begin{pmatrix}|\vec{v}_{+++}\cdot\vec{a}|\\|\vec{v}_{++-}\cdot\vec{a}|\\\vdots\\|\vec{v}_{---}\cdot\vec{a}|\end{pmatrix}\leq\left|\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}\right|\cdot\left|\begin{pmatrix}|\vec{v}_{+++}\cdot\vec{a}|\\|\vec{v}_{++-}\cdot\vec{a}|\\\vdots\\|\vec{v}_{---}\cdot\vec{a}|\end{pmatrix}\right| \tag{3.27}$$

$$\sum_{s_x,s_y,s_z=\pm1}|\vec{v}_{s_xs_ys_z}\cdot\vec{a}|\leq\sqrt{8}\cdot\sqrt{\sum_{s_x,s_y,s_z=\pm1}|\vec{v}_{s_xs_ys_z}\cdot\vec{a}|^2} \tag{3.28}$$

Now we rewrite the right-hand side. Here we denote $\vec{a}=(a_x,a_y,a_z)^T$:

$$\sqrt{\sum_{s_x,s_y,s_z=\pm1}|\vec{v}_{s_xs_ys_z}\cdot\vec{a}|^2}=\sqrt{\sum_{s_x,s_y,s_z=\pm1}(\vec{v}_{s_xs_ys_z}\cdot\vec{a})^2} \tag{3.29}$$

$$=\sqrt{(a_x+a_y+a_z)^2+(a_x+a_y-a_z)^2+...+(-a_x-a_y-a_z)^2} \tag{3.30}$$

$$=\sqrt{8\ a_x^2+8\ a_y^2+8\ a_z^2} \tag{3.31}$$

$$=\sqrt{8}\cdot\sqrt{a_x^2+a_y^2+a_z^2} \tag{3.32}$$

$$=\sqrt{8}\cdot|\vec{a}| \tag{3.33}$$

Note, that in the third line, all terms of the form $2a_xa_y$, $2a_xa_z$ or $2a_ya_z$ cancel each other out since each of these terms appear four times with a plus sign and four times with a minus sign. In total, we obtain the desired inequality:

$$\sum_{s_x,s_y,s_z=\pm1}|\vec{v}_{s_xs_ys_z}\cdot\vec{a}|\leq 8\cdot|\vec{a}|\,. \tag{3.34}$$

(2) The second inequality is a consequence of the first one.

$$\sum_{s_x,s_y,s_z=\pm1}\Theta(\vec{a}\cdot\vec{v}_{s_xs_ys_z})=\frac{1}{2}\sum_{s_x,s_y,s_z=\pm1}|\vec{a}\cdot\vec{v}_{s_xs_ys_z}|\leq\frac{8}{2}|\vec{a}|=4|\vec{a}|\,. \tag{3.35}$$

Here, we used that:

$$\Theta(\vec{a}\cdot\vec{v}_{s_xs_ys_z})+\Theta(\vec{a}\cdot\vec{v}_{-s_x-s_y-s_z})=|\vec{a}\cdot\vec{v}_{s_xs_ys_z}|=\frac{1}{2}(|\vec{a}\cdot\vec{v}_{s_xs_ys_z}|+|\vec{a}\cdot\vec{v}_{-s_x-s_y-s_z}|)\,, \tag{3.36}$$

which follows from $\Theta(x)+\Theta(-x)=|x|$ and $\vec{a}\cdot\vec{v}_{s_xs_ys_z}=-\vec{a}\cdot\vec{v}_{-s_x-s_y-s_z}$ (as a consequence of $\vec{v}_{-s_x-s_y-s_z}=-\vec{v}_{s_xs_ys_z}$).

(3) This property is a rather straightforward calculation. If we denote $\vec{a} = (a_x, a_y, a_z)^T$ and remember that $\vec{v}_{s_x s_y s_z} = (s_x, s_y, s_z)^T$, we obtain:

$$\sum_{s_x,s_y,s_z=\pm1} (\vec{v}_{s_x s_y s_z} \cdot \vec{a})\, \vec{v}_{s_x s_y s_z} \tag{3.37}$$

$$= \left((a_x + a_y + a_z) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + (a_x + a_y - a_z) \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} + ... + (-a_x - a_y - a_z) \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \right) \tag{3.38}$$

$$= \begin{pmatrix} 8a_x \\ 8a_y \\ 8a_z \end{pmatrix} = 8 \cdot \vec{a}\,. \tag{3.39}$$

(4) For this, we note that $\vec{v}_{-s_x - s_y - s_z} = (-s_x, -s_y, -s_z)^T = -(s_x, s_y, s_z)^T = -\vec{v}_{s_x s_y s_z}$ and therefore $\vec{v}_{-s_x - s_y - s_z} \cdot \vec{a} = -\vec{v}_{s_x s_y s_z} \cdot \vec{a}$. Combining both, we obtain:

$$(\vec{v}_{-s_x-s_y-s_z} \cdot \vec{a})\, \vec{v}_{-s_x-s_y-s_z} = (-\vec{v}_{s_x s_y s_z} \cdot \vec{a})\, (-\vec{v}_{s_x s_y s_z}) = (\vec{v}_{s_x s_y s_z} \cdot \vec{a})\, \vec{v}_{s_x s_y s_z}\,. \tag{3.40}$$

In addition, since $\Theta(x) - \Theta(-x) = x$ and again $\vec{v}_{-s_x-s_y-s_z} = -\vec{v}_{s_x s_y s_z}$ we observe:

$$\Theta(\vec{a} \cdot \vec{v}_{s_x s_y s_z})\, \vec{v}_{s_x s_y s_z} + \Theta(\vec{a} \cdot \vec{v}_{-s_x-s_y-s_z})\, \vec{v}_{-s_x-s_y-s_z} \tag{3.41}$$

$$=\Theta(\vec{a} \cdot \vec{v}_{s_x s_y s_z})\, \vec{v}_{s_x s_y s_z} - \Theta(-\vec{a} \cdot \vec{v}_{s_x s_y s_z})\, \vec{v}_{s_x s_y s_z} \tag{3.42}$$

$$=(\Theta(\vec{a} \cdot \vec{v}_{s_x s_y s_z}) - \Theta(-\vec{a} \cdot \vec{v}_{s_x s_y s_z}))\, \vec{v}_{s_x s_y s_z} \tag{3.43}$$

$$=(\vec{a} \cdot \vec{v}_{s_x s_y s_z})\, \vec{v}_{s_x s_y s_z} \tag{3.44}$$

$$=\frac{1}{2}((\vec{a} \cdot \vec{v}_{s_x s_y s_z})\, \vec{v}_{s_x s_y s_z} + (\vec{a} \cdot \vec{v}_{-s_x-s_y-s_z})\, \vec{v}_{-s_x-s_y-s_z}) \tag{3.45}$$

This implies together with property (3):

$$\sum_{s_x,s_y,s_z=\pm1} \Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a})\, \vec{v}_{s_x s_y s_z} = \frac{1}{2} \sum_{s_x,s_y,s_z=\pm1} (\vec{v}_{s_x s_y s_z} \cdot \vec{a})\, \vec{v}_{s_x s_y s_z} = 4 \cdot \vec{a}\,. \tag{3.46}$$

□

3.C. Properties of the function f_a

The theorem by Hausel, Makai, and Szűcs [134] states that for every real-valued and continuous function on the two-sphere S_2 that has the additional property that it is even, i.e. $f(\vec{x}) = f(-\vec{x})$, there exists an inscribed cube with all vertices lying on the sphere S_2 such that the function $f(\vec{x})$ has the same value on each vertex of that cube. We show now that these properties are fulfilled by the function $f_a(\vec{x})$. Strictly speaking, we apply the theorem not to the unit sphere but to the sphere with radius $|\vec{v}_{s_x s_y s_z}| = \sqrt{3}$. Alternatively, we can also apply the theorem to the unit sphere and the vectors $\vec{v}_{s_x s_y s_z}/\sqrt{3}$. Since the function satisfies $f_a(\gamma \cdot \vec{x}) = \gamma \cdot f_a(\vec{x})$ for $\gamma \geq 0$ (see below), the values of $f_a(\vec{v}_{s_x s_y s_z})$ coincide if and only if the values of $f_a(\vec{v}_{s_x s_y s_z}/\sqrt{3})$ coincide.

Lemma 3.2. *We consider the function* $f_a : \mathbb{R}^3 \mapsto \mathbb{R} :\ f_a(\vec{x}) = \sum_i p_i\ \Theta(\vec{x} \cdot \vec{a}_i)$*. Here,* $|\vec{a}_i| = 1\ \forall i$*,* $\sum_i p_i = 2$ *and* $\sum_i p_i\ \vec{a}_i = \vec{0}$*. In addition, the function* $\Theta(x)$ *is defined as* $\Theta(x) := x$ *if* $x \geq 0$ *and* $\Theta(x) := 0$ *if* $x < 0$ *(equivalently:* $\Theta(x) := (|x| + x)/2$*). The function* f_a *satisfies:*

$$(1)\ f_a(\vec{x}) = \frac{1}{2}\sum_i p_i\ |\vec{a}_i \cdot \vec{x}| \qquad \textit{and} \qquad (2) \sum_{s_x,s_y,s_z=\pm 1} f_a(\vec{v}_{s_xs_ys_z}) \leq 8\,. \tag{3.47}$$

Here, again the eight vectors $\vec{v}_{s_xs_ys_z}$ *with* $s_x, s_y, s_z \in \{+1, -1\}$ *form the vertices of a cube with sidelength two centered at the origin of the Bloch sphere. From property (1) we can conclude furthermore, that* $f_a(\vec{x})$ *is a sum of continuous functions and therefore continuous (also when restricted to the sphere* S_2*) that satisfies* $f_a(-\vec{x}) = f_a(\vec{x})$ *or more generally* $f_a(\gamma \cdot \vec{x}) = |\gamma| \cdot f_a(\vec{x})$ *for every* $\gamma \in \mathbb{R}$*.*

Proof. (1) We show that the function $f_a(\vec{x})$ can be rewritten as follows:

$$f_a(\vec{x}) = \sum_i p_i\ \Theta(\vec{a}_i \cdot \vec{x}) = \frac{1}{2}\sum_i p_i\ |\vec{a}_i \cdot \vec{x}|\,. \tag{3.48}$$

We want to remark that exactly the same property appears also in Lemma 1.2 and we restate the proof here: We prove first that $\sum_i p_i\ \Theta(\vec{a}_i \cdot \vec{x}) = \sum_i p_i\ \Theta(-\vec{a}_i \cdot \vec{x})$. Here, we use that $x = \Theta(x) - \Theta(-x)$ (for all $x \in \mathbb{R}$):

$$\vec{0} = \sum_i p_i\ \vec{a}_i \implies 0 = \vec{0} \cdot \vec{x} = \sum_i p_i\ \vec{a}_i \cdot \vec{x} = \sum_i p_i\ (\Theta(\vec{a}_i \cdot \vec{x}) - \Theta(-\vec{a}_i \cdot \vec{x})) \tag{3.49}$$

$$= \sum_i p_i\ \Theta(\vec{a}_i \cdot \vec{x}) - \sum_i p_i\ \Theta(-\vec{a}_i \cdot \vec{x})\,. \tag{3.50}$$

In the second step, we use this observation and $|x| = \Theta(x) + \Theta(-x)$ (for all $x \in \mathbb{R}$) to calculate:

$$\sum_i p_i\ |\vec{a}_i \cdot \vec{x}| = \sum_i p_i\ (\Theta(\vec{a}_i \cdot \vec{x}) + \Theta(-\vec{a}_i \cdot \vec{x})) \tag{3.51}$$

$$= \sum_i p_i\ \Theta(\vec{a}_i \cdot \vec{x}) + \sum_i p_i\ \Theta(-\vec{a}_i \cdot \vec{x}) = 2\sum_i p_i\ \Theta(\vec{a}_i \cdot \vec{x})\,. \tag{3.52}$$

(2) This is a direct consequence of property (1) in Lemma 3.1 we proved above together with multiplying by p_i and taking the sum over all i:

$$2 \sum_{s_x,s_y,s_z=\pm 1} f_a(\vec{v}_{s_xs_ys_z}) = \sum_{s_x,s_y,s_z=\pm 1} \sum_i p_i\ |\vec{v}_{s_xs_ys_z} \cdot \vec{a}_i| \tag{3.53}$$

$$= \sum_i p_i \sum_{s_x,s_y,s_z=\pm 1} |\vec{v}_{s_xs_ys_z} \cdot \vec{a}_i| \leq \sum_i p_i \cdot 8 \cdot |\vec{a}_i| = 16\,. \tag{3.54}$$

Note that $|\vec{a}_i| = 1$ for all i and $\sum_i p_i = 2$.

□

3.D. Proof of the protocol

In this section, we show that the protocol indeed simulates the noisy POVM with elements $A_{i|a}^{1/2}$. We have to show that

$$\int_{S_2} \mathrm{d}\vec{\lambda}\; p(i|a,\vec{\lambda}) G_{\vec{\lambda}} = \frac{p_i}{2}\left(\mathbb{1} + \frac{\vec{a}_i \cdot \vec{\sigma}}{2}\right) = A_{i|a}^{1/2}\,. \tag{3.55}$$

We also restate the definitions from the main text here:

$$p(i|a,\vec{\lambda}) := p_i \cdot \Theta(\vec{a}_i \cdot \vec{v}_{s_x s_y s_z}) + \frac{(1 - f_a(\vec{v}_{s_x s_y s_z}))\alpha_i}{\sum_i \alpha_i}\,, \tag{3.56}$$

$$\alpha_i := \frac{p_i}{2}\left(1 - \frac{1}{4}\sum_{s_x,s_y,s_z=\pm 1} \Theta(\vec{a}_i \cdot \vec{v}_{s_x s_y s_z})\right)\,. \tag{3.57}$$

It is important to recognize that the function $p(i|a,\vec{\lambda})$ is constant in each octant of the rotated coordinate frame x', y', z' since $\vec{v}_{s_x s_y s_z}$ is given by $s_k = \operatorname{sgn}(\vec{\lambda} \cdot \vec{k}')$ for $k \in \{x,y,z\}$. Intuitively speaking, we collect all the measurement results from one octant of the rotated frame together. This coarse-graining of the parent POVM can be calculated when we integrate over all vectors $\vec{\lambda}$ in the corresponding octant. We denote this as the operator $G_{s_x s_y s_z}$ that becomes (calculation in the next Subsection 3.D.1):

$$G_{s_x s_y s_z} := \int_{S_2|\operatorname{sgn}(\vec{\lambda}\cdot\vec{k}')=s_k} \mathrm{d}\vec{\lambda}\; G_{\vec{\lambda}} = \frac{1}{16}(2 \cdot \mathbb{1} + \vec{v}_{s_x s_y s_z} \cdot \vec{\sigma})\,. \tag{3.58}$$

This operator behaves like a noisy measurement in the direction of the corresponding vector $\vec{v}_{s_x s_y s_z}$. Using this, the above integration reduces to:

$$\int_{S_2} \mathrm{d}\vec{\lambda}\; p(i|a,\vec{\lambda})\; G_{\vec{\lambda}} = \sum_{s_x,s_y,s_z=\pm 1} p(i|a,\vec{\lambda})\; G_{s_x s_y s_z}\,, \tag{3.59}$$

where we again note that the conditional probabilities $p(i|a,\vec{\lambda})$ only depend on the signs of $\vec{\lambda}$ in the rotated frame. Now we can evaluate the right-hand side of this equation. We obtain:

$$\sum_{s_x,s_y,s_z=\pm 1} p(i|a,\vec{\lambda})\; G_{s_x s_y s_z} \tag{3.60}$$

$$= \sum_{s_x,s_y,s_z=\pm 1} p_i\; \Theta(\vec{a}_i \cdot \vec{v}_{s_x s_y s_z})\; G_{s_x s_y s_z} + \sum_{s_x,s_y,s_z=\pm 1} \frac{(1 - f_a(\vec{v}_{s_x s_y s_z}))\alpha_i}{\sum_i \alpha_i}\; G_{s_x s_y s_z}\,. \tag{3.61}$$

We evaluate the first term first:

$$\sum_{s_x,s_y,s_z=\pm1} p_i\ \Theta(\vec{a}_i\cdot\vec{v}_{s_xs_ys_z})\ G_{s_xs_ys_z} \tag{3.62}$$

$$= \frac{1}{8}\sum_{s_x,s_y,s_z=\pm1} p_i\ \Theta(\vec{a}_i\cdot\vec{v}_{s_xs_ys_z})\ \mathbb{1} + \frac{1}{16}\sum_{s_x,s_y,s_z=\pm1} p_i\ \Theta(\vec{a}_i\cdot\vec{v}_{s_xs_ys_z})\vec{v}_{s_xs_ys_z}\cdot\vec{\sigma} \tag{3.63}$$

$$= \frac{1}{8}\sum_{s_x,s_y,s_z=\pm1} p_i\ \Theta(\vec{a}_i\cdot\vec{v}_{s_xs_ys_z})\ \mathbb{1} + \frac{1}{4}p_i\ \vec{a}\cdot\vec{\sigma} \tag{3.64}$$

$$= \frac{p_i}{2}\left(\mathbb{1} + \frac{\vec{a}_i\cdot\vec{\sigma}}{2}\right) - \alpha_i\mathbb{1} = A^{1/2}_{i|a} - \alpha_i\mathbb{1}\,. \tag{3.65}$$

Here we used property (4) in Lemma 3.1 and the definition of α_i to rewrite the expression into the desired form. The second term becomes:

$$\sum_{s_x,s_y,s_z=\pm1} \frac{(1-f_a(\vec{v}_{s_xs_ys_z}))\alpha_i}{\sum_i\alpha_i}\ G_{s_xs_ys_z} = \frac{\sum_{s_x,s_y,s_z=\pm1}(1-f_a(\vec{v}_{s_xs_ys_z}))\alpha_i}{8\sum_i\alpha_i}\ \mathbb{1} = \alpha_i\mathbb{1}\,. \tag{3.66}$$

To see this, we note that the coefficient in front of $G_{s_xs_ys_z}$ and $G_{-s_x-s_y-s_z}$ are the same since $f_a(\vec{v}_{s_xs_ys_z}) = f_a(-\vec{v}_{s_xs_ys_z}) = f_a(\vec{v}_{-s_x-s_y-s_z})$ and the α_i do not depend on s_k. In addition, $G_{s_xs_ys_z}+G_{-s_x-s_y-s_z} = \mathbb{1}/4 = \mathbb{1}/8+\mathbb{1}/8$ which allows us to replace each $G_{s_xs_ys_z}$ by $\mathbb{1}/8$ in the first step. Furthermore, we observe that:

$$8\sum_i\alpha_i = 8\sum_i\frac{p_i}{2}\left(1-\frac{1}{4}\sum_{s_x,s_y,s_z=\pm1}\Theta(\vec{a}_i\cdot\vec{v}_{s_xs_ys_z})\right) \tag{3.67}$$

$$= 8 - \sum_i\sum_{s_x,s_y,s_z=\pm1} p_i\ \Theta(\vec{a}_i\cdot\vec{v}_{s_xs_ys_z}) \tag{3.68}$$

$$= \sum_{s_x,s_y,s_z=\pm1}(1-f_a(\vec{v}_{s_xs_ys_z}))\,, \tag{3.69}$$

which explains the last step in Eq. (3.66). Hence, the sum of the two terms in Eq. (3.61) is exactly $A^{1/2}_{i|a}$ as required.

3.D.1. Dividing the sphere into the eight octants

We divided the sphere into eight regions, according to the eight octants defined by the signs in the rotated coordinate system. Our goal here is to show that when we integrate all $\vec{\lambda}$ in one octant, the result is precisely $G_{s_xs_ys_z} = (2\cdot\mathbb{1}+\vec{v}_{s_xs_ys_z}\cdot\vec{\sigma})/16$, where $\vec{v}_{s_xs_ys_z}$ is the vertex of the cube that points to the midpoint of the octant. We do the calculation first in the standard coordinate frame (without rotation) for the vector $\vec{v}_{+++} = (+1,+1,+1)^T$ and show that $G_{+++} = (2\cdot\mathbb{1}+\vec{v}_{+++}\cdot\vec{\sigma})/16$. Here we use spherical coordinates $\vec{\lambda} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ and integrate the parent POVM $G_{\vec{\lambda}} = \frac{1}{4\pi}(\mathbb{1} + \vec{\lambda}\cdot\vec{\sigma})$ over all $\vec{\lambda}$ in that octant. Hence, all $\vec{\lambda}$ that have only positive components. This is true if and only if $0 \le \phi \le \pi/2$ as well as $0 \le \theta \le \pi/2$:

$$G_{+++} = \frac{1}{4\pi}\int_0^{\pi/2} \mathrm{d}\theta \int_0^{\pi/2} \mathrm{d}\phi\ (\mathbb{1} + \sigma_x \sin\theta\cos\phi + \sigma_y \sin\theta\sin\phi + \sigma_z\cos\theta)\sin\theta \tag{3.70}$$

$$= \frac{1}{8}\mathbb{1} + \frac{1}{16}\left(\sigma_x + \sigma_y + \sigma_z\right)\ . \tag{3.71}$$

However, in general, the integration is in a rotated frame. Nevertheless, the shape of an octant is always the same, only the position is rotated. Hence, by symmetry arguments, we can conclude that $G_{s_x s_y s_z} = (2\cdot\mathbb{1} + \vec{v}_{s_x s_y s_z}\cdot\vec{\sigma})/16$ holds for a general octant corresponding to the cube vertex $\vec{v}_{s_x s_y s_z}$.

3.E. Special cases

Here, we discuss some special cases and provide more illustrations. The case of two-outcome POVMs is already discussed in Appendix 3.A.

3.E.1. Three-outcome measurements

Suppose that all the vectors $\vec{a}_i$ lie in a plane. In particular, this is true if the POVM has only three outcomes since $p_1\vec{a}_1 + p_2\vec{a}_2 + p_3\vec{a}_3 = \vec{0}$ can only be satisfied if all three vectors lie in the same plane. In that case, we can choose the basis such that z' is orthogonal to the plane in which the vectors lie. With that choice of coordinate frame, we can observe that $f_a(\vec{v}_{s_x s_y +}) = f_a(\vec{v}_{s_x s_y -})$ since $\vec{a}_i\cdot\vec{z}' = 0$ and therefore the z'-component of $\vec{v}_{s_x s_y s_z}$ does not affect the value of $f_a(\vec{v}_{s_x s_y s_z})$. Together with $f_a(\vec{v}_{s_x s_y s_z}) = f_a(\vec{v}_{-s_x -s_y -s_z})$ (note that $\vec{v}_{s_x s_y s_z} = -\vec{v}_{-s_x -s_y -s_z}$ and $f_a(\vec{x}) = f_a(-\vec{x})$), we can denote $C_1 := f_a(\vec{v}_{+++}) = f_a(\vec{v}_{++-}) = f_a(\vec{v}_{--+}) = f_a(\vec{v}_{---})$ and $C_2 := f_a(\vec{v}_{+-+}) = f_a(\vec{v}_{+--}) = f_a(\vec{v}_{-++}) = f_a(\vec{v}_{-+-})$. As a consequence of property (2) in Lemma 3.2, we obtain $C_1 + C_2 \le 2$. Now we can show that there always exists a rotation around the z'-axis such that both values C_1 and C_2 are smaller or equal to one. If we fix at the beginning a coordinate frame where both values are smaller than one, we can use precisely that frame. On the other hand, if one value (suppose C_1) is above one, the other one (C_2) is smaller than one. Now we rotate the coordinate axes x' and y' around the z'-axis. If we rotate by 90 degrees, we map the vector $\vec{v}_{+-+}$ to the vector $\vec{v}_{+++}$ and therefore in the rotated coordinate frame we obtain $C_1^* = f_a(\vec{v}^*_{+++}) = f_a(\vec{v}_{+-+}) = C_2 \le 1$. By the intermediate value theorem and since f_a is continuous, there is a rotation (with less than 90 degrees) such that $C_1 = f_a(\vec{v}^*_{+++}) = 1$ which implies that $C_2^* = f_a(\vec{v}^*_{+-+}) \le 1$ since $C_1^* + C_2^* \le 2$ holds for each of these coordinate frames. In this way, we can construct a suitable coordinate system without relying on the theorem of Ref. [134] but only on the intermediate value theorem.

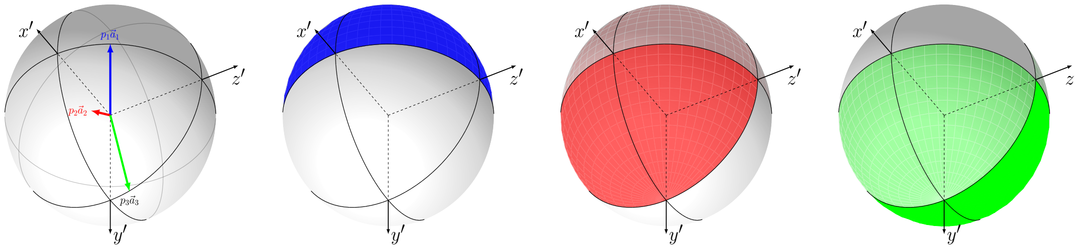


Figure 3.4.: An illustration of $p(i|a,\vec{\lambda})$ for a three-outcome POVM. Here the conditional probabilities do not depend on z' due to the choice of the coordinate frame. If $\vec{\lambda}$ is close to one of the colored vectors it is also more likely that this color is produced as an output.

3.E.2. SIC-POVM

We also want to give an example with a four-outcome measurement, namely a SIC-POVM. We can represent a SIC POVM as follows:

$$\vec{a}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad \vec{a}_2 = \begin{pmatrix} \sqrt{8}/3 \\ 0 \\ -1/3 \end{pmatrix} \qquad \vec{a}_3 = \begin{pmatrix} -\sqrt{2}/3 \\ \sqrt{6}/3 \\ -1/3 \end{pmatrix} \qquad \vec{a}_4 = \begin{pmatrix} -\sqrt{2}/3 \\ -\sqrt{6}/3 \\ -1/3 \end{pmatrix} \tag{3.72}$$

and the coefficients p_i are $p_1 = p_2 = p_3 = p_4 = 1/2$. It turns out, that we can use the standard coordinate frame and no rotation of the basis is necessary. In that basis, the eight vertices of the cube become $\vec{v}_{s_x s_y s_z} = (s_x, s_y, s_z)^T$ with $s_x, s_y, s_z \in \{+1, -1\}$ and we can indeed verify that $f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ for each $\vec{v}_{s_x s_y s_z}$. See the following table:

i	+ + +	+ + -	+ - +	+ - -	- + +	- + -	- - +	- - -	α_i
1 (blue)	0.5	0	0.5	0	0.5	0	0.5	0	0
2 (red)	0.305	0.638	0.305	0.638	0	0	0	0	0.014
3 (green)	0.006	0.339	0	0	0.477	0.811	0	0	0.046
4 (yelow)	0	0	0.006	0.339	0	0	0.477	0.811	0.046
$f_a(\vec{v}_{s_x s_y s_z})$	0.811	0.977	0.811	0.977	0.977	0.811	0.977	0.811	

Table 3.1.: Here we represent the functional values of $p_i\ \Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a}_i)$ for the chosen SIC-POVM (note that $p_i = 1/2$ for every i). The last row gives the values for the function $f_a(\vec{v}_{s_x s_y s_z}) = \sum_i p_i\ \Theta(\vec{v}_{s_x s_y s_z} \cdot \vec{a}_i)$ which is obtained by taking the sum over all i. We also calculate every α_i in the last column.

Every orthonormal basis is suitable for SIC-POVMs

For the construction of the SIC-POVM above, we simply used the standard coordinate frame. However, it turns out that any other choice of orthonormal basis (or any rotation

i	+ + +	+ + -	+ - +	+ - -	- + +	- + -	- - +	- - -	$\sum$
1 (blue)	0.5	0	0.5	0	0.5	0	0.5	0	2
2 (red)	0.330	0.641	0.330	0.641	0.003	0.026	0.003	0.026	2
3 (green)	0.088	0.349	0.082	0.010	0.487	0.893	0.010	0.082	2
4 (yellow)	0.082	0.010	0.088	0.349	0.010	0.082	0.487	0.893	2
	1	1	1	1	1	1	1	1	

Table 3.2.: Here we represent the conditional probabilities $p(i|a,\vec{\lambda})$ given the octant in which $\vec{\lambda} = (\lambda_x, \lambda_y, \lambda_z)^T$ lies (denoted as $s_x s_y s_z$ where $s_k = \mathrm{sgn}(\lambda_k)$ for $k = x, y, z$). Intuitively speaking, they are the same values as in the table above but we fill the rest with noise. The last row is obtained by taking the sum over all i which shows that the probabilities sum to one. A short calculation can also directly verify that $\sum_{s_x,s_y,s_z=\pm 1} p(i|a,\vec{\lambda})\; G_{s_x s_y s_z} = A_{i|a}^{1/2}$.

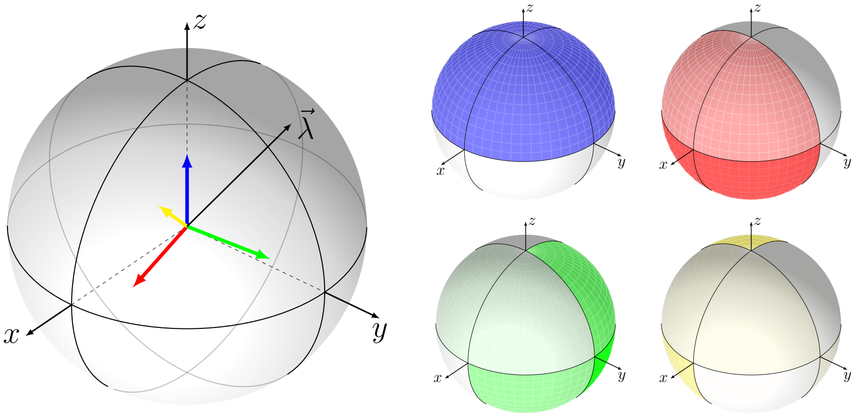


Figure 3.5.: (similar figure as in the main text) The coloured arrows denote the vectors $p_i \cdot \vec{a}_i$ according to $p_i = 1/2$ and the vectors $\vec{a}_i$ given above. The right part of the figure represents the conditional probabilities given in the table above. Here, $\vec{\lambda}$ lies in the octant that corresponds to "-++" ($s_x = -1$, $s_y = s_z = +1$) the outcome is $i = 1$ (blue) with $p(1|a,\vec{\lambda}) = 0.5$, $i = 2$ (red) with $p(2|a,\vec{\lambda}) = 0.003$, $i = 3$ (green) with $p(3|a,\vec{\lambda}) = 0.487$, and $i = 4$ (yellow) with $p(4|a,\vec{\lambda}) = 0.01$. Therefore, the outcome is most likely to be either $i = 1$ or $i = 3$.

of the cube) would be equally valid. Therefore, in a different coordinate frame, the functions for the conditional probabilities would change but a simulation is still possible. Indeed, we can prove that $f_a(\vec{x}) \leq 1$ for every vector $\vec{x}$ with $|\vec{x}| \leq \sqrt{3}$. Therefore for every rotation of the cube, the vertices $\vec{v}_{s_x s_y s_z}$ satisfy $f_a(\vec{v}_{s_x s_y s_z}) \leq 1$ since $|\vec{v}_{s_x s_y s_z}| = \sqrt{3}$. For the SIC-POVM the function becomes the following:

$$f_a(\vec{x}) = \sum_i p_i\ \Theta(\vec{x}\cdot\vec{a}_i) = \frac{\Theta(\vec{x}\cdot\vec{a}_1)+\Theta(\vec{x}\cdot\vec{a}_2)+\Theta(\vec{x}\cdot\vec{a}_3)+\Theta(\vec{x}\cdot\vec{a}_4)}{2} \tag{3.73}$$

Depending on the region where $\vec{x}$ lies, some of the terms $\vec{x}\cdot\vec{a}_i$ are positive and some of them are negative. We show that in any case, $f_a(\vec{x}) \leq 1$ if $|\vec{x}| \leq \sqrt{3}$. Suppose only the term $\vec{x}\cdot\vec{a}_1$ is positive and the remaining three terms $\vec{x}\cdot\vec{a}_i$ are negative (this happens for instance if $\vec{x} = (x,0,0)$). If this is the case, the function becomes

$$f_a(\vec{x}) = \frac{\Theta(\vec{x}\cdot\vec{a}_1)}{2} = \frac{\vec{x}\cdot\vec{a}_1}{2} \leq \frac{|\vec{x}|\cdot|\vec{a}_1|}{2} = \frac{|\vec{x}|}{2}, \tag{3.74}$$

where we used the Cauchy-Schwarz inequality and $|\vec{a}_1| = 1$. The same argument holds if one of the other terms is positive and the remaining three are negative. Now consider, that the two terms $\vec{x}\cdot\vec{a}_1$ and $\vec{x}\cdot\vec{a}_2$ are positive, and the remaining two are negative. Then the function becomes:

$$f_a(\vec{x}) = \frac{\Theta(\vec{x}\cdot\vec{a}_1)+\Theta(\vec{x}\cdot\vec{a}_2)}{2} = \frac{\vec{x}\cdot\vec{a}_1+\vec{x}\cdot\vec{a}_2}{2} = \frac{\vec{x}\cdot(\vec{a}_1+\vec{a}_2)}{2} \tag{3.75}$$

$$\leq \frac{|\vec{x}|\cdot|\vec{a}_1+\vec{a}_2|}{2} = \frac{|\vec{x}|\cdot\sqrt{\frac{4}{3}}}{2} = \frac{|\vec{x}|}{\sqrt{3}}. \tag{3.76}$$

Here we used again the Cauchy-Schwarz inequality and note that $\vec{a}_1+\vec{a}_2 = (\sqrt{8}/3, 0, 2/3)^T$ (hence $|\vec{a}_1+\vec{a}_2| = \sqrt{4/3}$). Due to symmetry reasons (or by a similar calculation), the same applies to any other combination of these four terms, in which exactly two of them are positive.

In the case where three terms are positive, we obtain similarly (note that $\vec{a}_1+\vec{a}_2+\vec{a}_3+\vec{a}_4 = \vec{0}$):

$$f_a(\vec{x}) = \frac{\Theta(\vec{x}\cdot\vec{a}_1)+\Theta(\vec{x}\cdot\vec{a}_2)+\Theta(\vec{x}\cdot\vec{a}_3)}{2} = \frac{\vec{x}\cdot(\vec{a}_1+\vec{a}_2+\vec{a}_3)}{2} = \frac{\vec{x}\cdot(-\vec{a}_4)}{2} \tag{3.77}$$

$$\leq \frac{|\vec{x}|\cdot|\vec{a}_4|}{2} = \frac{|\vec{x}|}{2}. \tag{3.78}$$

If none of the terms is positive, the function becomes $f_a(\vec{x}) = 0$. If all of the terms are positive, the function becomes also $f_a(\vec{x}) = (\vec{x}\cdot(\vec{a}_1+\vec{a}_2+\vec{a}_3+\vec{a}_4))/2 = 0$ since $\vec{a}_1+\vec{a}_2+\vec{a}_3+\vec{a}_4 = \vec{0}$. (However, there are no vectors except $\vec{x} = \vec{0}$ where either none or all of the terms are positive due to geometric arguments.) One can see that, no matter in which case we are, for every vector with $|\vec{x}| \leq \sqrt{3}$ the function satisfies $f_a(\vec{x}) \leq 1$, and therefore every orthonormal frame (respectively, any rotation of the cube) can be chosen.

Part II.

Quantum computation with indefinite causal structures

4. Reassessing the computational advantage of quantum-controlled ordering of gates

This chapter is based on the article:

Martin J. Renner, Časlav Brukner "Reassessing the computational advantage of quantum-controlled ordering of gates," Physical Review Research 3 (4), 043012 (2021) (Received: 9 April 2021, Accepted: 13 August 2021, Published: 6 October 2021)

Contributions: Časlav supervised the work and introduced the problem to me. The results were found and the manuscript was written by myself. All technical proofs were carried out by myself.

Abstract

Research on indefinite causal structures is a rapidly evolving field that has a potential not only to make a radical revision of the classical understanding of space-time but also to achieve enhanced functionalities of quantum information processing. For example, it is known that indefinite causal structures provide exponential advantage in communication complexity when compared to causal protocols. In quantum computation, such structures can decide whether two unitary gates commute or anticommute with a single call to each gate, which is impossible with conventional (causal) quantum algorithms. A generalization of this effect to n unitary gates, originally introduced in M. Araújo et al., Phys. Rev. Lett. 113, 250402 (2014) and often called Fourier promise problem (FPP), can be solved with the quantum-n-switch and a single call to each gate, while the best known causal algorithm so far calls $O(n^2)$ gates. In this work, we show that this advantage is smaller than expected. In fact, we present a causal algorithm that solves the *only* known specific FPP with $O(n\log(n))$ queries and a causal algorithm that solves *every* FPP with $O(n\sqrt{n})$ queries. Besides the interest in such algorithms on their own, our results limit the expected advantage of indefinite causal structures for these problems.

4.1. Introduction

One of the most fundamental concepts in science is that of causality: the idea that events occur in a fixed order. It is embedded in the very structure of computation in which operations are performed one after the other. In particular, a quantum circuit is built out of wires, representing the quantum states, and boxes, representing the gates acting on these states in fixed order. However, it was suggested that the interplay between general relativity and quantum theory might require superseding such a paradigm [21, 155]. Within the last decade, quantum frameworks have been developed that enable the description of indefinite causal structures in which no well-defined global order of events exists [21, 23, 22].

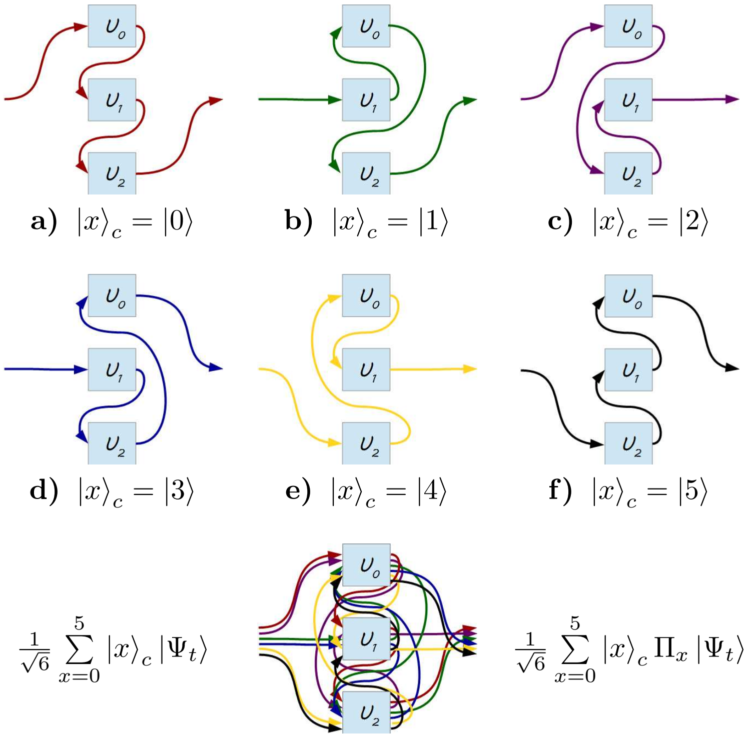
It was observed that the use of indefinite causal structures in information processing can solve certain tasks which cannot be completed by causally ordered quantum circuits [26]. They exponentially reduce the communication cost in communication complexity problems [156] and provide an advantage for channel discrimination tasks [157]. Furthermore, they can boost the rate of communication through noisy channels [158, 159, 160, 161, 162], although causal circuits can achieve the same or even better noise reduction [163, 164, 165]. The computational complexity of indefinite causal structures has been studied [166, 167] and their experimental accessibility was demonstrated in enhanced quantum photonics experiments [168, 169, 170, 171, 172, 28].

The most simple example of indefinite causality is based on the quantum switch [23]. In the quantum switch, two gates act on a target system and the order in which the two gates are applied is controlled by a qubit: if the state of the control qubit is $|0\rangle$, the gate U_0 is applied before U_1 whereas if the control qubit is in the state $|1\rangle$, the order is reversed. With this quantum-controlled ordering of gates, one can solve certain tasks more efficiently than with any conventional (causal) quantum algorithm. Specifically, one can determine whether two unitary gates commute or anticommute with a single call to each gate, while with any causal quantum algorithm, at least one gate has to be called twice [26].

A generalization of the quantum switch to an arbitrary number of gates is the quantum-n-switch. Here, depending on the state of the control system, any permutation of the n gates can be applied on the target system. In order to study the computational power of this quantum-controlled ordering of gates, a promise problem was introduced in Ref. [27]. This task, which we will call Fourier promise problem (FPP) here, can be solved with the quantum-n-switch and a single call to each gate (n queries). At the same time, it was expected that solving the same task with a causal quantum algorithm requires $O(n^2)$ queries. In a recent study, this idea is extended to other promise problems that are easier to realize experimentally [28].

In this work, we consider the solutions to the specific and general Fourier promise problems using both the quantum-n-switch and causal quantum algorithms. We find that the reduction in the query complexity using the quantum-n-switch is smaller than what was assumed so far. More precisely, we present a causal algorithm that solves the *only* known specific FPP with $O(n \log n)$ queries and further, a causal algorithm that solves *every* FPP with $O(n\sqrt{n})$ queries. This reduces the expected advantage of indefinite causal structures in solving this computational task as compared to causal circuits.

The article is structured as follows: In Section 4.2, we give an overview of the Fourier promise problem, the solution with the quantum-n-switch and the best causal algorithm that uses $O(n^2)$ queries. In Section 4.3, we derive the property that allows us to find more efficient causal algorithms and give a first example of such an algorithm in Section 4.4. The two main results of this article can be found thereafter. In Section 4.5, we present a causal algorithm that solves a specific FPP with $O(n \log n)$ queries. In Section 4.6, we give a causal algorithm that solves every FPP with $O(n\sqrt{n})$ queries.



a) $|x\rangle_c = |0\rangle$ **b)** $|x\rangle_c = |1\rangle$ **c)** $|x\rangle_c = |2\rangle$

d) $|x\rangle_c = |3\rangle$ **e)** $|x\rangle_c = |4\rangle$ **f)** $|x\rangle_c = |5\rangle$

$\frac{1}{\sqrt{6}} \sum_{x=0}^{5} |x\rangle_c |\Psi_t\rangle$ $\frac{1}{\sqrt{6}} \sum_{x=0}^{5} |x\rangle_c \Pi_x |\Psi_t\rangle$

Figure 4.1.: The quantum-3-switch: Depending on the state of the control system, the gates act on the target system in a different order. For the case of $n = 3$, each basis state of the six-dimensional control system realizes a different permutation of the gates. If the control system is initialized in a superposition, the n-switch can be used to solve Fourier promise problems. In this way, each unitary U_i is called only once.

4.2. The Fourier Promise Problem

Fourier promise problems, originally introduced in Ref. [27], are defined as follows:

Definition 4.1 (Fourier promise problem)**.** *A set of d-dimensional unitary gates $\{U_i\}_0^{n-1}$ is given. Each permutation σ_x of the n unitaries is denoted as $\Pi_x = U_{\sigma_x(n-1)}...U_{\sigma_x(1)}U_{\sigma_x(0)}$*

and labeled by a number $x \in \{0, 1, ..., n! - 1\}$. It is promised that for some value $y \in \{0, 1, ..., n! - 1\}$, the permutations satisfy the following relations:

$$\forall x \in \{0, 1, ..., n! - 1\} : \ \Pi_x = \omega^{x \cdot y} \cdot \Pi_0 \,. \tag{4.1}$$

Here, ω is defined as $\omega := e^{\frac{2\pi i}{n!}}$ and the task is to find the value y for which the above promise is satisfied.

For example, in the case of two unitaries U_0 and U_1, the two permutations $U_1 U_0$ and $U_0 U_1$ can be either labeled by $\Pi_0 = U_1 U_0$ and $\Pi_1 = U_0 U_1$ or the other way around ($\Pi_1 = U_1 U_0$ and $\Pi_0 = U_0 U_1$). While the promise for $x = 0$ is trivially satisfied, the promise for $x = 1$, namely $\Pi_1 = \omega^{1 \cdot y} \cdot \Pi_0$, translates for both labelings into the fact that U_0 and U_1 either commute ($y = 0$) or anticommute ($y = 1$):

$$U_0 U_1 = (-1)^y \cdot U_1 U_0 \,. \tag{4.2}$$

The task is to find out which property is the correct one.

4.2.1. FPPs are an entire class of problems

For $n \geq 3$, there are different ways to label the permutations that lead in general to inequivalent tasks (examples are given in Subsection 4.4.1). In this sense, Fourier promise problems form an entire class of problems and we use the term "specific Fourier promise problem" whenever we refer to a precise labeling of the permutations. To show that this class of problems is non-trivial, one has to prove that for every n there is at least one specific FPP for which there indeed exist unitaries that satisfy the promise. This is shown in Appendix A of the original work of M. Araújo et al. [27], where they construct for every $n \geq 2$ and every $y \in \{0, 1, ..., n! - 1\}$ a set of unitaries $\{U_i\}_0^{n-1}$ that satisfy the promise $\Pi_x = \omega^{x \cdot y} \cdot \Pi_0$ for a given labeling of the permutations.[1] We want to point out that, for a given n, this is the *only* specific FPP for which the existence of these unitaries is explicitly shown (and hence the only task that is proven to be non-trivial). For this specific task, we will present in Section 4.5 a causal algorithm that is very efficient in the amount of called black-box unitaries (queries), but has the disadvantage that it cannot be adapted directly to (other possibly existing non-trivial) FPPs where the permutations are labeled differently. In this sense, we say that this algorithm is able to solve only a

[1] While the precise form of these unitaries is not relevant in this article (since we will only use the fact that they satisfy the promise), we want to mention that their Hilbert space dimension must be at least $d = n!$. Since these unitaries satisfy the promise, we can consider $\Pi_1 = \omega^y \cdot \Pi_0$ and take the determinant on both sides:

$$det(\Pi_1) = \omega^{y \cdot d} \cdot det(\Pi_0) \,. \tag{4.3}$$

Since Π_0 and Π_1 are products of the same unitaries in different order, we obtain $det(\Pi_1) = det(\Pi_0)$ and therefore $\omega^{y \cdot d} = 1$. A solution for every $y \in \{0, 1, ..., n! - 1\}$ can only exist if $d \geq n!$ (remember that $\omega = e^{\frac{2\pi i}{n!}}$).

specific FPP. On the other hand, we say that a (causal or non-causal) quantum algorithm is able to solve every FPP for a given n, if the algorithm is able to solve every specific FPP with n black-box unitaries. More precisely, if the algorithm can be adapted to every possible labeling of the permutations (usually by adjusting the numbering of the states in the control system to the labeling of the permutations). It turns out that this is true for every algorithm in this work except the ones given in Section 4.5. Note, however, that in Ref. [27] the distinction between specific FPPs is not made explicitly, since only algorithms are considered that can be adapted to every FPP.

4.2.2. Solution with the quantum-n-switch

The quantum-n-switch (denoted as S_n and called n-switch for short) is the quantum gate that applies, depending on the state of the control system $|x\rangle$, the permutation Π_x on the target system $|\Psi_t\rangle$:

$$\forall x \in \{0, 1, ..., n! - 1\} : \ S_n \left|x\right\rangle_c \otimes |\Psi_t\rangle = \left|x\right\rangle_c \otimes \Pi_x |\Psi_t\rangle \ . \tag{4.4}$$

Moreover, since the $n!$-dimensional Fourier transform is frequently used in this article, we formally introduce it here. In symbols,

$$\forall y \in \{0, 1, ..., n! - 1\} : \ F_{n!} |y\rangle = \frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \omega^{x \cdot y} |x\rangle \ . \tag{4.5}$$

With the use of the n-switch, one can solve every FPP, as described in Ref. [27]; the control system is initialized in the $n!$-dimensional state $\left|0\right\rangle_c$ and the target system $|\Psi_t\rangle$ in an arbitrary d-dimensional state. The Fourier transform $F_{n!}$ transforms the control system into an equal superposition of all states $x \in \{0, 1, ..., n! - 1\}$:

$$(F_{n!} \left|0\right\rangle_c) \otimes |\Psi_t\rangle = \left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \left|x\right\rangle_c\right) \otimes |\Psi_t\rangle \ . \tag{4.6}$$

Afterwards, the n-switch applies, depending on the state $|x\rangle$ of the control system, the permutation Π_x on the target system $|\Psi_t\rangle$ (see Fig. 4.1 for an illustration of the map for the case of $n = 3$):

$$S_n \left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \left|x\right\rangle_c\right) \otimes |\Psi_t\rangle = \frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \left|x\right\rangle_c \otimes \Pi_x |\Psi_t\rangle \ . \tag{4.7}$$

With the promise $\Pi_x = \omega^{x \cdot y} \cdot \Pi_0$, this state can be rewritten into:

$$\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \left|x\right\rangle_c \otimes \Pi_x |\Psi_t\rangle = \left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \omega^{x \cdot y} \left|x\right\rangle_c\right) \otimes \Pi_0 |\Psi_t\rangle \ . \tag{4.8}$$

In this way, the target system becomes independent of x and factorizes out in the state $\Pi_0 |\Psi_t\rangle$. After applying the inverse Fourier transform on the control system, the desired value of y can be read out with a measurement of the control system in the computational basis:

$$F_{n!}^{-1}\left(\frac{1}{\sqrt{n!}}\sum_{x=0}^{n!-1}\omega^{x\cdot y}\left|x\right\rangle_c\right)\otimes\Pi_0\left|\Psi_t\right\rangle=\left|y\right\rangle_c\otimes\Pi_0\left|\Psi_t\right\rangle\ . \tag{4.9}$$

Since the n-switch can apply every permutation of the unitaries with a single call to each gate, the total query complexity of this algorithm is n.

4.2.3. Solution with causal quantum algorithms

In this section, we give an overview of the best causal algorithms for FPPs that are known. All of them are based on the simulation of the n-switch and call $O(n^2)$ black-box unitaries. A causal quantum algorithm simulates the action of the n-switch (denoted as $S_n^{\text{sim.}}$) if it implements the transformation

$$S_n^{\text{sim.}}\left|x\right\rangle_c\otimes\left|\Psi_t\right\rangle\otimes\left(\bigotimes_{i=0}^{n-1}\left|a_i\right\rangle\right)=\left|x\right\rangle_c\otimes\Pi_x\left|\Psi_t\right\rangle\otimes\left(\bigotimes_{i=0}^{n-1}(U_i)^{k_i}\left|a_i\right\rangle\right) \tag{4.10}$$

for every $x\in\{0,1,...,n!-1\}$, arbitrary d-dimensional states $\left|\Psi_t\right\rangle$ and $\left|a_i\right\rangle$ as well as constants k_i that do not depend on x. Every simulation of the n-switch can be used in combination with the algorithm in Fig. 4.2 to solve every FPP; analogously to the n-switch in the last subsection, the control system is prepared with a quantum Fourier transform in an equal superposition over all states $x\in\{0,1,...,n!-1\}$. By linearity, an algorithm that simulates the n-switch implements the transformation:

$$\begin{aligned}
&S_n^{\text{sim.}}\left(\frac{1}{\sqrt{n!}}\sum_{x=0}^{n!-1}\left|x\right\rangle_c\right)\otimes\left|\Psi_t\right\rangle\otimes\left(\bigotimes_{i=0}^{n-1}\left|a_i\right\rangle\right)\\
&=\left(\frac{1}{\sqrt{n!}}\sum_{x=0}^{n!-1}\left|x\right\rangle_c\otimes\Pi_x\left|\Psi_t\right\rangle\right)\otimes\left(\bigotimes_{i=0}^{n-1}(U_i)^{k_i}\left|a_i\right\rangle\right)\\
&=\left(\frac{1}{\sqrt{n!}}\sum_{x=0}^{n!-1}\omega^{x\cdot y}\left|x\right\rangle_c\right)\otimes\Pi_0\left|\Psi_t\right\rangle\otimes\left(\bigotimes_{i=0}^{n-1}(U_i)^{k_i}\left|a_i\right\rangle\right)\ .
\end{aligned} \tag{4.11}$$

Again, the promise $\Pi_x=\omega^{x\cdot y}\cdot\Pi_0$ is used to obtain the last equality. After applying the inverse Fourier transform to the control system, the solution y can be read out in the control system.

One algorithm that can implement the transformation $S_n^{\text{sim.}}$ is given in Fig. 4.3. This one was originally introduced in Ref. [173] and is also presented in Appendix C of Ref. [27]. In each step $i=0,1,...,n-1$, the gate $\mathcal{S}$ swaps, controlled on $\left|x\right\rangle$, the target system $\left|\Psi_t\right\rangle$ with the auxiliary system $\left|a_{\sigma_x(i)}\right\rangle$. After the gate $U_{\sigma_x(i)}$ acts on $\left|\Psi_t\right\rangle$, another gate $\mathcal{S}$ swaps the two systems $\left|\Psi_t\right\rangle$ and $\left|a_{\sigma_x(i)}\right\rangle$ back. In this way, the permutation $\Pi_x=U_{\sigma_x(n-1)}...U_{\sigma_x(1)}U_{\sigma_x(0)}$ is applied to the target system. In this algorithm, each gate

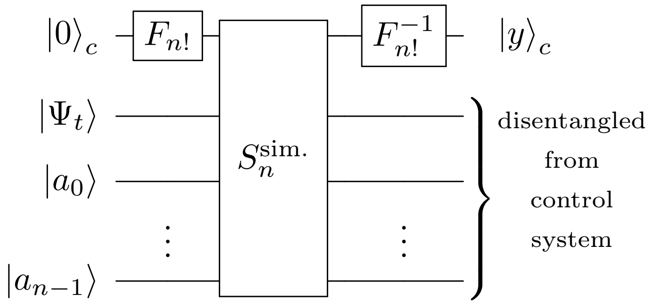


Figure 4.2.: Solution of every FPP with the simulation of the quantum-n-switch: With a Fourier transform the control system is prepared in an equal superposition of all states x. The simulation of the n-switch S_n^{sim} applies, depending on the state of the control system $|x\rangle$, the permutation Π_x on the target system. The solution y can be read out after applying the inverse Fourier transform to the control system.

U_i is used exactly n times, so the query complexity of this algorithm is n^2. Furthermore, for every permutation Π_x, each auxiliary system $|a_i\rangle$ is swapped back and forth with $|\Psi_t\rangle$ exactly once. Hence, independently of the state of the control system $|x\rangle$, the gate U_i is applied once on $|\Psi_t\rangle$ and the remaining $(n-1)$ times on $|a_i\rangle$. In this way, each auxiliary system $|a_i\rangle$ ends up in the state $(U_i)^{n-1}|a_i\rangle$ and the algorithm implements the transformation given in Eq. (4.10) for $k_i = (n-1)$. [2]

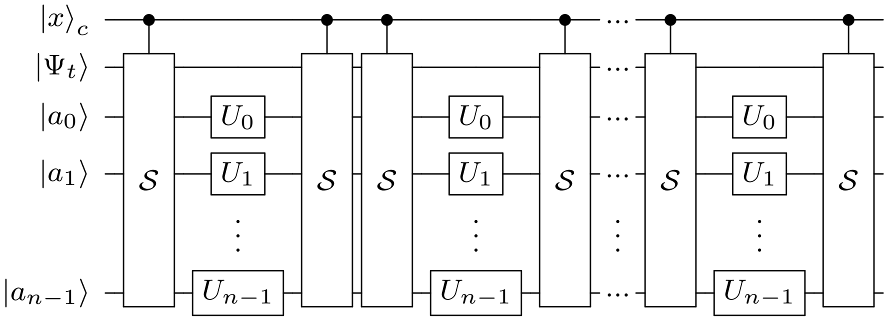


Figure 4.3.: A simulation of the n-switch with a causal algorithm: The permutation Π_x is applied on $|\Psi_t\rangle$ by swapping the target system in each step $i = 0, 1, ..., n-1$ with the auxiliary system $|a_{\sigma_x(i)}\rangle$.

There are simulations of the quantum-n-switch with causal quantum circuits that are

[2] Further details about the representation of the control system $|x\rangle_c$ and the implementation of the $\mathcal{S}$-gate can be found in Appendix C of Ref. [27] and in Ref. [173].

slightly more efficient. All of them call $O(n^2)$ black-box gates in total. This is studied in Ref. [174] (and also in Ref. [27]).

4.3. Towards more efficient causal algorithms

4.3.1. Pairwise commutation relations

In this article, we show that there are algorithms that solve Fourier promise problems by calling significantly less queries than the simulation of the n-switch requires. The main ingredient is a property of the unitaries that can be directly inferred from the promise.

Definition 4.2 (Pairwise commutation relations). *A set of unitaries $\{U_i\}_0^{n-1}$ satisfy* "pairwise commutation relations"*, if for every pair of unitaries U_j and U_k ($j, k \in \{0, 1, ..., n-1\}$) there exist $\alpha_{jk} \in \mathbb{C}$ such that:*

$$U_j U_k = \alpha_{jk} \cdot U_k U_j \, . \tag{4.12}$$

Proposition 4.1. *Every set of unitaries $\{U_i\}_0^{n-1}$ that satisfy the promise of a Fourier promise problem,*

$$\forall x \in \{0, 1, ..., n! - 1\} : \Pi_x = \omega^{x \cdot y} \cdot \Pi_0 \, , \tag{4.13}$$

satisfy pairwise commutation relations. Furthermore, if for a specific FPP the labeling of the permutations is given, the pairwise commutation relations read:

$$U_j U_k = \omega^{(x_{jk}^1 - x_{jk}^2) \cdot y} \cdot U_k U_j \, , \tag{4.14}$$

where x_{jk}^1 is the label of the permutation $\Pi_{x_{jk}^1} = U_{n-1} ... U_1 U_0 U_j U_k$ and x_{jk}^2 is the label of the permutation $\Pi_{x_{jk}^2} = U_{n-1} ... U_1 U_0 U_k U_j$ (with $U_{n-1} ... U_1 U_0$, we denote all unitaries without U_j and U_k in descending order).[3]

Proof. For every pair of black-box unitaries U_j and U_k, we focus on the two permutations introduced in the statement above:

$$\Pi_{x_{jk}^1} = U_{n-1} ... U_1 U_0 U_j U_k \text{ and} \tag{4.15}$$

$$\Pi_{x_{jk}^2} = U_{n-1} ... U_1 U_0 U_k U_j \, . \tag{4.16}$$

[3] Note that with some caution, one could turn this statement into an equivalence: the condition of pairwise commutation relations is not only a necessary condition for the promise to hold. One can also check by direct calculation that whenever all permutation relations are pairwise ($\forall j \neq k \; \exists \; x_{jk} \in \mathbb{Z}$ s.t. $U_j U_k = \omega^{x_{jk} \cdot y} \cdot U_k U_j$), all permutations of the n unitaries are related by a phase ($\Pi_x = \omega^{x \cdot y} \, \Pi_0$). In order for the promise to hold, one has to choose the pairwise phases x_{jk} such that every $x \in \{0, 1, ..., n! - 1\}$ appears exactly once. Nevertheless, it is enough for our purpose that the promise induces pairwise commutation relations.

Due to the promise $\Pi_x = \omega^{x\cdot y}\,\Pi_0$, both permutations are equal to Π_0 up to the phases $\omega^{-x^1_{jk}\cdot y}$ and $\omega^{-x^2_{jk}\cdot y}$, respectively:

$$\omega^{-x^1_{jk}\cdot y}\cdot\Pi_{x^1_{jk}} = \Pi_0 = \omega^{-x^2_{jk}\cdot y}\cdot\Pi_{x^2_{jk}}\,. \tag{4.17}$$

Using the above expression for the two permutations $\Pi_{x^1_{jk}}$ and $\Pi_{x^2_{jk}}$, we obtain

$$\omega^{-x^1_{jk}\cdot y}\cdot U_{n-1}...U_0U_jU_k = \omega^{-x^2_{jk}\cdot y}\cdot U_{n-1}...U_0U_kU_j\,. \tag{4.18}$$

Multiplying from the left step by step with the inverses of U_{n-1}, U_{n-2}, ..., U_1 and U_0 (for U_j and U_k this is left out), this expression is equivalent to:

$$U_jU_k = \omega^{(x^1_{jk}-x^2_{jk})\cdot y}\cdot U_kU_j\,. \tag{4.19}$$

With $\alpha_{jk} := \omega^{(x^1_{jk}-x^2_{jk})\cdot y}\in\mathbb{C}$, we conclude that every set of unitaries that satisfies the promise also satisfies pairwise commutation relations. □

4.3.2. Structure of the new algorithms

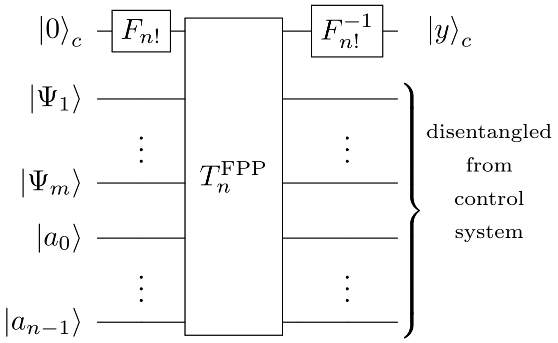


Figure 4.4.: General structure of the algorithms in this article: With a quantum Fourier transform the control system is initialized in an equal superposition of all states $|x\rangle$. After the transformation T^{FPP}_n (see main text) is applied, the final state of each target and auxiliary system becomes independent of x and the solution y can be read out by a measurement of the control system in the Fourier basis.

All causal algorithms that we present in this article are of the form given in Fig. 4.4. The target systems $|\Psi_j\rangle$ and auxiliary systems $|a_i\rangle$ are initialized in an arbitrary d-dimensional state. The important part of the algorithm is the one that realizes the transformation

T_n^{FPP}:

$$\begin{aligned}T_n^{\mathrm{FPP}} \left|x\right\rangle_c \otimes \bigotimes_{j=1}^{m} \left|\Psi_j\right\rangle \otimes \bigotimes_{i=0}^{n-1} \left|a_i\right\rangle &= \left|x\right\rangle_c \otimes \bigotimes_{j=1}^{m} f_j^x(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle \otimes \bigotimes_{i=0}^{n-1} (U_i)^{k_i} \left|a_i\right\rangle \\ &= \omega^{x\cdot y} \left|x\right\rangle_c \otimes \bigotimes_{j=1}^{m} f_j^0(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle \otimes \bigotimes_{i=0}^{n-1} (U_i)^{k_i} \left|a_i\right\rangle \ .\end{aligned} \tag{4.20}$$

On each target system $\left|\Psi_j\right\rangle$, some of the n black-box unitaries from the set are applied, such that each of them ends up in a state $f_j^x(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle$. The unitaries contained within $f_j^x(U_0,...,U_{n-1})$ are the same for every x, but the order in which these unitaries are applied on the target system will explicitly depend on x. Since all commutation relations are pairwise, one can always rewrite this expression into $f_j^0(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle$ and a phase is obtained whenever two unitaries are commuted (with $e^{i\phi_j(x)}$ we denote the product of these phases):

$$f_j^x(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle = e^{i\phi_j(x)}\ f_j^0(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle \ . \tag{4.21}$$

In this way, the final state of each target system becomes independent of x and if the algorithm is designed carefully, all these phases multiply together to $\omega^{x\cdot y}$.

Whenever we find an implementation of this transformation T_n^{FPP} for every $x \in \{0, 1, ..., n! - 1\}$ we can solve the corresponding FPP. The control system is initialized in an equal superposition of all x and due to linearity, the transformation T_n^{FPP} realizes:

$$\begin{aligned}&T_n^{\mathrm{FPP}} \left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \left|x\right\rangle_c \right) \otimes \left(\bigotimes_{j=1}^{m} \left|\Psi_j\right\rangle \right) \otimes \left(\bigotimes_{i=0}^{n-1} \left|a_i\right\rangle \right) \\ &= \left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \omega^{x\cdot y} \left|x\right\rangle_c \right) \otimes \left(\bigotimes_{j=1}^{m} f_j^0(U_0,...,U_{n-1}) \left|\Psi_j\right\rangle \right) \otimes \left(\bigotimes_{i=0}^{n-1} (U_i)^{k_i} \left|a_i\right\rangle \right) \ .\end{aligned} \tag{4.22}$$

At the end, the solution y can be read out after applying the inverse Fourier transform to the control system (see Fig. 4.4).

Intuitively speaking, the pairwise commutation relations allow us to simulate different parts of the total phase $\omega^{x\cdot y}$ on different target systems. Note that the best causal algorithms so far that are based on the simulation of the n-switch (Subsection 4.2.3) can be seen as a special case of this new method with only one target system ($m = 1$) and $f_1^x(U_0,...,U_{n-1}) = \Pi_x$. Hence, this new procedure is more general and usually more efficient than simulating every permutation on its own. We want to point out that these ideas can in principle be applied to every set of unitaries satisfying pairwise commutation relations. In this sense, this method may have some applications beyond Fourier promise problems.

4.4. More efficient solutions for FPPs with three unitaries

In this section, we show in a first example how pairwise commutation relations are useful to find more efficient causal algorithms. More precisely, we present an algorithm that solves every FPP for $n = 3$ with six queries, while the best causal algorithm that was known so far used seven queries. While this difference might not seem significant at first, we will show in the next sections that similar ideas can be used for a significant reduction in the number of used black-box gates in the asymptotic limit.

4.4.1. Two possible ways to label the permutations

Before we present the algorithms, we give two specific examples of labelings for $n = 3$. We denote with $x_{ijk} \in \{0, 1, ..., 3! - 1\}$ the label of the permutation $U_i U_j U_k$:

$$0 = x_{210} \qquad \Pi_0 = U_2 U_1 U_0 \tag{4.23}$$

$$1 = x_{201} \qquad \Pi_1 = U_2 U_0 U_1 = \omega^{1 \cdot y}\ U_2 U_1 U_0 \tag{4.24}$$

$$2 = x_{120} \qquad \Pi_2 = U_1 U_2 U_0 = \omega^{2 \cdot y}\ U_2 U_1 U_0 \tag{4.25}$$

$$3 = x_{021} \qquad \Pi_3 = U_0 U_2 U_1 = \omega^{3 \cdot y}\ U_2 U_1 U_0 \tag{4.26}$$

$$4 = x_{102} \qquad \Pi_4 = U_1 U_0 U_2 = \omega^{4 \cdot y}\ U_2 U_1 U_0 \tag{4.27}$$

$$5 = x_{012} \qquad \Pi_5 = U_0 U_1 U_2 = \omega^{5 \cdot y}\ U_2 U_1 U_0\,. \tag{4.28}$$

If it is promised that the three unitaries satisfy these conditions, it is straightforward to read off the pairwise commutation relations: from the second line $U_2 U_0 U_1 = \omega^{1 \cdot y}\ U_2 U_1 U_0$, it follows that $U_0 U_1 = \omega^{1 \cdot y}\ U_1 U_0$, and from the third line, one can conclude that $U_1 U_2 = \omega^{2 \cdot y}\ U_2 U_1$. For the last pair U_0 and U_2, we have to put in some more work: Due to Proposition 4.1, we can compare the fifth ($U_1 U_0 U_2$) and the third line ($U_1 U_2 U_0$) to obtain:

$$\omega^{-4 \cdot y}\ U_1 U_0 U_2 = U_2 U_1 U_0 = \omega^{-2 \cdot y}\ U_1 U_2 U_0 \qquad \Longrightarrow\ U_0 U_2 = \omega^{2 \cdot y}\ U_2 U_0\,. \tag{4.29}$$

There exist other specific FPPs that correspond to other labelings. One example is the following:

$$0 = x_{210} \qquad \Pi_0 = U_2 U_1 U_0 \tag{4.30}$$

$$1 = x_{102} \qquad \Pi_1 = U_1 U_0 U_2 = \omega^{1 \cdot y}\ U_2 U_1 U_0 \tag{4.31}$$

$$2 = x_{201} \qquad \Pi_2 = U_2 U_0 U_1 = \omega^{2 \cdot y}\ U_2 U_1 U_0 \tag{4.32}$$

$$3 = x_{012} \qquad \Pi_3 = U_0 U_1 U_2 = \omega^{3 \cdot y}\ U_2 U_1 U_0 \tag{4.33}$$

$$4 = x_{120} \qquad \Pi_4 = U_1 U_2 U_0 = \omega^{4 \cdot y}\ U_2 U_1 U_0 \tag{4.34}$$

$$5 = x_{021} \qquad \Pi_5 = U_0 U_2 U_1 = \omega^{5 \cdot y}\ U_2 U_1 U_0\,. \tag{4.35}$$

Here, the pairwise commutation relations can be read off as follows: $U_0 U_1 = \omega^{2 \cdot y}\ U_1 U_0$, $U_0 U_2 = \omega^{3 \cdot y}\ U_2 U_0$ and $U_1 U_2 = \omega^{4 \cdot y}\ U_2 U_1$. On the other hand, knowing all pairwise phases uniquely determines the labeling (up to the freedom of choosing Π_0).

Note that not every labeling is meaningful. Some of them lead to trivial statements:

$$0 = x_{210} \qquad \Pi_0 = U_2U_1U_0 \tag{4.36}$$

$$1 = x_{201} \qquad \Pi_1 = U_2U_0U_1 = \omega^{1\cdot y}\ U_2U_1U_0 \tag{4.37}$$

$$2 = x_{102} \qquad \Pi_2 = U_1U_0U_2 = \omega^{2\cdot y}\ U_2U_1U_0 \tag{4.38}$$

$$3 = x_{120} \qquad \Pi_3 = U_1U_2U_0 = \omega^{3\cdot y}\ U_2U_1U_0 \tag{4.39}$$

$$4 = x_{012} \qquad \Pi_4 = U_0U_1U_2 = \omega^{4\cdot y}\ U_2U_1U_0 \tag{4.40}$$

$$5 = x_{021} \qquad \Pi_5 = U_0U_2U_1 = \omega^{5\cdot y}\ U_2U_1U_0\,. \tag{4.41}$$

From the second line, it follows $U_0U_1 = \omega^{1\cdot y}\ U_1U_0$, while from comparing $\Pi_2 = U_1U_0U_2$ with $\Pi_4 = U_0U_1U_2$, one obtains $U_0U_1 = \omega^{2\cdot y}\ U_1U_0$. This is a contradiction whenever $y \neq 0$ (note that $\omega = e^{\frac{2\pi i}{n!}} \neq 1$ for $n \geq 2$). More precisely, only for $y = 0$, there exist unitaries U_0, U_1 and U_2 that satisfy this promise, and the task becomes trivial (since one can conclude directly that the solution must be $y = 0$). By counting, we found that there are 24 different possible labelings of the six permutations that lead to non-trivial solutions for $n = 3$ if we restrict ourselves to $x_{210} = 0$ ($\Pi_0 = U_2U_1U_0$).

4.4.2. Standard causal algorithm with seven queries

The best causal algorithms known so far that solve these problems are based on the simulation of the 3-switch and call seven black-box unitaries. One possible algorithm that can achieve this is given in Fig. 4.5. The gates $\mathcal{R}$ denote rewirings of the target and auxiliary systems (a combination of controlled swaps). Depending on the state of the control system $|x_{ijk}\rangle$, they interchange the wires in a way that the gates act on the systems according to Table 4.1. All underlined gates $\underline{U_i}$ act on $|\Psi_t\rangle$ and simulate the permutation $U_iU_jU_k$, while the remaining (unused) gates U_0 and U_1 act on the corresponding auxiliary systems $|a_0\rangle$ and $|a_1\rangle$, respectively.

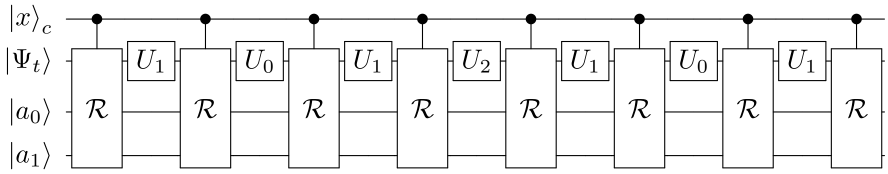


Figure 4.5.: Simulation of the 3-switch ($S_3^{\text{sim.}}$) with the smallest possible number of used black-box gates.

In total, the combined control and target system simulate the action of the 3-switch, since every permutation of the three unitaries can be applied on the target system, while the two auxiliary systems always end up in the same state

$$S_3^{\text{sim.}}\ |x_{ijk}\rangle \otimes |\Psi_t\rangle \otimes |a_0\rangle \otimes |a_1\rangle = |x_{ijk}\rangle \otimes U_iU_jU_k\,|\Psi_t\rangle \otimes U_0\,|a_0\rangle \otimes (U_1)^3\,|a_1\rangle\ . \tag{4.42}$$

	$S_3^{\text{sim.}} \left(\lvert x_{ijk} \rangle \otimes \lvert \Psi_t \rangle \otimes \lvert a_0 \rangle \otimes \lvert a_1 \rangle \right)$
$U_1\underline{U_0U_1U_2}U_1U_0U_1$	$\lvert x_{210} \rangle \otimes \underline{U_2U_1U_0} \lvert \Psi_t \rangle \otimes U_0 \lvert a_0 \rangle \otimes (U_1)^3 \lvert a_1 \rangle$
$\underline{U_1U_0}U_1\underline{U_2}U_1U_0U_1$	$\lvert x_{201} \rangle \otimes \underline{U_2U_0U_1} \lvert \Psi_t \rangle \otimes U_0 \lvert a_0 \rangle \otimes (U_1)^3 \lvert a_1 \rangle$
$U_1\underline{U_0}U_1\underline{U_2U_1}U_0U_1$	$\lvert x_{120} \rangle \otimes \underline{U_1U_2U_0} \lvert \Psi_t \rangle \otimes U_0 \lvert a_0 \rangle \otimes (U_1)^3 \lvert a_1 \rangle$
$U_1U_0\underline{U_1U_2}U_1\underline{U_0}U_1$	$\lvert x_{021} \rangle \otimes \underline{U_0U_2U_1} \lvert \Psi_t \rangle \otimes U_0 \lvert a_0 \rangle \otimes (U_1)^3 \lvert a_1 \rangle$
$U_1U_0U_1\underline{U_2}U_1\underline{U_0U_1}$	$\lvert x_{102} \rangle \otimes \underline{U_1U_0U_2} \lvert \Psi_t \rangle \otimes U_0 \lvert a_0 \rangle \otimes (U_1)^3 \lvert a_1 \rangle$
$U_1U_0U_1\underline{U_2U_1U_0}U_1$	$\lvert x_{012} \rangle \otimes \underline{U_0U_1U_2} \lvert \Psi_t \rangle \otimes U_0 \lvert a_0 \rangle \otimes (U_1)^3 \lvert a_1 \rangle$

Table 4.1.: Final state for every $\lvert x_{ijk} \rangle$ of the algorithm in Fig. 4.5.

As explained in Subsection 4.2.3, this solves every FPP as the 3-switch itself.

It is essential for an algorithm that simulates the n-switch that every permutation of the unitaries can be applied on the target system by rewiring the systems in some way. Here, for $n = 3$, every permutation has to appear as a substring in $U_1U_0U_1U_2U_1U_0U_1$. The minimal length of a string of elements $U_0, U_1, ..., U_{n-1}$ such that all possible permutations of the n elements are contained in the string as a substring is a well-studied problem in combinatorics: it is known that the minimal number of elements in such a string is of the order of $O(n^2)$ [175]. For $3 \leq n \leq 7$, the shortest string containing all permutations as a substring has length $n^2 - 2n + 4$ [176]. For higher n, more efficient constructions are known [177, 178]. For this reason, no string of length smaller than seven can contain all the six permutations of three unitaries.

4.4.3. More efficient causal algorithm with six queries

Here, we will show that we can solve every FPP for $n = 3$ with only six queries using the algorithm given in Fig. 4.6. In accordance with the methods introduced in Subsection 4.3.2, we introduce a second target system. As before, each of the gates U_0, U_1 and U_2 are applied on the first target system exactly once. Since we only use six queries, not every permutation of the three unitaries can be realized on the first target system. Nevertheless, we can use the second target system, on which each of the gates U_0 and U_1 are applied exactly once, to simulate the pairwise phase between U_0 and U_1 if necessary. Via the "phase-kickback," the phases of both registers are accumulated in the control system at the end. In this way, the algorithm is not able to simulate every permutation on its own but is able to simulate the product of the pairwise phases for each permutation, which is enough to solve Fourier promise problems. This procedure requires less queries and is the central idea of all the algorithms in our work.

As in the last subsection, the gates $\mathcal{R}$ are rewirings of the target and auxiliary systems. They interchange the wires in a way that the gates act on the systems according to Tab. 4.2. All underlined gates $\underline{U_i}$ act on $\lvert \Psi_1 \rangle$, all overlined gates $\overline{U_i}$ on $\lvert \Psi_2 \rangle$ and the remaining gate U_1 acts on the auxiliary system $\lvert a_1 \rangle$.

The crucial difference here is the permutation $U_2U_0U_1$. It is not possible that the first target system $\lvert \Psi_1 \rangle$ ends up in the state $U_2U_0U_1 \lvert \Psi_1 \rangle$ directly, since the permutation $U_1U_0U_2$ is not contained as a substring of $U_0U_1U_2U_1U_0U_1$ (remember that the order

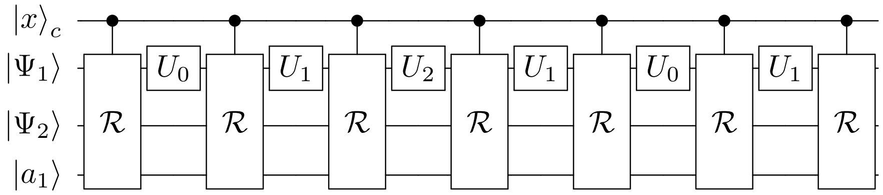


Figure 4.6.: An implementation of T_3^{FPP} that solves every FPP for three unitaries.

	$T_3^{\text{FPP}}\left(\lvert x_{ijk}\rangle \otimes \lvert\Psi_1\rangle \otimes \lvert\Psi_2\rangle \otimes \lvert a_1\rangle\right)$
$\underline{U_0U_1U_2}U_1\overline{U_0U_1}$	$\lvert x_{210}\rangle \otimes \underline{U_2U_1U_0}\lvert\Psi_1\rangle \otimes \overline{U_1U_0}\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$
$\underline{U_0U_1U_2}\overline{U_1U_0}U_1$	$\lvert x_{201}\rangle \otimes \underline{U_2U_1U_0}\lvert\Psi_1\rangle \otimes \overline{U_0U_1}\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$
	$= \lvert x_{201}\rangle \otimes U_2U_0U_1\lvert\Psi_1\rangle \otimes U_1U_0\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$
$\underline{U_0}U_1\underline{U_2U_1}\overline{U_0U_1}$	$\lvert x_{120}\rangle \otimes \underline{U_1U_2U_0}\lvert\Psi_1\rangle \otimes \overline{U_1U_0}\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$
$\overline{U_0}\underline{U_1U_2}U_1\underline{U_0}\overline{U_1}$	$\lvert x_{021}\rangle \otimes \underline{U_0U_2U_1}\lvert\Psi_1\rangle \otimes \overline{U_1U_0}\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$
$\overline{U_0U_1}\underline{U_2}U_1\underline{U_0U_1}$	$\lvert x_{102}\rangle \otimes \underline{U_1U_0U_2}\lvert\Psi_1\rangle \otimes \overline{U_1U_0}\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$
$\overline{U_0U_1}\underline{U_2U_1U_0}U_1$	$\lvert x_{012}\rangle \otimes \underline{U_0U_1U_2}\lvert\Psi_1\rangle \otimes \overline{U_1U_0}\lvert\Psi_2\rangle \otimes U_1\lvert a_1\rangle$

Table 4.2.: Final state for every $|x_{ijk}\rangle$ of the algorithm in Fig. 4.6. Using pairwise commutation relations, one can rewrite the second line (see main text).

is reversed since U_1 has to act first, then U_0 and U_2). However, since every set of unitaries that satisfies the promise satisfies pairwise commutation relations, we can use $U_1U_0 = \alpha_{10}\ U_0U_1$ to rewrite the second line of Table 4.2 into:

$$\begin{aligned}
&|x_{201}\rangle \otimes U_2U_1U_0\,|\Psi_1\rangle \otimes U_0U_1\,|\Psi_2\rangle \otimes U_1\,|a_1\rangle \\
&= \alpha_{10}\ \ |x_{201}\rangle \otimes U_2U_0U_1\,|\Psi_1\rangle \otimes U_0U_1\,|\Psi_2\rangle \otimes U_1\,|a_1\rangle \\
&= |x_{201}\rangle \otimes U_2U_0U_1\,|\Psi_1\rangle \otimes U_1U_0\,|\Psi_2\rangle \otimes U_1\,|a_1\rangle\ .
\end{aligned} \tag{4.43}$$

Due to this, the algorithm implements a transformation that is very similar to the one that is implemented by the algorithm in the last subsection (Eq. (4.42)). For every x_{ijk} the permutation $U_iU_jU_k$ is applied on the first target system, while the second target system and the auxiliary system always end up in the state $U_1U_0\,|\Psi_2\rangle \otimes U_1\,|a_1\rangle$:

$$\begin{aligned}
&T_3^{\text{FPP}}\ \ |x_{ijk}\rangle \otimes |\Psi_1\rangle \otimes |\Psi_2\rangle \otimes |a_1\rangle \\
&\quad = |x_{ijk}\rangle \otimes U_iU_jU_k\,|\Psi_1\rangle \otimes U_1U_0\,|\Psi_2\rangle \otimes U_1\,|a_1\rangle\ .
\end{aligned} \tag{4.44}$$

In this way, the combined control and first target system simulate the action of the 3-switch for unitaries satisfying pairwise commutation relations. Therefore, this algorithm solves every FPP for $n = 3$ in the same way as the algorithm in the last subsection.[4]

[4] Alternatively, to stay within the methods developed in Subsection 4.3.2, note that Eq. (4.44) can be rewritten into $\omega^{x_{ijk}\cdot y}\ |x_{ijk}\rangle \otimes \Pi_0\,|\Psi_1\rangle \otimes U_1U_0\,|\Psi_2\rangle \otimes U_1\,|a_1\rangle$ by using the promise $\Pi_{x_{ijk}} = U_iU_jU_k = \omega^{x_{ijk}\cdot y}\ \Pi_0$. In this way, the final state of each target and auxiliary system is independent of x_{ijk} and the algorithm in Fig. 4.4 can be applied directly.

In Section 4.6, we use similar ideas and present an algorithm that simulates the action of the n-switch for unitaries that satisfy pairwise commutation relations (and is therefore able to solve every FPP) with $O(n\sqrt{n})$ queries. In order to avoid confusion, we want to point out that the algorithm of this subsection is not the particular case of the algorithm in Section 4.6 for $n = 3$.

4.5. Causal algorithms with query complexity $O(n \log n)$

In this section, we present an algorithm that solves the FPP with the labeling used in Appendix A of Ref. [27] with $O(n \log n)$ queries. First, we recall how the permutations are labeled and derive the pairwise commutation relations for this labeling.

The identity permutation is defined as:

$$\Pi_0 := U_{n-1}U_{n-2}...U_1U_0 \,. \tag{4.45}$$

The labeling of all other permutations Π_x is based on the factorial number system; every x is represented with $n-1$ integers $(a_{n-1}, ..., a_1)$ where $a_k \in \{0, 1, ..., k\}$: [5]

$$x = \sum_{k=1}^{n-1} a_k \cdot k! \,. \tag{4.46}$$

Starting with the identity permutation $\Pi_0 = U_{n-1}...U_1U_0$, we obtain the permutation Π_x by shifting first U_1 $a_1 \in \{0, 1\}$ steps to the right, then U_2 $a_2 \in \{0, 1, 2\}$ steps to the right and so on. The labeling for $n = 3$ is given as the first example in Subsection 4.4.1 (Eq. (4.23)-(4.28)). We call this labeling of the permutations the “factoradic” labeling.

Due to Proposition 4.1, we can read off the commutation relations for every pair of unitaries U_j and U_k (w.l.o.g. we assume here $j < k$). The two permutations we have to focus on are:

$$\Pi_{x^1_{jk}} = U_{n-1}...U_1U_0U_jU_k \text{ and} \tag{4.47}$$

$$\Pi_{x^2_{jk}} = U_{n-1}...U_1U_0U_kU_j \,. \tag{4.48}$$

To construct the first permutation $\Pi_{x^1_{jk}}$ from the identity permutation Π_0, the unitary U_j is first shifted j steps to the right. In a second step, U_k is shifted k steps to the right, while all other unitaries are not shifted. Hence, the label of $\Pi_{x^1_{jk}}$ is $x^1_{jk} = k \cdot k! + j \cdot j!$. To obtain the permutation $\Pi_{x^2_{jk}}$ from the identity permutation, the unitary U_j is first shifted j steps to the right and afterwards, U_k is shifted $(k-1)$ steps to the right. The remaining unitaries are not shifted. Therefore, the label of the second permutation is $x^2_{jk} = (k-1) \cdot k! + j \cdot j!$. If we combine this with Eq. (4.14), we obtain:

$$U_jU_k = \omega^{(x^1_{jk} - x^2_{jk}) \cdot y} \cdot U_kU_j = \omega^{k! \cdot y} \cdot U_kU_j \,. \tag{4.49}$$

[5] This is Eq. (A4) in Ref. [27].

Hence, whenever a unitary U_k of the set for which the promise holds is commuted with a unitary of a smaller index, the result remains unchanged up to a phase $\omega^{k!\cdot y}$. [6]

To introduce the idea, we give the algorithm for $n = 4$ in the next subsection and generalize the procedure thereafter. Note that the query complexity of this example is actually worse than with the conventional method; it requires 18 queries, while the most efficient simulation of the quantum-4-switch calls twelve black-box unitaries.[7] Nonetheless, it is an instructive example whose generalization results in a significant reduction of the query complexity. As a further remark, we want to point out that we use the notation of controlled unknown unitaries merely for convenience. Note, however, that controlling unknown unitaries is impossible within the standard quantum circuit model [179] but can be realized in the interferometric type of setups [180, 181, 182]. At the end of this section (Subsection 4.5.2), we show that it is possible to rewrite the algorithm in a form that does not control unknown unitaries.

4.5.1. The algorithm for n=4

For our purpose, it is useful to represent every number $x \in \{0, 1, ..., 4! - 1\}$ in a basis of bits $c^x_{k,i} \in \{0, 1\}$. More precisely, we identify the state $|x\rangle_c$ with a six-qubit state

$$|x\rangle_c = \bigotimes_{\substack{1\leq k\leq 3\\ 1\leq i\leq 2}} \left|c^x_{k,i}\right\rangle . \tag{4.50}$$

The bits $c^x_{k,i}$ satisfy the following equation:

$$x = c^x_{3,1} \cdot 12 + c^x_{3,2} \cdot 6 + c^x_{2,1} \cdot 2 + c^x_{2,2} \cdot 2 + c^x_{1,1} \cdot 1 + c^x_{1,2} \cdot 1 . \tag{4.51}$$

For example, $x = 16$ can be written as $16 = 12 + 2 + 2$. Hence, $|x = 16\rangle_c$ is represented by $\left|c^{16}_{3,1} = 1\right\rangle \otimes \left|c^{16}_{3,2} = 0\right\rangle \otimes \left|c^{16}_{2,1} = 1\right\rangle \otimes \left|c^{16}_{2,2} = 1\right\rangle \otimes \left|c^{16}_{1,1} = 0\right\rangle \otimes \left|c^{16}_{1,2} = 0\right\rangle$. It is simple to check that indeed every $x \in \{0, 1, ..., 4! - 1\}$ can be represented in this way. Note that this representation is not unique and most numbers can be decomposed in more than one way. For our purpose, it is enough to choose one representation for every $x \in \{0, 1, ..., 4! - 1\}$.

In accordance with the methods introduced in Subsection 4.3.2, we will show that the algorithm in Fig. 4.7 can implement the transformation T^{FPP}_4 for this FPP. To see this, we look first at the target system $|\Psi_{2,1}\rangle$. The gates U_3, U_1 and U_0 act on this system but the order in which they are applied depends on the control qubits $\left|c^x_{3,1}\right\rangle$ and $\left|c^x_{1,1}\right\rangle$. If both control qubits are in the state $|0\rangle$, this target system ends up in the state $U_3U_1U_0\,|\Psi_{2,1}\rangle$.

[6] This is equivalent to Eq. (A3) of Ref. [27]. The difference is that here, we derive the pairwise commutation relations from the promise and in Ref. [27], the pairwise commutation relations are used to show that unitaries which satisfy the promise exist (an explicit example of unitaries that satisfy the promise can also be found in Appendix A of Ref. [27]).

[7] Following the example given in Subsection 4.4.2, the shortest string containing all permutations of the four unitaries has twelve elements, for example $U_1U_2U_3U_4U_1U_2U_3U_1U_4U_2U_1U_3$ [176].

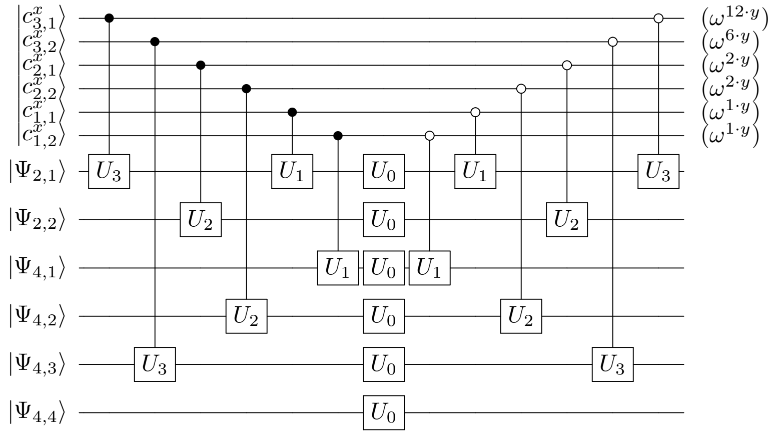


Figure 4.7.: Implementation of the transformation T_4^{FPP} for the factoradic labeling of the permutations. The labeling of the target systems $|\Psi_{2^i,j}\rangle$ is introduced in Subsec. 4.5.2. For the moment, it is only important that each target system is initialized in an arbitrary d-dimensional state.

On the other hand, if one or both of the two control qubits are in the state $|1\rangle$, the order of these gates is different. Nevertheless, due to the pairwise commutation relations given in Eq. (4.49), we can always rewrite the final state of the system $|\Psi_{2,1}\rangle$ into $U_3U_1U_0\,|\Psi_{2,1}\rangle$. By doing so, a phase is obtained whenever two unitaries are commuted (see Table 4.3).

$\lvert c^x_{3,1}\rangle$	$\lvert c^x_{1,1}\rangle$	Final state of $\lvert\Psi_{2,1}\rangle$				
$\lvert 0\rangle$	$\lvert 0\rangle$	$U_3U_1U_0\,\lvert\Psi_{2,1}\rangle$	$=$			$U_3U_1U_0\,\lvert\Psi_{2,1}\rangle$
$\lvert 0\rangle$	$\lvert 1\rangle$	$U_3U_0U_1\,\lvert\Psi_{2,1}\rangle$	$=$		$\omega^{1\cdot y}$	$U_3U_1U_0\,\lvert\Psi_{2,1}\rangle$
$\lvert 1\rangle$	$\lvert 0\rangle$	$U_1U_0U_3\,\lvert\Psi_{2,1}\rangle$	$=$	$\omega^{12\cdot y}$		$U_3U_1U_0\,\lvert\Psi_{2,1}\rangle$
$\lvert 1\rangle$	$\lvert 1\rangle$	$U_0U_1U_3\,\lvert\Psi_{2,1}\rangle$	$=$	$\omega^{12\cdot y}\cdot$	$\omega^{1\cdot y}$	$U_3U_1U_0\,\lvert\Psi_{2,1}\rangle$

Table 4.3.: Final state of the target system $|\Psi_{2,1}\rangle$ for all combinations of the control qubits $|c^x_{3,1}\rangle$ and $|c^x_{1,1}\rangle$.

For instance, the qubit $|c^x_{3,1}\rangle$ controls whether the gate U_3 is applied before the two gates U_0 and U_1, or after these two gates. Whenever $|c^x_{3,1}\rangle = |1\rangle$, U_3 needs to be commuted with two unitaries of a smaller index (independent of the order of these two gates). In this way, a factor of $\omega^{3!\cdot y}$ is picked up twice, which multiplies together to $\omega^{12\cdot y}$. Independently of this, $|c^x_{1,1}\rangle$ controls the order of U_1 and U_0. If $|c^x_{1,1}\rangle = |1\rangle$, U_1 needs to be commuted with U_0 and a relative phase of $\omega^{1!\cdot y}$ is picked up.

Analogous arguments hold for the other systems; the control qubits control the order in which the involved gates are applied on the target system. Due to the pairwise commutation relations, the final state of each target system can always be rewritten in a form that is independent of the control system. By doing so, we pick up a phase $\omega^{(\cdot)\cdot y}$ (the expression in brackets at the end of each line of a control qubit in Fig. 4.7), whenever a control qubit is in the state $|1\rangle$. For example, if $x = 16$, which is represented by $c^{16}_{3,1} = c^{16}_{2,1} = c^{16}_{2,2} = 1$ and $c^{16}_{3,2} = c^{16}_{1,1} = c^{16}_{1,2} = 0$, the circuit realizes the following transformation:

$$\begin{aligned}
T^{\mathrm{FPP}}_4 & \left|c^{16}_{3,1} = 1\right\rangle \otimes \left|c^{16}_{3,2} = 0\right\rangle \otimes \left|c^{16}_{2,1} = 1\right\rangle \otimes \left|c^{16}_{2,2} = 1\right\rangle \\
&\otimes \left|c^{16}_{1,1} = 0\right\rangle \otimes \left|c^{16}_{1,2} = 0\right\rangle \otimes |\Psi_{2,1}\rangle \otimes |\Psi_{2,2}\rangle \\
&\otimes |\Psi_{4,1}\rangle \otimes |\Psi_{4,2}\rangle \otimes |\Psi_{4,3}\rangle \otimes |\Psi_{4,4}\rangle \\
\\
= & \left|c^{16}_{3,1} = 1\right\rangle \otimes \left|c^{16}_{3,2} = 0\right\rangle \otimes \left|c^{16}_{2,1} = 1\right\rangle \otimes \left|c^{16}_{2,2} = 1\right\rangle \\
&\otimes \left|c^{16}_{1,1} = 0\right\rangle \otimes \left|c^{16}_{1,2} = 0\right\rangle \otimes U_1U_0U_3 |\Psi_{2,1}\rangle \otimes U_0U_2 |\Psi_{2,2}\rangle \\
&\otimes U_1U_0 |\Psi_{4,1}\rangle \otimes U_0U_2 |\Psi_{4,2}\rangle \otimes U_3U_0 |\Psi_{4,3}\rangle \otimes U_0 |\Psi_{4,4}\rangle \\
\\
= & \, \omega^{16\cdot y} \; \left|c^{16}_{3,1} = 1\right\rangle \otimes \left|c^{16}_{3,2} = 0\right\rangle \otimes \left|c^{16}_{2,1} = 1\right\rangle \otimes \left|c^{16}_{2,2} = 1\right\rangle \\
&\otimes \left|c^{16}_{1,1} = 0\right\rangle \otimes \left|c^{16}_{1,2} = 0\right\rangle \otimes U_3U_1U_0 |\Psi_{2,1}\rangle \otimes U_2U_0 |\Psi_{2,2}\rangle \\
&\otimes U_1U_0 |\Psi_{4,1}\rangle \otimes U_2U_0 |\Psi_{4,2}\rangle \otimes U_3U_0 |\Psi_{4,3}\rangle \otimes U_0 |\Psi_{4,4}\rangle \ .
\end{aligned} \tag{4.52}$$

Here, we obtain the factor of $\omega^{16\cdot y}$ as a composition of

$$\omega^{16\cdot y} = \omega^{12\cdot y} \cdot \omega^{2\cdot y} \cdot \omega^{2\cdot y} \, . \tag{4.53}$$

The first factor $\omega^{12\cdot y}$ stems from commuting U_3 with U_1 and U_0 on $|\Psi_{2,1}\rangle$, while the two factors of $\omega^{2\cdot y}$ arise from commuting U_2 with U_0 on $|\Psi_{2,2}\rangle$ and $|\Psi_{4,2}\rangle$, respectively. Since every $x \in \{0, 1, ..., 4! - 1\}$ can be represented as in Eq. (4.51), the algorithm in Fig. 4.7 applies the following transformation for every such x:

$$\begin{aligned}
T^{\mathrm{FPP}}_4 & |x\rangle_c \otimes |\Psi_{2,1}\rangle \otimes |\Psi_{2,2}\rangle \otimes |\Psi_{4,1}\rangle \\
&\otimes |\Psi_{4,2}\rangle \otimes |\Psi_{4,3}\rangle \otimes |\Psi_{4,4}\rangle \\
= & \, \omega^{x\cdot y} \; |x\rangle_c \otimes U_3U_1U_0 |\Psi_{2,1}\rangle \otimes U_2U_0 |\Psi_{2,2}\rangle \otimes U_1U_0 |\Psi_{4,1}\rangle \\
&\otimes U_2U_0 |\Psi_{4,2}\rangle \otimes U_3U_0 |\Psi_{4,3}\rangle \otimes U_0 |\Psi_{4,4}\rangle \ .
\end{aligned} \tag{4.54}$$

Note, that the final state of the target system is independent of the control system $|x\rangle$. Hence, we can use our algorithm and the procedure introduced in Subsection 4.3.2 to solve this specific FPP. More precisely, applying a Fourier transform, the control system is initialized in an equal superposition of all $|x\rangle$. After applying the algorithm, a measurement of the control system in the Fourier basis will yield the desired value of y.

As mentioned before, the query complexity of this example is actually worse than with the simulation of the quantum-4-switch. Nevertheless, for larger n, the method that we introduce here solves this specific FPP with only $O(n \log n)$ queries. The reason for this scaling advantage comes from the fact that every number $x \in \{0, 1, ..., n! - 1\}$ can be represented with $O(n \log n)$ bits $c^x_{k,i}$ (remember that $n! \leq 2^{n \log_2 n}$). In the algorithms presented here, every such bit corresponds to a control qubit and only two queries couple to every control qubit.

As a remark, note that the target system $|\Psi_{4,4}\rangle$ is technically redundant. Moreover, the target system $|\Psi_{4,1}\rangle$ and the corresponding control qubit $\left|c^x_{1,2}\right\rangle$ are also not needed since the five bits $c^x_{3,1}$, $c^x_{3,2}$, $c^x_{2,1}$, $c^x_{2,2}$ and $c^x_{1,1}$ are sufficient to represent every number $x \in \{0, 1, ..., 4! - 1\}$ in the form given in Eq. (4.51). In this specific example for $n = 4$, we kept them in for completeness. It turns out that for every n, there are some target systems that can be left out. In order to keep the notation as simple as possible, and since this does not affect the overall scaling, we refrain from doing so.

4.5.2. The algorithm for every n

In this section, we show how the idea of the above example can be generalized to solve this specific FPP for arbitrary n. As above, it is convenient to introduce a specific representation of the control state $|x\rangle$ into qubits. More precisely, we use $(n-1) \cdot \lceil \log_2 n \rceil$ control qubits $\left|c^x_{k,i}\right\rangle$ ($c^x_{k,i} \in \{0, 1\}$) where $k = 1, 2, ..., n - 1$ and $i = 1, 2, ..., \lceil \log_2 n \rceil$. For convenience, we define $\hat{\imath} := \lceil \log_2 n \rceil$. The state $|x\rangle_c$ is identified with

$$|x\rangle_c = \bigotimes_{\substack{1 \leq k \leq n-1 \\ 1 \leq i \leq \lceil \log_2 n \rceil}} \left|c^x_{k,i}\right\rangle , \tag{4.55}$$

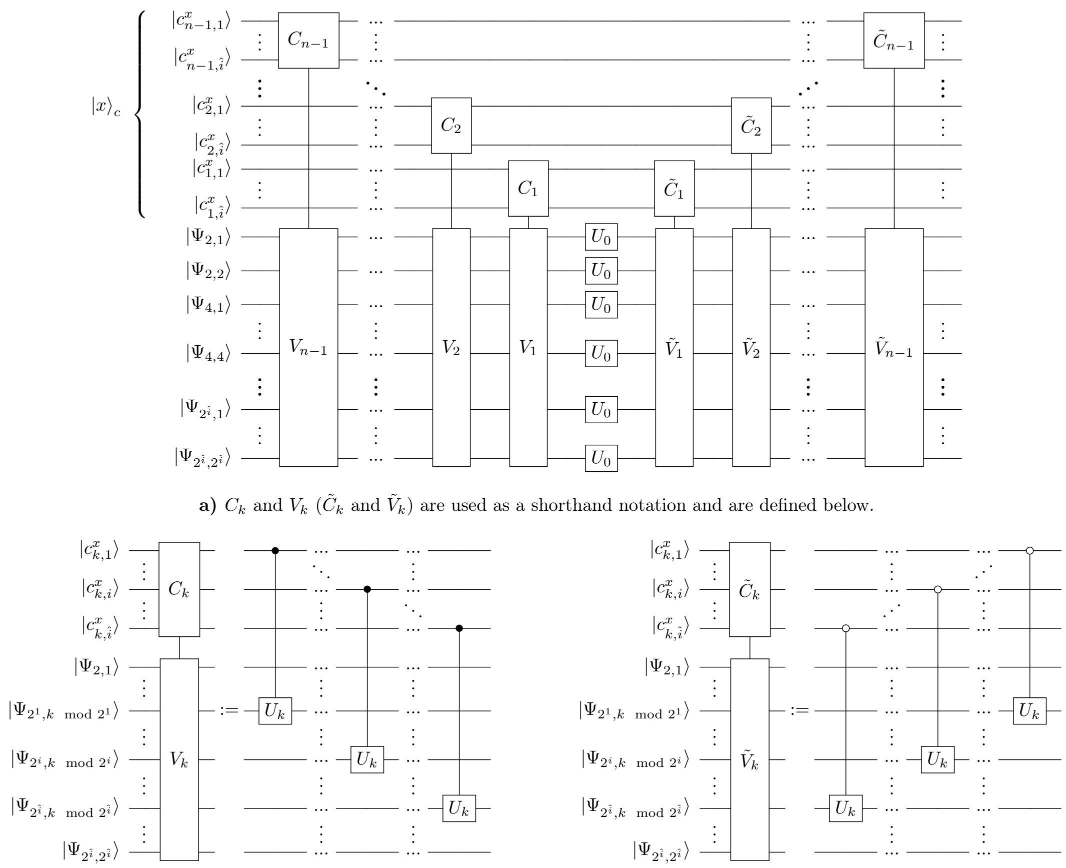
where the bits $c^x_{k,i}$ satisfy the equation

$$x = \sum_{k=1}^{n-1} \sum_{i=1}^{\hat{\imath}} c^x_{k,i} \cdot \left\lceil \frac{k}{2^i} \right\rceil \cdot k! \,. \tag{4.56}$$

While the motivation for this basis will become clearer below, we give the formal proof that every $x \in \{0, 1, ..., n! - 1\}$ can be represented in this way in Appendix 4.A. [8]

Now, we will show that the quantum circuit given in Fig. 4.8 solves this specific FPP with $O(n \log n)$ queries. The control system consists of the $(n - 1) \cdot \lceil \log_2 n \rceil$ control qubits $\left|c^x_{k,i}\right\rangle$ introduced above. Furthermore, we use several target systems $\left|\Psi_{2^i,j}\right\rangle$, where $i \in \{1, 2, ..., \hat{\imath} := \lceil \log_2 n \rceil\}$ and $j \in \{1, 2, 3, ..., 2^i\}$. As for every algorithm in this article, they are initialized in an arbitrary d-dimensional state.

[8] Note that the representation of x in this basis is not unique. For our purpose, it is enough to pick one such representation for every x. Furthermore, this representation is related to the factorial number system that is used to label the permutations Π_x in Eq. (4.46). More precisely, $c^x_{k,i} \cdot \lceil \frac{k}{2^i} \rceil$ is a representation of a_k with $\lceil \log_2 n \rceil$ bits.



a) C_k and V_k ($\tilde{C}_k$ and $\tilde{V}_k$) are used as a shorthand notation and are defined below.

b) Definition of C_k and V_k (The black dot denotes a control on $|1\rangle$, in symbols: $|0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes U_k$.)

c) Definition of $\tilde{C}_k$ and $\tilde{V}_k$ (The white dot denotes a control on $|0\rangle$, in symbols: $|0\rangle\langle 0| \otimes U_k + |1\rangle\langle 1| \otimes \mathbb{1}$.)

Figure 4.8.: The quantum circuit implementing the transformation T_n^{FPP} for the FPP with the factoradic labeling of the permutations: Depending on the state of the control qubits, the gates are applied on the target systems in a certain order. Due to the pairwise commutation relations, the final state of each target system can always be reordered but certain phases are picked up when two unitaries are commuted. For every $x \in \{0, 1, ..., n! - 1\}$, these phases multiply together to $\omega^{x \cdot y}$ (see main text).

The idea of the algorithm is, as usual, that the black-box gates act on the target systems in a certain order and due to the pairwise commutation relations certain phases are picked up by rewriting the final state of each target system. More precisely, we will show that whenever a control qubit $\left|c^x_{k,i}\right\rangle$ is in the state $|1\rangle$, we obtain a relative phase of $\omega^{\left\lceil \frac{k}{2^i} \right\rceil \cdot k! \cdot y}$, independent of the states of the other control qubits. For every $x \in \{0, 1, ..., n! - 1\}$ these phases multiply together to $\omega^{x \cdot y}$:

$$\begin{aligned}
&T_n^{\mathrm{FPP}} \left|x\right\rangle_c \otimes \left(\bigotimes_{\substack{1 \le i \le \hat{\imath} \\ 1 \le j \le 2^i}} \left|\Psi_{2^i,j}\right\rangle \right) \\
&= T_n^{\mathrm{FPP}} \left(\bigotimes_{\substack{1 \le k \le n-1 \\ 1 \le i \le \hat{\imath}}} \left|c_{k,i}^x\right\rangle \right) \otimes \left(\bigotimes_{\substack{1 \le i \le \hat{\imath} \\ 1 \le j \le 2^i}} \left|\Psi_{2^i,j}\right\rangle \right) \\
&= \left(\prod_{\substack{1 \le k \le n-1 \\ 1 \le i \le \hat{\imath}}} \omega^{c_{k,i}^x \cdot \left\lceil \frac{k}{2^i} \right\rceil \cdot k! \cdot y} \right) \cdot \left(\bigotimes_{\substack{1 \le k \le n-1 \\ 1 \le i \le \hat{\imath}}} \left|c_{k,i}^x\right\rangle \right) \otimes \left(\bigotimes_{\substack{1 \le i \le \hat{\imath} \\ 1 \le j \le 2^i}} (\ldots U_{2^i+j} U_j U_0) \left|\Psi_{2^i,j}\right\rangle \right) \\
&= \omega^{x \cdot y} \left|x\right\rangle_c \otimes \left(\bigotimes_{\substack{1 \le i \le \hat{\imath} \\ 1 \le j \le 2^i}} (\ldots U_{2^i+j} U_j U_0) \left|\Psi_{2^i,j}\right\rangle \right) .
\end{aligned} \tag{4.57}$$

Here, we used the representation of x in the basis given in Eq. (4.56) to show that all the phases accumulate to:

$$\prod_{\substack{1 \le k \le n-1 \\ 1 \le i \le \hat{\imath}}} \omega^{c_{k,i}^x \cdot \left\lceil \frac{k}{2^i} \right\rceil \cdot k! \cdot y} = \omega^{\left(\sum_{k=1}^{n-1} \sum_{i=1}^{\hat{\imath}} c_{k,i}^x \cdot \left\lceil \frac{k}{2^i} \right\rceil \cdot k! \cdot y \right)} = \omega^{x \cdot y} . \tag{4.58}$$

One can observe that the target system becomes independent of the control system. Hence, if the control system is initialized in an equal superposition of all x, the circuit applies, by linearity, the transformation

$$\begin{aligned}
&\left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \left|x\right\rangle_c \right) \otimes \left(\bigotimes_{\substack{1 \le i \le \hat{\imath} \\ 1 \le j \le 2^i}} \left|\Psi_{2^i,j}\right\rangle \right) \\
\mapsto &\left(\frac{1}{\sqrt{n!}} \sum_{x=0}^{n!-1} \omega^{x \cdot y} \left|x\right\rangle_c \right) \otimes \left(\bigotimes_{\substack{1 \le i \le \hat{\imath} \\ 1 \le j \le 2^i}} (\ldots U_{2^i+j} U_j U_0) \left|\Psi_{2^i,j}\right\rangle \right)
\end{aligned} \tag{4.59}$$

and all target systems factorize out at the end of the algorithm. After applying the inverse Fourier transform to the control system, the correct value of y can be read out with a measurement of the control system in the computational basis. For a better understanding of the algorithm, in addition to the example of $n = 4$ in the last subsection, we give the circuit for $n = 8$ in Appendix 4.C (Fig. 4.14).

How the algorithm works

To show that this algorithm realizes the desired transformation in Eq. (4.57), we focus on one target system $\left|\Psi_{2^i,j}\right\rangle$ and all the gates that act on it. These are exactly the gates U_k that satisfy $k \equiv j \pmod{2^i}$. The order in which these gates are applied on the target system $\left|\Psi_{2^i,j}\right\rangle$ depends on the states of the corresponding control qubits $\left|c^x_{k,i}\right\rangle$ (see Fig. 4.9).

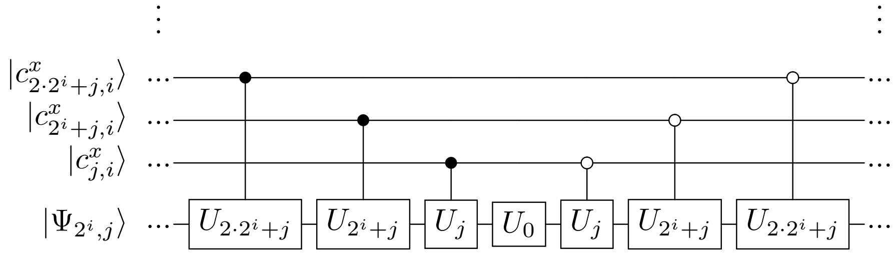


Figure 4.9.: Each control qubit $\left|c^x_{k,i}\right\rangle$ controls whether the gate U_k (with $k \equiv j \pmod{2^i}$) is applied either before or after an entire block of unitaries with a smaller index. For example, if $\left|c^x_{2^i+j,i} = 1\right\rangle$ the gate U_{2^i+j} is applied before U_0 and U_j, while if $\left|c^x_{2^i+j,i} = 0\right\rangle$ it is applied after these two gates. Rewriting the final state leads to a factor of $\omega^{2\cdot(2^i+j)!\cdot y}$ whenever $\left|c^x_{2^i+j,i} = 1\right\rangle$.

On the one hand, if all control qubits are in the state $|0\rangle$, U_0 acts first, then U_j and so on. This way, the final state of the target system becomes $(...U_{2\cdot 2^i+j}U_{2^i+j}U_jU_0)\left|\Psi_{2^i,j}\right\rangle$. On the other hand, if some of the control qubits are in the state $|1\rangle$, the gates act in a different order. The structure of the algorithm is chosen such that a gate is either applied immediately before or after an entire block of unitaries with a smaller index. By rewriting the final state into the form $(...U_{2\cdot 2^i+j}U_{2^i+j}U_jU_0)\left|\Psi_{2^i,j}\right\rangle$, a unitary has to be commuted either with all unitaries within this block or with none of them. One can check that every gate U_k that appears on the target system has to be commuted with $\left\lceil \frac{k}{2^i} \right\rceil$ unitaries of smaller index if and only if $\left|c^x_{k,i} = 1\right\rangle$, leading to an additional factor of $\omega^{\left\lceil \frac{k}{2^i} \right\rceil \cdot k! \cdot y}$. This shows that the algorithm realizes the transformation given in Eq. (4.57).

Query complexity

To count the total number of black-box unitaries that are used in this algorithm, we observe that for every $k \in \{1, 2, ..., n-1\}$, the gate U_k appears only in V_k and $\tilde{V}_k$. In each of them, it is used exactly $\hat{i} = \lceil \log_2 n \rceil$ times. In addition, the gate U_0 acts on each target system exactly once. In total, there are

$$2 + 4 + 8 + ... + 2^{\lceil \log_2 n \rceil} = 2^{\lceil \log_2 n \rceil + 1} - 2 \tag{4.60}$$

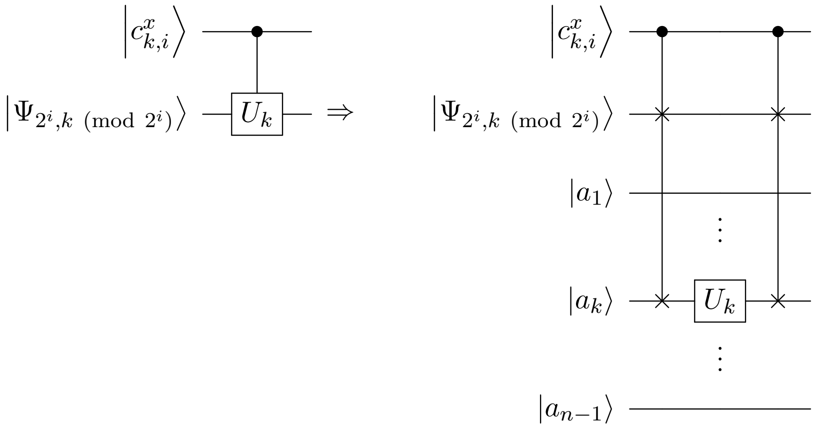
target systems. To see this, remember that the target systems are labeled with $\left|\Psi_{2^i,j}\right\rangle$, where $i = 1, 2, ..., \lceil \log_2 n \rceil$ and $j = 1, 2, 3, ..., 2^i$. Hence, the total number of black-box calls is

$$\begin{aligned} Q &= 2 \cdot (n-1) \cdot \lceil \log_2 n \rceil + 2^{\lceil \log_2 n \rceil + 1} - 2 \\ &< 2 \cdot (n-1) \cdot (\log_2 n + 1) + 2^{\log_2 n + 2} - 2 \\ &= 2 \cdot (n-1) \cdot (\log_2 n + 1) + 4 \cdot n - 2\,. \end{aligned} \tag{4.61}$$

We conclude that the query complexity of this algorithm is $O(n \log n)$.

Control of unknown unitaries

In this circuit, we control unknown unitaries. This operation is not well-defined within the standard quantum circuit model [179]. Nevertheless, one can circumvent this issue by introducing auxiliary systems. More precisely, for every $k \in \{1, 2, ..., n-1\}$, we add an auxiliary system $|a_k\rangle$ initialized in an arbitrary d-dimensional state. Whenever an unknown unitary U_k, controlled on $\left|c^x_{k,i}\right\rangle$, shall be applied on $\left|\Psi_{2^i,k \pmod{2^i}}\right\rangle$, we perform instead a controlled swap of $\left|\Psi_{2^i,k \pmod{2^i}}\right\rangle$ and $|a_k\rangle$:


$\left|c^x_{k,i}\right\rangle$
$\left|\Psi_{2^i,k \pmod{2^i}}\right\rangle$
U_k
⇒
$\left|c^x_{k,i}\right\rangle$
$\left|\Psi_{2^i,k \pmod{2^i}}\right\rangle$
$|a_1\rangle$
$|a_k\rangle$
U_k
$|a_{n-1}\rangle$

If the control qubit is in the state $\left|c^x_{k,i} = 1\right\rangle$, the two systems are swapped and the gate U_k is applied on the target system. On the other hand, if the control qubit is in the state $\left|c^x_{k,i} = 0\right\rangle$, the two systems are not swapped and the gate U_k is applied on the auxiliary system $|a_k\rangle$ instead. In the case where the gates are applied conditioned on

$\left|c^x_{k,i}=0\right\rangle$ (for the boxes $\tilde{V}$), the swaps are also conditioned on $\left|c^x_{k,i}=0\right\rangle$ (all black dots are replaced by white dots).

It is important to ensure that this replacement does not affect the functionality of our algorithm. To see that this is true, note that for every $i \in \{1,2,...,\hat{i}\}$, the gate U_k acts exactly once on the target system $\left|\Psi_{2^i,k \pmod{2^i}}\right\rangle$ and once on the auxiliary system $|a_k\rangle$, independent of the state of the control qubit $\left|c^x_{k,i}\right\rangle$. (The control qubit controls only if it is first applied on the target system and thereafter on the auxiliary system or vice versa.) In total, each auxiliary system ends up in the state $(U_k)^{\hat{i}}\,|a_k\rangle$, independent of the state of the control system. In this way, the auxiliary systems factorize out when the control system is initialized in a superposition of all states $x \in \{0,1,...,n!-1\}$ and do not affect the outcome of the measurement of the control system at the very end of the algorithm.

4.5.3. $O(n\log n)$ causal algorithms for every FPP?

A natural question that appears is whether it is possible to solve other specific FPPs with $O(n\log n)$ queries as well. The algorithms that we presented in this section can only be used for this specific labeling of the permutations, since we explicitly use the relations $U_jU_k = \omega^{k!\cdot y}\, U_kU_j$. If the permutations are labeled differently, the pairwise phases will change and the above algorithm cannot be used directly. Nevertheless, we think that the structure of our algorithm can be adapted to solve other specific FPPs as well. The idea that a certain phase $\omega^{\phi(k,i)\cdot y}$ is picked up whenever a control qubit $\left|c^x_{k,i}\right\rangle$ is in the state $|1\rangle$ (by using a structure as in Fig. 4.9) can be used for different pairwise commutation relations as well. If every $x \in \{0,1,...,n!-1\}$ can be written as

$$x = \sum_{k,i} c^x_{k,i}\cdot\phi(k,i) \tag{4.62}$$

for some bits $c^x_{k,i} \in \{0,1\}$, we can use the control qubits $\left|c^x_{k,i}\right\rangle$ as the control system $|x\rangle_c$:

$$|x\rangle_c = \bigotimes_{k,i} \left|c^x_{k,i}\right\rangle\ . \tag{4.63}$$

By initializing the control system in an equal superposition of all $x \in \{0,1,...,n!-1\}$, such an algorithm will apply the transformation

$$\left(\frac{1}{\sqrt{n!}}\sum_{x=0}^{n!-1}|x\rangle_c\right)\otimes\bigotimes_{i=1}^{m}|\Psi_i\rangle \mapsto \left(\frac{1}{\sqrt{n!}}\sum_{x=0}^{n!-1}\omega^{x\cdot y}\,|x\rangle_c\right)\otimes\bigotimes_{i=1}^{m} f_0^i(U_0,...,U_{n-1})\,|\Psi_i\rangle\ , \tag{4.64}$$

where for every x, the phase $\omega^{x\cdot y}$ is obtained as the product of:

$$\omega^{x\cdot y} = \prod_{k,i}\omega^{c^x_{k,i}\cdot\phi(k,i)\cdot y}\ . \tag{4.65}$$

Here, the equality follows from Eq. (4.62). Again, the solution y can be read out after applying the inverse Fourier transform to the control system.

Since every number $x \in \{0, 1, ..., n! - 1\}$ can in principle be represented with $O(n \log n)$ bits ($n! \leq 2^{n \log_2 n}$) and since two queries couple to every control qubit, it seems likely that every FPP can be solved with $O(n \log n)$ queries. The crucial point is whether it is possible to find an implementation as in Fig. 4.9, a combination of gates and target systems such that this can be done efficiently. The disadvantage of this procedure is that it requires some rather involved combinatorics and that one has to adapt this algorithm by hand. While it remains open whether this is always possible, we present in the next section an algorithm that can solve *every* FPP with $O(n\sqrt{n})$ queries, independent of the labeling of the permutations.

4.6. A causal algorithm that solves every FPP with $O(n\sqrt{n})$ queries

In this section, we present an algorithm that solves every Fourier promise problem with $O(n\sqrt{n})$ queries. The main idea is based on the fact that the existence of pairwise commutation relations ($U_j U_k = \alpha_{jk}\ U_k U_j$) allows us to rewrite every permutation $\Pi_x = U_{\sigma_x(n-1)}...U_{\sigma_x(1)}U_{\sigma_x(0)}$ into:

$$U_{\sigma_x(n-1)}...U_{\sigma_x(1)}U_{\sigma_x(0)} = \alpha_x\ U_{n-1}...U_1 U_0\,, \tag{4.66}$$

where the total phase α_x is a product of pairwise phases α_{jk}. We use this fact to decompose the total phase of every permutation into different factors and simulate each factor on a different target system. So instead of simulating every permutation Π_x on its own (which requires the simulation of the (full) n-switch and hence $O(n^2)$ queries), we construct other expressions that can simulate these factors and call only $O(n\sqrt{n})$ gates in total. Via the "phase-kickback," all these factors accumulate in the control system and multiply together to the total phase α_x.

More precisely, we decompose every permutation $\Pi_x = U_{\sigma_x(n-1)}...U_{\sigma_x(1)}U_{\sigma_x(0)}$ into blocks of length $\hat{n} := \lceil\sqrt{n}\rceil$ (the last block contains all remaining unitaries and is usually shorter). The number of blocks obtained in this way is $\hat{k} := \lceil\frac{n}{\hat{n}}\rceil$. Formally, we define:

Definition 4.3. *For every permutation*

$$\Pi_x = (U_{\sigma_x(n-1)}...U_{\sigma_x((\hat{k}-1)\cdot\hat{n})})...\ (U_{\sigma_x(2\cdot\hat{n}-1)}...U_{\sigma_x(\hat{n})})\ (U_{\sigma_x(\hat{n}-1)}...U_{\sigma_x(0)})$$

(the organization into blocks is merely for illustrative reasons) of the n unitaries, let

$$\Pi_{xk} := \begin{cases} [U_{\sigma_x(n-1)} ... U_{\sigma_x(\hat{n})}] \; (U_{\sigma_x(\hat{n}-1)} ... U_{\sigma_x(0)}) & k=0 \\ [U_{\sigma_x(n-1)} ... U_{\sigma_x((k+1)\cdot\hat{n})}] \; (U_{\sigma_x((k+1)\cdot\hat{n}-1)} ... U_{\sigma_x(k\cdot\hat{n})}) \; [U_{\sigma_x(k\cdot\hat{n}-1)} ... U_{\sigma_x(0)}] & k=1,...,\hat{k}-2 \\ (U_{\sigma_x(n-1)} ... U_{\sigma_x((\hat{k}-1)\cdot\hat{n})}) \; [U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)} ... U_{\sigma_x(0)}] & k=\hat{k}-1 \end{cases} \tag{4.67}$$

$$\tilde{\Pi}^r_{xk} := \begin{cases} \{U_{\sigma_x(k\cdot\hat{n}-1)} ... U_{\sigma_x(0)}\} \; \{U_{\sigma_x(n-1)} ... U_{\sigma_x(k\cdot\hat{n})}\} & k=1,...,\hat{k}-2 \\ \{U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)} ... U_{\sigma_x(0)}\} \; \{U_{\sigma_x(n-1)} ... U_{\sigma_x((\hat{k}-1)\cdot\hat{n})}\} & k=\hat{k}-1 \end{cases} \tag{4.68}$$

where $[U_{i_1}U_{i_2}...U_{i_j}]$ is defined to be the descending ordering of the unitaries $U_{i_1}U_{i_2}...U_{i_j}$, while $\{U_{i_1}U_{i_2}...U_{i_j}\}$ is the ascending ordering of them and $(U_{i_1}U_{i_2}...U_{i_j}) = U_{i_1}U_{i_2}...U_{i_j}$ leaves the string invariant.

As an example, consider $n = 9$ and $\Pi_x = (U_3U_5U_8)\;(U_0U_2U_7)\;(U_4U_6U_1)$:

$$\begin{aligned} \Pi_{x0} &= (U_8U_7U_5U_3U_2U_0)\;(U_4U_6U_1)\,, \\ \Pi_{x1} &= (U_8U_5U_3)\;(U_0U_2U_7)\;(U_6U_4U_1)\,, \\ \Pi_{x2} &= (U_3U_5U_8)\;(U_7U_6U_4U_2U_1U_0)\,, \\ \\ \tilde{\Pi}^r_{x1} &= (U_1U_4U_6)\;(U_0U_2U_3U_5U_7U_8) \text{ and} \\ \tilde{\Pi}^r_{x2} &= (U_0U_1U_2U_4U_6U_7)\;(U_3U_5U_8)\,. \end{aligned} \tag{4.69}$$

They are defined in a way that Π_{xk} and $\tilde{\Pi}^r_{xk}$ simulate exactly all pairwise phases α_{jk} for the unitaries within the block $(U_{\sigma_x((k+1)\cdot\hat{n}-1)}...U_{\sigma_x(k\cdot\hat{n})})$. If they act on different target systems $|\Psi_k\rangle$ and $|\Phi_k\rangle$ respectively, all the pairwise phases are accumulated and we obtain as the product the total phase of the original permutation Π_x:

Lemma 4.1. *For every set of (d-dimensional) unitaries $\{U_i\}_0^{n-1}$ that satisfy pairwise commutation relations, the following relation holds:*

$$\bigotimes_{k=0}^{\hat{k}-1} \Pi_{xk} |\Psi_k\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \tilde{\Pi}^r_{xk} |\Phi_k\rangle = \Pi_x |\Psi_0\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \Pi |\Psi_k\rangle \otimes \Pi^r |\Phi_k\rangle \ . \tag{4.70}$$

Here, $\Pi := U_{n-1}...U_1U_0$ denotes the descending order of all unitaries, $\Pi^r := U_0U_1...U_{n-1}$ denotes the ascending order of all unitaries and $|\Psi_k\rangle, |\Phi_k\rangle \in \mathcal{H}^d$ are arbitrary d-dimensional states.

Proof. See Appendix 4.B. □

It turns out that the permutations Π_{xk} and $\tilde{\Pi}^r_{xk}$, due to the fact that a large part of each of them is already ordered, can be simulated with a causal algorithm and $O(n\sqrt{n})$ queries. The algorithm that achieves this is presented in the next subsection.

In the algorithm, $(2 \cdot \hat{k} - 1)$ target systems, denoted as $|\Psi_k\rangle$ and $|\Phi_k\rangle$, as well as n auxiliary systems $|a_i\rangle$ are used. All of them are initialized in an arbitrary d-dimensional state. The control system is a system of at least $n!$ dimensions and the algorithm applies, depending on the state $|x\rangle$ of the control system, the permutations Π_{xk} on $|\Psi_k\rangle$ and the permutations $\tilde{\Pi}^r_{xk}$ on $|\Phi_k\rangle$. All the remaining gates U_i act on the corresponding auxiliary system $|a_i\rangle$. In this way, the algorithm realizes the transformation

$$\begin{aligned} T_n^{\mathrm{FPP}} & |x\rangle_c \otimes \bigotimes_{k=0}^{\hat{k}-1} |\Psi_k\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} |\Phi_k\rangle \otimes \bigotimes_{i=0}^{n-1} |a_i\rangle \\ &= |x\rangle_c \otimes \bigotimes_{k=0}^{\hat{k}-1} \Pi_{xk} |\Psi_k\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \tilde{\Pi}^r_{xk} |\Phi_k\rangle \otimes \bigotimes_{i=0}^{n-1} (U_i)^{k_i} |a_i\rangle \\ &= |x\rangle_c \otimes \Pi_x |\Psi_0\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \Pi |\Psi_k\rangle \otimes \Pi^r |\Phi_k\rangle \otimes \bigotimes_{i=0}^{n-1} (U_i)^{k_i} |a_i\rangle \ . \end{aligned} \tag{4.71}$$

Here, $k_i = \hat{n} + 2 \cdot \hat{k} - 3$ is a constant that only depends on n and Lemma 4.1 is used to rewrite the state in the second step. Except for the first target system $|\Psi_0\rangle$, which ends up in the state $\Pi_x |\Psi_0\rangle$, the final state of each target and auxiliary system is independent of x and we conclude that this algorithm simulates the action of the n-switch for unitaries satisfying pairwise commutation relations and is therefore able to solve every Fourier promise problem. More precisely, as described in Subsection 4.3.2, with a Fourier transform, the control system is initialized in an equal superposition of all states $x \in \{0, 1, ..., n! - 1\}$ and after the algorithm is applied, the solution y can be read out with a measurement in the Fourier basis.[9] Note that at no point we refer to a specific labeling of the permutations Π_x and hence, we can solve every possible Fourier promise problem with this causal algorithm. The advantage stems merely from the fact that every set of unitaries that satisfies the promise also satisfies pairwise commutation relations which imply that Lemma 4.1 holds.

4.6.1. The quantum algorithm

Here we present the quantum circuit that realizes the transformation described in the last subsection (Eq. (4.71)) and show that this algorithm uses $O(n\sqrt{n})$ queries. To keep the procedure as clear as possible, we divide the quantum circuit into three parts.

Part 1

First, all target systems $|\Psi_k\rangle$ undergo the transformations

$$\forall\ 1 \leq k \leq \hat{k} - 1 : \ |\Psi_k\rangle \mapsto [U_{\sigma_x(k \cdot \hat{n} - 1)} ... U_{\sigma_x(0)}] \, |\Psi_k\rangle \ . \tag{4.72}$$

[9] For the comparison with Eq. (4.20), note that by using the promise $\Pi_x = \omega^{x \cdot y}\, \Pi_0$, the term $|x\rangle_c \otimes \Pi_x |\Psi_0\rangle$ in Eq. (4.71) can be rewritten into $\omega^{x \cdot y}\ |x\rangle_c \otimes \Pi_0 |\Psi_0\rangle$.

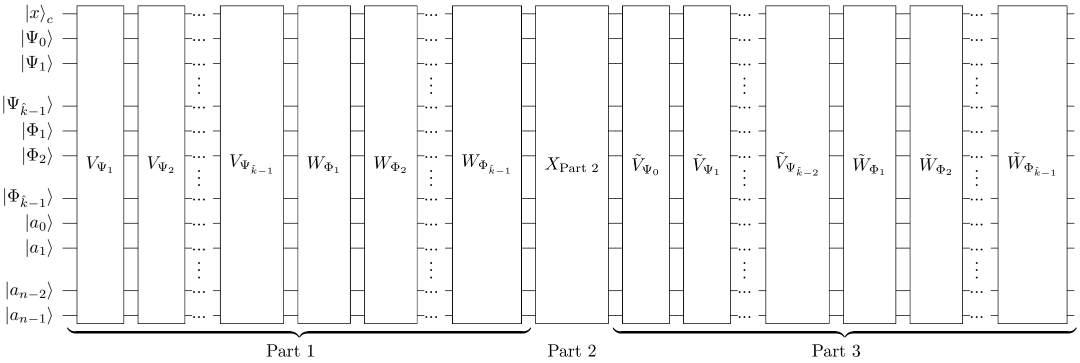


Figure 4.10.: The quantum algorithm that implements T_n^{FPP} for every FPP with $O(n\sqrt{n})$ queries. The circuit is decomposed into three parts that are explained below.

For each $|\Psi_k\rangle$, this is realized by the algorithm V_{Ψ_k} given in Fig. 4.11. Here, depending on the state $|x\rangle$, in each step $i = 0, 1, ..., n-1$, the target system $|\Psi_k\rangle$ is swapped with $|a_i\rangle$ if and only if U_i is contained in $[U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}]$.

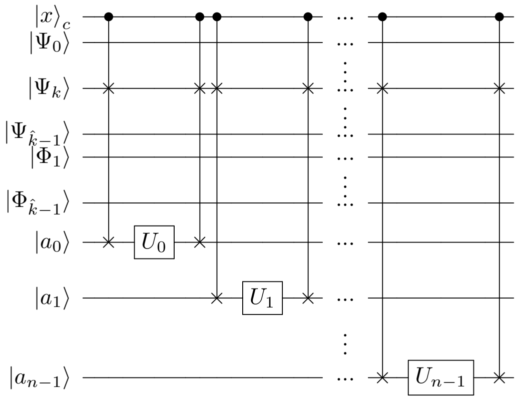


Figure 4.11.: Implementation of V_{Ψ_k}

To understand why this circuit realizes the above transformation, note that the block $[U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}]$ is by construction a block of $k\cdot\hat{n}$ unitaries in descending order and the unitary with the smallest index has to be applied first. By going step by step through each of the n possible unitaries U_0 until U_{n-1}, exactly those unitaries contained in the ordered block are applied on the target system $|\Psi_k\rangle$ and all the others are applied on the corresponding auxiliary system. Each of these unitaries V_{Ψ_k} consumes n queries.

Similarly, the transformations

$$\forall\ 1 \leq k \leq \hat{k}-1:\ |\Phi_k\rangle \mapsto \{U_{\sigma_x(n-1)}...U_{\sigma_x(k\cdot\hat{n})}\}\,|\Phi_k\rangle \tag{4.73}$$

are realized with the algorithm W_{Φ_k} given by the circuit in Fig. 4.12. Here, in each step, the target system $|\Phi_k\rangle$ is swapped with $|a_i\rangle$ if and only if the gate U_i is contained in $\{U_{\sigma_x(n-1)}...U_{\sigma_x(k\cdot\hat{n})}\}$.

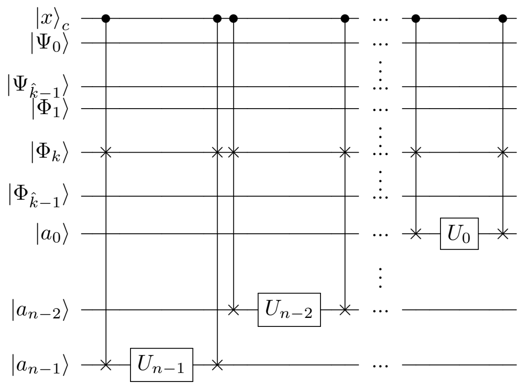


Figure 4.12.: Implementation of W_{Φ_k}

The only difference is that the unitaries in this block are arranged in ascending order and the unitary with the highest index is applied first. Each of these W_{Φ_k} consumes again n queries.

Part 2

In the second part, we realize the transformations:

$$\forall\ 0 \leq k \leq \hat{k} - 2 :$$
$$[U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}]\,|\Psi_k\rangle \mapsto (U_{\sigma_x((k+1)\cdot\hat{n}-1)}...U_{\sigma_x(k\cdot\hat{n})})[U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}]\,|\Psi_k\rangle$$

and for $k = \hat{k} - 1$:

$$[U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)}...U_{\sigma_x(0)}]\left|\Psi_{\hat{k}-1}\right\rangle \mapsto (U_{\sigma_x(n-1)}...U_{\sigma_x((\hat{k}-1)\cdot\hat{n})})[U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)}...U_{\sigma_x(0)}]\left|\Psi_{\hat{k}-1}\right\rangle \tag{4.74}$$

with the algorithm in Fig. 4.13. In each step $i = 0, 1, ..., \hat{n} - 1$, every $|\Psi_k\rangle$ is swapped with the auxiliary system $\left|a_{\sigma_x(k\cdot\hat{n}+i)}\right\rangle$. In this way, $U_{\sigma_x(k\cdot\hat{n}+i)}$ acts on $|\Psi_k\rangle$, and afterwards $|\Psi_k\rangle$ and $\left|a_{\sigma_x(k\cdot\hat{n}+i)}\right\rangle$ are swapped back. After $\hat{n}$ steps, the entire block $(U_{\sigma_x((k+1)\cdot\hat{n}-1)}...U_{\sigma_x(k\cdot\hat{n})})$ is applied on each target system $|\Psi_k\rangle$. (For $k = \hat{k} - 1$, we already stop after the step in which $U_{\sigma_x(n-1)}$ is applied on $\left|\Psi_{\hat{k}-1}\right\rangle$.)

Note that the target systems $|\Phi_k\rangle$ are unaffected by this part of the algorithm. In each of the $\hat{n}$ steps, n queries are consumed. This part of the algorithm is similar to the algorithm presented in Subsection 4.2.3. The difference is that we swap several target systems simultaneously, instead of only one.

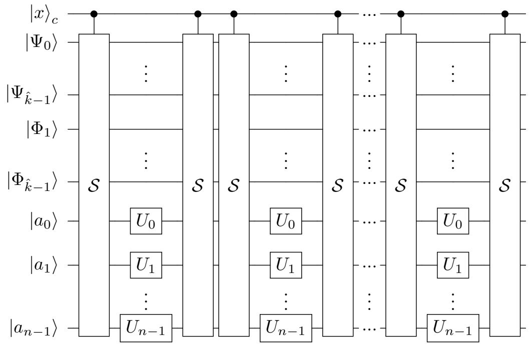


Figure 4.13.: Implementation of $X_{\text{Part 2}}$

Part 3

Finally, the remaining blocks $[U_{\sigma_x(n-1)}...U_{\sigma_x((k+1)\cdot\hat{n})}]$ need to be applied on the target systems $|\Psi_k\rangle$:

$$\forall\ 0 \leq k \leq \hat{k}-2:\ (U_{\sigma_x((k+1)\cdot\hat{n}-1)}...U_{\sigma_x(k\cdot\hat{n})})[U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}]\,|\Psi_k\rangle \mapsto \Pi_{xk}\,|\Psi_k\rangle\ . \tag{4.75}$$

Similarly, the blocks $\{U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}\}$ need to be applied on $|\Phi_k\rangle$:

$$\forall\ 1 \leq k \leq \hat{k}-1:\ \{U_{\sigma_x(n-1)}...U_{\sigma_x(k\cdot\hat{n})}\}\,|\Phi_k\rangle \mapsto \tilde{\Pi}^r_{xk}\,|\Phi_k\rangle\ . \tag{4.76}$$

Both transformations are completely analogous to the procedure in Part 1 and require $2\cdot(\hat{k}-1)\cdot n$ queries in total.

Query complexity

The number of queries in Part 1 amounts to $2\cdot(\hat{k}-1)\cdot n$. For Part 2, we need $\hat{n}\cdot n$ queries, while for the last Part, $2\cdot(\hat{k}-1)\cdot n$ black-box gates are called. Summing these together gives

$$Q = (\hat{n} + 4\cdot\hat{k} - 4)\cdot n \tag{4.77}$$

queries in total. Using $\hat{n} := \lceil\sqrt{n}\rceil < \sqrt{n}+1$ and $\hat{k} := \lceil\frac{n}{\hat{n}}\rceil < \frac{n}{\hat{n}}+1 \leq \frac{n}{\sqrt{n}}+1 = \sqrt{n}+1$ we obtain:

$$Q < (5\cdot\sqrt{n}+1)\cdot n\,. \tag{4.78}$$

It is important to ensure that the auxiliary systems factorize out at the end of the algorithm. To see that this is indeed true, we observe that in each part of the algorithm a gate U_i acts either on a target system $|\Psi_k\rangle$, a target system $|\Phi_k\rangle$ or the auxiliary system

$|a_i\rangle$. All together, for every $i \in \{0, 1, ..., n-1\}$, the gate U_i appears exactly $\hat{n} + 4 \cdot \hat{k} - 4$ times in the algorithm. Furthermore, it acts, independently of the permutation Π_x, on each of the $\hat{k}$ target systems $|\Psi_k\rangle$ and each of the $\hat{k} - 1$ target systems $|\Phi_k\rangle$ exactly once. This is true since the expressions Π_{xk} and $\tilde{\Pi}_{xk}$ are by themselves permutations of the n unitaries and contain each U_i exactly once. In all other remaining instances, U_i acts on the auxiliary system $|a_i\rangle$, which therefore ends up in the state $(U_i)^{k_i} |a_i\rangle$ with $k_i = \hat{n} + 2 \cdot \hat{k} - 3$, independent of the state of the control system $|x\rangle$. In total, we have shown that the algorithm realizes the desired transformation given in Eq. (4.71) for every $x \in \{0, 1, ..., n! - 1\}$:

$$\begin{aligned} &T_n^{\mathrm{FPP}} |x\rangle_c \otimes \bigotimes_{k=0}^{\hat{k}-1} |\Psi_k\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} |\Phi_k\rangle \otimes \bigotimes_{i=0}^{n-1} |a_i\rangle \\ &= |x\rangle_c \otimes \bigotimes_{k=0}^{\hat{k}-1} \Pi_{xk} |\Psi_k\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \tilde{\Pi}^r_{xk} |\Phi_k\rangle \otimes \bigotimes_{i=0}^{n-1} (U_i)^{k_i} |a_i\rangle \ . \end{aligned} \tag{4.79}$$

Hence, we conclude that this algorithm solves every Fourier promise problem with $O(n\sqrt{n})$ queries.

4.7. Conclusion

The introduction of indefinite causal structures raised the question of the existence of computational tasks which can be solved more efficiently using these structures, compared to causally ordered protocols. Fourier promise problems were initially introduced to demonstrate that such a computational advantage exists even in the asymptotic case. The problems were shown to be solved with n queries using the quantum-n-switch and it was expected that the most efficient solution with a causal protocol requires the simulation of the quantum-n-switch and, hence, $O(n^2)$ queries. We showed that for the specific task of solving Fourier promise problems, the advantage of using the quantum-n-switch is significantly smaller than previously expected; in fact, we presented a causal quantum algorithm, within the standard quantum circuit model, which solves the same computational tasks almost as efficiently. More precisely, we presented a causal algorithm that solves a specific Fourier promise problem with $O(n \log n)$ queries and conjectured that all problems of this class can be solved with a similar efficiency. Furthermore, we presented a causal algorithm that solves every Fourier promise problem with $O(n\sqrt{n})$ queries.

We conclude that for the specific class of problems considered here, the advantage of algorithms that use a quantum-controlled ordering of gates, compared to causally ordered algorithms, is smaller than first expected. Nevertheless, although we could show that the simulation of the quantum-n-switch is not the most efficient causal algorithm for solving FPPs, it is in principle possible to construct tasks which profit more from using the quantum-n-switch. One promising class of problems is already introduced in Ref. [28]. The so-called Hadamard promise problems are a variation of the Fourier promise problems with the advantage that they are better suited for experimental realization. In Ref. [28], the authors present only one specific task with four unitaries without providing a generalization to an arbitrary number of gates. In this sense, it needs further investigation whether there is a significant asymptotic scaling advantage for Hadamard promise problems or whether the methods we developed in this work also apply to these problems. All in all, this raises the important challenge of finding computational tasks for which the quantum-n-switch and indefinite causal structures in general provide a significant advantage.

Acknowledgements

We thank Mateus Araújo, Philippe Allard Guérin, Veronika Baumann, Ämin Baumeler, and an anonymous referee for insightful discussions and/or useful comments on the manuscript. We acknowledge financial support from the Austrian Science Fund (FWF) through BeyondC (F7103-N38), the project no. I-2906, as well as support by the John Templeton Foundation through grant 61466, The Quantum Information Structure of Spacetime (qiss.fr), the Foundational Questions Institute (FQXi) and the research platform TURIS. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. Furthermore, we are thankful for the very helpful tutorial on Q-circuit [183] that helped a lot to create the quantum circuits in LaTeX.

4.A. Representation of x in the algorithm for the FPP with the factoradic labeling

For the algorithm in Section 4.5 to work, it is necessary that every $x \in \{0, 1, ..., n! - 1\}$ can be represented in the basis given in Eq. (4.56). Here, we prove this statement:

Lemma 4.2. *For every* $x \in \{0, 1, ..., n! - 1\}$, *there exists* $(n-1) \cdot \lceil \log_2(n) \rceil$ *bits* $c^x_{k,i} \in \{0, 1\}$ *such that:*

$$x = \sum_{k=1}^{n-1} \sum_{i=1}^{\lceil \log_2(n) \rceil} c^x_{k,i} \cdot \left\lceil \frac{k}{2^i} \right\rceil \cdot k! \; . \tag{4.80}$$

Proof. First, we note that every $x \in \{0, ..., n! - 1\}$ can be represented in the factorial number system:

$$x = \sum_{k=1}^{n-1} a_k \cdot k! \, , \tag{4.81}$$

with coefficients $a_k \in \{0, 1, ..., k\}$. Hence, it is enough to show that for every $1 \leq k \leq n-1$, every $a_k \in \{0, ..., k\}$ can be written as:

$$a_k = \sum_{i=1}^{\lceil \log_2(n) \rceil} c_{k,i} \cdot \left\lceil \frac{k}{2^i} \right\rceil , \tag{4.82}$$

where $c_{k,i} \in \{0, 1\}$. For a clearer notation, we will drop the index "x" in $c^x_{k,i}$ from now on. Furthermore, let $1 \leq k \leq n - 1$ throughout the proof. In a first step, we prove by induction that the bits $c_{k,1}, c_{k,2}, ..., c_{k,\lceil \log_2(n) \rceil}$ can represent every number between 0 and $\sum_{i=1}^{\lceil \log_2(n) \rceil} \left\lceil \frac{k}{2^i} \right\rceil$ (according to Eq. (4.82)). In a second step, we show that $k \leq \sum_{i=1}^{\lceil \log_2(n) \rceil} \left\lceil \frac{k}{2^i} \right\rceil$, from which we conclude that indeed every $a_k \in \{0, ..., k\}$ can be written in the above form.

Step 1:
For every number $0 \leq a_k \leq \sum_{i=1}^{\lceil \log_2(n) \rceil} \left\lceil \frac{k}{2^i} \right\rceil$, there exist bits $c_{k,1}, c_{k,2}, ..., c_{k,\lceil \log_2(n) \rceil} \in \{0, 1\}$ such that

$$a_k = \sum_{i=1}^{\lceil \log_2(n) \rceil} c_{k,i} \cdot \left\lceil \frac{k}{2^i} \right\rceil . \tag{4.83}$$

Base case:
Every number $0 \leq a_k \leq \sum_{i=\lceil \log_2(n) \rceil}^{\lceil \log_2(n) \rceil} \left\lceil \frac{k}{2^i} \right\rceil = 1$ can be represented with $c_{k,1} = ... =$

$c_{k,\lceil\log_2(n)\rceil-1} = 0$ and $c_{k,\lceil\log_2(n)\rceil} = a_k$, since $\left\lceil \frac{k}{2^{\lceil\log_2(n)\rceil}} \right\rceil = 1$ whenever $1 \leq k \leq n-1$.

Induction step:
Suppose that $c_{k,1} = ... = c_{k,(j-1)} = 0$ and the bits $c_{k,j}, c_{k,(j+1)}, ..., c_{k,\lceil\log_2(n)\rceil} \in \{0,1\}$ are sufficient to represent every number a_k between 0 and $\sum_{i=j}^{\lceil\log_2(n)\rceil} \lceil \frac{k}{2^i} \rceil$ (according to Eq. (4.83)). If we now flip the bit $c_{k,(j-1)}$ to one, we can also represent every number a_k between $\lceil \frac{k}{2^{j-1}} \rceil$ and $\lceil \frac{k}{2^{j-1}} \rceil + \sum_{i=j}^{\lceil\log_2(n)\rceil} \lceil \frac{k}{2^i} \rceil = \sum_{i=j-1}^{\lceil\log_2(n)\rceil} \lceil \frac{k}{2^i} \rceil$, although $c_{k,1} = ... = c_{k,(j-2)} = 0$. To conclude that the bits $c_{k,(j-1)}, c_{k,j}, c_{k,(j+1)}, ..., c_{k,\lceil\log_2(n)\rceil} \in \{0,1\}$ (while all other bits are set to zero) are enough to represent every number between 0 and $\sum_{i=j-1}^{\lceil\log_2(n)\rceil} \lceil \frac{k}{2^i} \rceil$, we have to show that:

$$\left\lceil \frac{k}{2^{j-1}} \right\rceil \leq \sum_{i=j}^{\lceil\log_2(n)\rceil} \left\lceil \frac{k}{2^i} \right\rceil + 1 \,. \tag{4.84}$$

Since $\lceil x+y \rceil \leq \lceil x \rceil + \lceil y \rceil$ for all real numbers x and y, this can be inferred from repeating

$$\left\lceil \frac{k}{2^{j-1}} \right\rceil = \left\lceil \frac{k}{2^j} + \frac{k}{2^j} \right\rceil \leq \left\lceil \frac{k}{2^j} \right\rceil + \left\lceil \frac{k}{2^j} \right\rceil \tag{4.85}$$

for $j+1$, $j+2$, ..., $\lceil\log_2(n)\rceil$:

$$\left\lceil \frac{k}{2^{j-1}} \right\rceil \leq \left\lceil \frac{k}{2^j} \right\rceil + \left\lceil \frac{k}{2^j} \right\rceil \leq \left\lceil \frac{k}{2^j} \right\rceil + \left\lceil \frac{k}{2^{j+1}} \right\rceil + \left\lceil \frac{k}{2^{j+1}} \right\rceil \tag{4.86}$$

$$\leq ... \leq \sum_{i=j}^{\lceil\log_2(n)\rceil} \left\lceil \frac{k}{2^i} \right\rceil + \left\lceil \frac{k}{2^{\lceil\log_2(n)\rceil}} \right\rceil = \sum_{i=j}^{\lceil\log_2(n)\rceil} \left\lceil \frac{k}{2^i} \right\rceil + 1 \tag{4.87}$$

and using $\left\lceil \frac{k}{2^{\lceil\log_2(n)\rceil}} \right\rceil = 1$ (for $1 \leq k \leq n-1$) in the last step.

Hence, by induction, we have shown that the bits $c_{k,1}, c_{k,2}, ..., c_{k,\lceil\log_2(n)\rceil}$ can represent every number a_k between 0 and $\sum_{i=1}^{\lceil\log_2(n)\rceil} \lceil \frac{k}{2^i} \rceil$ (according to Eq. (4.83)).

Step 2:
To show that every $a_k \in \{0, ..., k\}$ can be represented with these bits, we have to prove that:

$$\sum_{i=1}^{\lceil\log_2(n)\rceil} \left\lceil \frac{k}{2^i} \right\rceil \geq k \,. \tag{4.88}$$

This is true since:

$$\sum_{i=1}^{\lceil\log_2(n)\rceil} \left\lceil \frac{k}{2^i} \right\rceil \geq \left\lceil \sum_{i=1}^{\lceil\log_2(n)\rceil} \frac{k}{2^i} \right\rceil = \left\lceil \left(1 - \frac{1}{2^{\lceil\log_2(n)\rceil}}\right) k \right\rceil \geq \left\lceil k - \frac{k}{n} \right\rceil \geq k \,, \tag{4.89}$$

where we have used $\sum_i \lceil x_i \rceil \geq \lceil \sum_i x_i \rceil$ in the first step, the geometric series in the second, $2^{\lceil \log_2(n) \rceil} \geq n$ in the third and $\frac{k}{n} < 1$ (for $1 \leq k \leq n-1$) in the last step. This completes the proof. □

If one is interested in obtaining the bits $c_{k,i}$ for a given a_k, one can use recursion: $c_{k,1} = 1$ if $a_k \geq \lceil \frac{k}{2} \rceil$ and 0 if $a_k < \lceil \frac{k}{2} \rceil$. The bit $c_{k,2}$ is 1 if $a_k - c_{k,1} \cdot \lceil \frac{k}{2} \rceil \geq \lceil \frac{k}{2^2} \rceil$ and 0 if $a_k - c_{k1} \cdot \lceil \frac{k}{2} \rceil < \lceil \frac{k}{2^2} \rceil$. Following this, the bit $c_{k,j}$ is 1 if $a_k - \sum_{i=1}^{j-1} c_{k,i} \cdot \lceil \frac{k}{2^i} \rceil \geq \lceil \frac{k}{2^j} \rceil$ and 0 if $a_k - \sum_{i=1}^{j-1} c_{k,i} \cdot \lceil \frac{k}{2^i} \rceil < \lceil \frac{k}{2^j} \rceil$. Note however that the representation of x (and the corresponding coefficients a_k of x) in this basis is not unique and this procedure is not the only one that leads to a representation of x in this basis.

4.B. Proof of Lemma 4.1

In this section, we will prove the remaining statement of Section 4.6. As a help to the reader, we restate the important definition for this lemma.

Definition 4.3. *For every permutation*

$$\Pi_x = (U_{\sigma_x(n-1)}...U_{\sigma_x((\hat{k}-1)\cdot\hat{n})})...\ (U_{\sigma_x(2\cdot\hat{n}-1)}...U_{\sigma_x(\hat{n})})\ (U_{\sigma_x(\hat{n}-1)}...U_{\sigma_x(0)})$$

of the n unitaries, let

$$\Pi_{xk} := \begin{cases} [U_{\sigma_x(n-1)}...U_{\sigma_x(\hat{n})}]\ (U_{\sigma_x(\hat{n}-1)}...U_{\sigma_x(0)}) & k=0 \\ [U_{\sigma_x(n-1)}...U_{\sigma_x((k+1)\cdot\hat{n})}]\ (U_{\sigma_x((k+1)\cdot\hat{n}-1)}...U_{\sigma_x(k\cdot\hat{n})})\ [U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}] & k=1,...,\hat{k}-2 \\ (U_{\sigma_x(n-1)}...U_{\sigma_x((\hat{k}-1)\cdot\hat{n})})\ [U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)}...U_{\sigma_x(0)}] & k=\hat{k}-1 \end{cases} \tag{4.90}$$

$$\tilde{\Pi}^r_{xk} := \begin{cases} \{U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}\}\ \{U_{\sigma_x(n-1)}...U_{\sigma_x(k\cdot\hat{n})}\} & k=1,...,\hat{k}-2 \\ \{U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)}...U_{\sigma_x(0)}\}\ \{U_{\sigma_x(n-1)}...U_{\sigma_x((\hat{k}-1)\cdot\hat{n})}\} & k=\hat{k}-1 \end{cases} \tag{4.91}$$

where $[U_{i_1}U_{i_2}...U_{i_j}]$ is defined to be the descending ordering of the unitaries $U_{i_1}U_{i_2}...U_{i_j}$, while $\{U_{i_1}U_{i_2}...U_{i_j}\}$ is the ascending ordering of them and $(U_{i_1}U_{i_2}...U_{i_j}) = U_{i_1}U_{i_2}...U_{i_j}$ leaves the string invariant. ($\hat{n} := \lceil \sqrt{n} \rceil$ and $\hat{k} := \lceil \frac{n}{\hat{n}} \rceil$)

Lemma 4.1. *For every set of (d-dimensional) unitaries $\{U_i\}_0^{n-1}$ that satisfy pairwise commutation relations, the following relation holds:*

$$\bigotimes_{k=0}^{\hat{k}-1} \Pi_{xk} |\Psi_k\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \tilde{\Pi}^r_{xk} |\Phi_k\rangle = \Pi_x |\Psi_0\rangle \otimes \bigotimes_{k=1}^{\hat{k}-1} \Pi |\Psi_k\rangle \otimes \Pi^r |\Phi_k\rangle\ . \tag{4.92}$$

Here, $\Pi := U_{n-1}...U_1U_0$ denotes the descending order of all unitaries, $\Pi^r := U_0U_1...U_{n-1}$ denotes the ascending order of all unitaries and $|\Psi_k\rangle, |\Phi_k\rangle \in \mathcal{H}^d$ are arbitrary d-dimensional states.

Proof. We will prove this by expressing both sides in terms of pairwise phases and comparing them at the end. Recall that the pairwise phase α_{jk} is defined via $U_j U_k = \alpha_{jk}\ U_k U_j$ and from comparing this with $U_k U_j = \alpha_{kj}\ U_j U_k$, we obtain $\alpha_{jk} = (\alpha_{kj})^{-1}$.

The total phase of the permutation Π_x is the product of the following pairwise phases α_{ij}:

$$\Pi_x = \prod_{\substack{n-1 \geq i > j \geq 0 \\ \text{with } \sigma_x(i) < \sigma_x(j)}} \alpha_{\sigma_x(i)\sigma_x(j)} \cdot \Pi\,. \tag{4.93}$$

To obtain this expression, we can compare every pair of unitaries and check whether they have the same order as in Π. If not, the corresponding phase appears in the product. To be more precise, we can start with $j = 0$ (with the matrix $U_{\sigma(0)}$) and check for every $i > j = 0$ if $\sigma_x(i) < \sigma_x(j = 0)$. If so, then the order of $U_{\sigma_x(0)}$ and $U_{\sigma_x(i)}$ is reversed between Π_x and Π and the corresponding phase appears as a factor in the total phase of Π_x with respect to Π. By repeating this procedure for every $n-1 \geq j \geq 0$, we obtain the above expression. For example, for $n = 4$ and the permutation $\Pi_x = U_1 U_2 U_0 U_3$ we obtain:

$$U_1 U_2 U_0 U_3 = \alpha_{03} \cdot \alpha_{23} \cdot \alpha_{13} \cdot \alpha_{12} \cdot U_3 U_2 U_1 U_0\,. \tag{4.94}$$

For the left hand side of the statement, we can compute for every Π_{xk} the relative phase to Π and for every $\tilde{\Pi}^r_{xk}$ the relative phase to Π^r in terms of the pairwise phases α_{ij}. One could calculate these phases in a direct way. Here we will follow this approach but with some shortcuts; in a first step, we rewrite for every $k \in \{0, 1, .., \hat{k} - 2\}$ the first two blocks of Π_{xk}, namely $[U_{\sigma_x(n-1)} ... U_{\sigma_x((k+1)\cdot\hat{n})}]\ (U_{\sigma_x((k+1)\cdot\hat{n}-1)} ... U_{\sigma_x(k\cdot\hat{n})})$, into:

$$\begin{aligned} &[U_{\sigma_x(n-1)} ... U_{\sigma_x((k+1)\cdot\hat{n})}]\ (U_{\sigma_x((k+1)\cdot\hat{n}-1)} ... U_{\sigma_x(k\cdot\hat{n})}) && (4.95) \\ &= \prod_{\substack{(k+1)\cdot\hat{n} > j \geq k\cdot\hat{n} \\ \text{and } n-1 \geq i > j, \\ \text{with } \sigma_x(i) < \sigma_x(j)}} \alpha_{\sigma_x(i)\sigma_x(j)} \cdot [U_{\sigma_x(n-1)} ... U_{\sigma_x(k\cdot\hat{n})}]\,. && (4.96) \end{aligned}$$

To see that this is true, remember that we obtain the phase as a product of the pairwise phases by comparing each pair of positions $n-1 \geq i > j \geq k\cdot\hat{n}$. But since the left block $[U_{\sigma_x(n-1)} ... U_{\sigma_x((k+1)\cdot\hat{n})}]$ is already ordered, we do not have to consider the cases for which $n-1 \geq i > j \geq (k+1)\cdot\hat{n}$. Similarly for $k = \hat{k} - 1$, we rewrite:

$$(U_{\sigma_x(n-1)} ... U_{\sigma_x((\hat{k}-1)\cdot\hat{n})}) = \prod_{\substack{n-1 \geq j \geq (\hat{k}-1)\cdot\hat{n} \\ \text{and } n-1 \geq i > j \\ \text{with } \sigma_x(i) < \sigma_x(j)}} \alpha_{\sigma_x(i)\sigma_x(j)} \cdot [U_{\sigma_x(n-1)} ... U_{\sigma_x((\hat{k}-1)\cdot\hat{n})}]\,. \tag{4.97}$$

To summarize, we can rewrite the permutations Π_{xk} into:

$$\Pi_{xk} = \alpha_{xk} \cdot \tilde{\Pi}_{xk}\,, \tag{4.98}$$

where:

$$\tilde{\Pi}_{xk} = \begin{cases} [U_{\sigma_x(n-1)}...U_{\sigma_x(0)}] & k=0 \\ [U_{\sigma_x(n-1)}...U_{\sigma_x(k\cdot\hat{n})}]\ [U_{\sigma_x(k\cdot\hat{n}-1)}...U_{\sigma_x(0)}] & k=1,...,\hat{k}-2 \\ [U_{\sigma_x(n-1)}...U_{\sigma_x((\hat{k}-1)\cdot\hat{n})}]\ [U_{\sigma_x((\hat{k}-1)\cdot\hat{n}-1)}...U_{\sigma_x(0)}] & k=\hat{k}-1 \end{cases} \tag{4.99}$$

and α_{xk} for $k \in \{0,1,...,\hat{k}-2\}$ becomes:

$$\alpha_{xk} = \prod_{\substack{(k+1)\cdot\hat{n}>j\geq k\cdot\hat{n} \\ \text{and } n-1\geq i>j, \\ \text{with } \sigma_x(i)<\sigma_x(j)}} \alpha_{\sigma_x(i)\sigma_x(j)}\,, \tag{4.100}$$

and for $k = \hat{k}-1$:

$$\alpha_{x(\hat{k}-1)} = \prod_{\substack{n-1\geq j\geq(\hat{k}-1)\cdot\hat{n} \\ \text{and } n-1\geq i>j \\ \text{with } \sigma_x(i)<\sigma_x(j)}} \alpha_{\sigma_x(i)\sigma_x(j)}\,. \tag{4.101}$$

The product of all the α_{xk} is exactly the same expression as in Eq. (4.93):

$$\prod_{k=0}^{\hat{k}-1} \alpha_{xk} = \prod_{\substack{n-1\geq j\geq 0 \\ \text{and } n-1\geq i>j \\ \text{with } \sigma_x(i)<\sigma_x(j)}} \alpha_{\sigma_x(i)\sigma_x(j)}\,. \tag{4.102}$$

In this way, we already obtain all required phases. For the above example of $\Pi_x = U_1U_2\ U_0U_3$, these expressions read:

$$\Pi_{x0} = U_2U_1\ U_0U_3 = \alpha_{03}\cdot\alpha_{13}\cdot\alpha_{23}\cdot U_3U_2\ U_1U_0 \qquad (\tilde{\Pi}_{x0} = U_3U_2\ U_1U_0) \tag{4.103}$$

$$\Pi_{x1} = U_1U_2\ U_3U_0 = \alpha_{12}\cdot U_2U_1\ U_3U_0 \qquad (\tilde{\Pi}_{x1} = U_2U_1\ U_3U_0) \tag{4.104}$$

$$(\tilde{\Pi}^r_{x1} = U_0U_3\ U_1U_2)\,. \tag{4.105}$$

Note that for $k=0$, the permutation $\tilde{\Pi}_{x0} = [U_{\sigma_x(n-1)}...U_{\sigma_x(0)}]$ always equals Π. On the other hand, the permutations $\tilde{\Pi}_{xk}$ (for $k \geq 1$) would lead in general to additional (unnecessary) phases relative to Π but they are exactly compensated by the phases of $\tilde{\Pi}^r_{xk}$ relative to Π^r. To see this, note that $\tilde{\Pi}_{xk}$ and $\tilde{\Pi}^r_{xk}$ are (by construction) the same permutations but in reversed order. This has the property that the relative phase between $\tilde{\Pi}_{xk}$ and Π is exactly the inverse of the relative phase between $\tilde{\Pi}^r_{xk}$ and Π^r. This is true since, whenever two unitaries U_i and U_j are commuted in $\tilde{\Pi}_{xk}$ relative to Π (and we obtain α_{ij} as a factor in the relative phase between $\tilde{\Pi}_{xk}$ and Π), then the two unitaries are also commuted in $\tilde{\Pi}^r_{xk}$ relative to Π^r (and we obtain the phase $\alpha_{ji} = (\alpha_{ij})^{-1}$ as a factor in the relative phase between $\tilde{\Pi}^r_{xk}$ and Π^r). For the example of $n=4$ and $\Pi_x = U_1U_2U_0U_3$

this becomes:

$$\begin{aligned}\tilde{\Pi}_{x1} &= U_2U_1\ U_3U_0 = \alpha_{13}\cdot\alpha_{23}\cdot U_3U_2\ U_1U_0\\ \tilde{\Pi}^r_{x1} &= U_0U_3\ U_1U_2 = \alpha_{31}\cdot\alpha_{32}\cdot U_0U_1\ U_2U_3 = (\alpha_{13})^{-1}\cdot(\alpha_{23})^{-1}\cdot U_0U_1\ U_2U_3\end{aligned}$$

$$\begin{aligned}\implies\ \tilde{\Pi}_{x1}\left|\Psi_1\right\rangle\otimes\tilde{\Pi}^r_{x1}\left|\Phi_1\right\rangle =& U_2U_1\ U_3U_0\left|\Psi_1\right\rangle\otimes U_0U_3\ U_1U_2\left|\Phi_1\right\rangle\\ =& U_3U_2\ U_1U_0\left|\Psi_1\right\rangle\otimes U_0U_1\ U_2U_3\left|\Phi_1\right\rangle = \Pi\left|\Psi_1\right\rangle\otimes\Pi^r\left|\Phi_1\right\rangle\ .\end{aligned} \tag{4.106}$$

In this way, we obtain for $k=0$ $\tilde{\Pi}_{x0}=\Pi$ and for every $k\in\{1,2,...,\hat{k}-1\}$ we obtain:

$$\tilde{\Pi}_{xk}\left|\Psi_k\right\rangle\otimes\tilde{\Pi}^r_{xk}\left|\Phi_k\right\rangle = \Pi\left|\Psi_k\right\rangle\otimes\Pi^r\left|\Phi_k\right\rangle\ . \tag{4.107}$$

Putting everything together, we obtain the desired equation:

$$\bigotimes_{k=0}^{\hat{k}-1}\Pi_{xk}\left|\Psi_k\right\rangle\otimes\bigotimes_{k=1}^{\hat{k}-1}\tilde{\Pi}^r_{xk}\left|\Phi_k\right\rangle \tag{4.108}$$

$$=\bigotimes_{k=0}^{\hat{k}-1}\alpha_{xk}\cdot\tilde{\Pi}_{xk}\left|\Psi_k\right\rangle\otimes\bigotimes_{k=1}^{\hat{k}-1}\tilde{\Pi}^r_{xk}\left|\Phi_k\right\rangle \tag{4.109}$$

$$=\left(\prod_{k=0}^{\hat{k}-1}\alpha_{xk}\right)\cdot\tilde{\Pi}_{x0}\left|\Psi_0\right\rangle\otimes\bigotimes_{k=1}^{\hat{k}-1}\tilde{\Pi}_{xk}\left|\Psi_k\right\rangle\otimes\tilde{\Pi}^r_{xk}\left|\Phi_k\right\rangle \tag{4.110}$$

$$=\left(\prod_{k=0}^{\hat{k}-1}\alpha_{xk}\right)\cdot\Pi\left|\Psi_0\right\rangle\otimes\bigotimes_{k=1}^{\hat{k}-1}\Pi\left|\Psi_k\right\rangle\otimes\Pi^r\left|\Phi_k\right\rangle \tag{4.111}$$

$$=\left(\prod_{\substack{n-1\geq j\geq 0\\ \text{and } n-1\geq i>j\\ \text{with } \sigma_x(i)<\sigma_x(j)}}\alpha_{\sigma_x(i)\sigma_x(j)}\right)\cdot\Pi\left|\Psi_0\right\rangle\otimes\bigotimes_{k=1}^{\hat{k}-1}\Pi\left|\Psi_k\right\rangle\otimes\Pi^r\left|\Phi_k\right\rangle \tag{4.112}$$

$$=\Pi_x\left|\Psi_0\right\rangle\otimes\bigotimes_{k=1}^{\hat{k}-1}\Pi\left|\Psi_k\right\rangle\otimes\Pi^r\left|\Phi_k\right\rangle\ , \tag{4.113}$$

where we used Eq. (4.98) in the first step, Eq. (4.107) in the third step, Eq. (4.102) in the fourth step and Eq. (4.93) in the last one. This completes the proof. □

4.C. $O(n \log n)$-algorithm for $n = 8$

In Figure 4.14, we present the algorithm that solves the FPP with the factoradic labeling of the permutations for $n = 8$. It consists of 14 d-dimensional target systems and 21 control qubits. The gate U_0 appears once on every target system and the gates $U_1, ..., U_7$ appear each six times. Hence, in total 56 black-box unitaries are used. Note that the shortest known string containing all 8! permutations of the eight unitaries $U_0, ..., U_7$ has length 51 [177]. Nevertheless, if we omit the target systems $|\Psi_{8,1}\rangle$, $|\Psi_{8,2}\rangle$, $|\Psi_{8,3}\rangle$ and $|\Psi_{8,8}\rangle$ from our circuit together with the involved black-box unitaries and the corresponding control qubits $|c_{1,3}\rangle$, $|c_{2,3}\rangle$ and $|c_{3,3}\rangle$ (since they are unnecessary to represent every number $x \in \{0, ..., 8! - 1\}$), we save 10 queries and solve the problem by calling only 46 black-box unitaries.

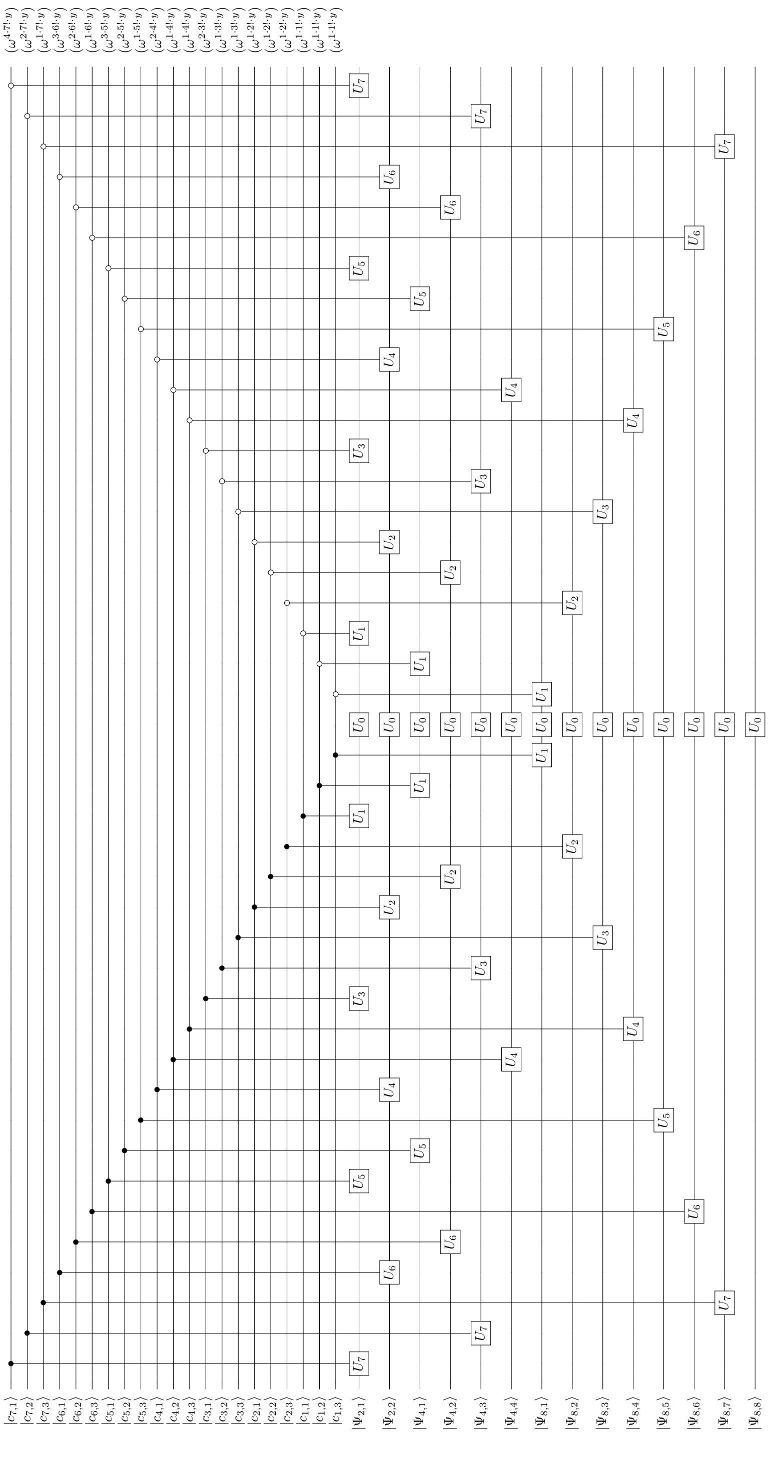


Figure 4.14.: The algorithm of Subsection 4.5.2 (Fig. 4.8) for $n = 8$.

5. Computational advantage from a quantum superposition of qubit gate orders

This chapter is based on the article:

Martin J. Renner, Časlav Brukner "Computational Advantage from a Quantum Superposition of Qubit Gate Orders," Physical Review Letters 128 (23), 230503 (2022)
(Received: 4 January 2022, Accepted: 31 March 2022, Published: 10 June 2022)

Contributions: Časlav supervised the work and introduced the problem to me. The results were found and the manuscript was written by myself. All technical proofs were carried out by myself.

Abstract

In an ordinary quantum algorithm the gates are applied in a fixed order on the systems. The introduction of indefinite causal structures allows to relax this constraint and control the order of the gates with an additional quantum state. It is known that this quantum-controlled ordering of gates can reduce the query complexity in deciding a property of black-box unitaries with respect to the best algorithm in which the gates are applied in a fixed order. However, all tasks explicitly found so far require unitaries that either act on unbounded dimensional quantum systems in the asymptotic limit (the limiting case of a large number of black-box gates) or act on qubits, but then involve only a few unitaries. Here we introduce tasks (1) for which there is a provable computational advantage of a quantum-controlled ordering of gates in the asymptotic case and (2) that require only qubit gates and are therefore suitable to demonstrate this advantage experimentally. We study their solutions with the quantum-n-switch and within the quantum circuit model and find that while the n-switch requires to call each gate only once, a causal algorithm has to call at least $2n-1$ gates. Furthermore, the best known solution with a fixed gate ordering calls $O(n\log_2(n))$ gates.

5.1. Introduction

Causality is one of the most fundamental concepts in science and deeply embedded in the concept of computation. In ordinary quantum algorithms, represented within the quantum circuit model, the gates act in a fixed order on the systems. However, the study of causality at the intersection between quantum mechanics and gravity within the last two decades [21, 155] suggested that quantum computation can be extended to more general scenarios, in which the order of the gates is controlled with an additional quantum state [23, 22]. The use of indefinite causal structures provide numerous advantages in the field of quantum information. For instance, they lead to an exponential reduction for certain communication tasks [156] and offer advantages in channel discrimination tasks [157]. Moreover, they allow to transfer information through zero-capacity channels [158, 159, 160, 161, 162], although the same effect appears in causal circuits [163, 164, 165]. Beside the theoretical interest of indefinite causal structures, including the study of the computational complexity [166, 167], they were experimentally demonstrated in enhanced tabletop experiments [168, 169, 170, 171, 172, 28].

The simplest example of an indefinite causal structure is the quantum-n-switch. Here, any permutation of the n unitaries can be applied on the target system but the order in which these unitaries are applied depends on the state of an additional quantum system. For example, in the case of the quantum-2-switch, a qubit controls whether the gate U_0 is applied before or after another gate U_1. It is known that using these structures one can decide whether the two gates U_0 and U_1 commute or anticommute with a single call to each gate. Solving the same task within the standard quantum circuit model, however, requires to call at least one gate twice [26]. This effect has also been experimentally demonstrated by Procopio et al. [168]. In this way, the use of indefinite causal structures allows for an advantage in the number of gates that has to be called (queries).

A generalization of this task to n unitary gates, originally introduced in Araújo et al. [27] and often called Fourier promise problems (FPP), can be solved with the quantum-n-switch and a single call to each gate. At the same time, the best known solution with a causal algorithm calls $O(n \log_2(n))$ gates [184]. This result suggests that a quantum computer with a quantum-controlled ordering of gates require asymptotically fewer resources than a quantum computer with a fixed gate ordering to solve the same task. Unfortunately, the physical conditions to achieve this advantage are very demanding: for the tasks with n unitaries the dimension of the control and target systems must be at least $n!$. This makes it virtually impossible to demonstrate this computational advantage experimentally. For this reason, another generalization of the task to more unitary gates has been proposed and experimentally demonstrated (for $n = 4$) by Taddei et al. [28]. These problems, called Hadamard promise problems (HPP), offer an advantage by using the quantum-n-switch compared to causal circuits as well, but most importantly require only qubits. However, so far only one task of this class with four gates is explicitly known, and it remained open whether this advantage is preserved in the limiting case of a large number of black-box gates.

Here we generalize these tasks to an arbitrary number of unitary gates and show that they (1) provide a provable gap in query complexity between a quantum-controlled ordering of gates and causal quantum circuits in the asymptotic case, and (2) require only qubit gates. In fact, while all of these tasks can be solved with the quantum-n-switch and a single call to each gate, we prove that a causal algorithm requires at least $2n - 1$

calls to the gates. Furthermore, we show that the best known techniques with a fixed gate ordering require $O(n \log_2(n))$ queries and conjecture that no better causal solution exists. Our findings allow to verify experimentally the scalable computational advantage of indefinite causal structures.

5.2. The Hadamard Promise Problem

In the Hadamard promise problem, originally introduced in Ref. [28], a set of d-dimensional unitary gates $\{U_i\}_0^{n-1}$ is given and certain permutations of these unitaries are chosen. These permutations are denoted by Π_x where the index x ranges from 0 to $n_x - 1$ and $n_x \leq n!$ is the number of selected permutations. It is promised that for some $y \in \{0, 1, ..., n_x - 1\}$ the following relations hold:

$$\forall x \in \{0, 1, ..., n_x - 1\} : \; \Pi_x = s(x, y) \cdot \Pi_0 \,. \tag{5.1}$$

Here, the coefficients $s(x, y)$ form a $n_x \times n_x$ Hadamard matrix, an orthogonal matrix whose entries are either $+1$ or -1. More formally, $s(x, y) \in \{+1, -1\}$ and the rows are pairwise orthogonal to each other:[1]

$$\forall y, y' \in \{0, 1, ..., n_x - 1\} : \; \sum_{x=0}^{n_x - 1} s(x, y) \cdot s(x, y') = n_x \cdot \delta_{y,y'} \,. \tag{5.2}$$

The task is to find the value y for which these promises are satisfied.

The simplest HPP involves two black-box unitaries U_0 and U_1. For the two permutations $\Pi_0 = U_1 U_0$ and $\Pi_1 = U_0 U_1$ it is promised that $\Pi_x = s(x, y)\ \Pi_0$ where $s(x, y) = (-1)^{x \cdot y}$. While the promise for $x = 0$ becomes $\Pi_0 = \Pi_0$, which is trivially satisfied, for $x = 1$ it translates into:

$$U_0 U_1 = (-1)^y \cdot U_1 U_0 \,. \tag{5.3}$$

Hence, the two gates either commute $(y = 0)$ or anticommute $(y = 1)$ and the task is to find out which property is the correct one. As already mentioned in the introduction, it is known that this task can be solved with the quantum-2-switch by calling each gate only once, while in any causal quantum algorithm at least one gate has to be called twice [26].

5.3. Generalizing HPPs

For higher n only a few explicit HPPs are known. In this work, we will introduce a procedure that allows us to find a HPP for any number of involved black-box gates. The main idea is that we can combine two HPPs each with m and n (d-dimensional) unitary gates into another HPP with $m + n - 1$ (d-dimensional) unitary gates. To do so, we

[1] To avoid confusion, we want to mention that we label the columns with x and the rows with y.

x	$x=0$	$x=1$	Examples	
y	$(\Pi_0 = \Pi_0)$	$(\Pi_1 = (-1)^y\ \Pi_0)$	U_0	U_1
$y=0$	1	1	σ_x	σ_x
$y=1$	1	-1	σ_y	σ_x

Table 5.1.: The Hadamard matrix for the simplest HPP in which two unitaries either commute ($y=0$) or anticommute ($y=1$). The task is to find the correct value of y.

denote the m_x permutations of the m unitaries in the first HPP with $\Pi^{(1)}_{x_1}$ such that they satisfy the following promises:

$$\forall x_1 \in \{0,1,...,m_x-1\}:\ \Pi^{(1)}_{x_1} = s_1(x_1,y_1)\cdot\Pi^{(1)}_0\,. \tag{5.4}$$

In the second HPP there are n involved d-dimensional black-box unitaries and the n_x permutations, denoted as $\Pi^{(2)}_{x_2}$, satisfy the following promises:

$$\forall x_2 \in \{0,1,...,n_x-1\}:\ \Pi^{(2)}_{x_2} = s_2(x_2,y_2)\cdot\Pi^{(2)}_0\,. \tag{5.5}$$

Now we choose one of the m unitaries from the first HPP and replace this unitary in each of the permutations $\Pi^{(1)}_{x_1}$ with $\Pi^{(2)}_{x_2}$. In this way, we obtain $n_x \cdot m_x$ new permutations that we label with $\Pi_{(x_1,x_2)}$. One can observe that these new permutations satisfy the following relations:

$$\Pi_{(x_1,x_2)} = s_2(x_2,y_2)\cdot\Pi_{(x_1,0)} = s_2(x_2,y_2)\cdot s_1(x_1,y_1)\cdot\Pi_{(0,0)}\,. \tag{5.6}$$

Since $s_1(x_1,y_1)$ and $s_2(x_2,y_2)$ form an $m_x \times m_x$ and $n_x \times n_x$ Hadamard matrix, respectively, the resulting matrix with entries $s((x_1,x_2),(y_1,y_2)) := s_2(x_2,y_2)\cdot s_1(x_1,y_1)$ is a $(m_x \cdot n_x)\times(m_x\cdot n_x)$ Hadamard matrix. We prove this formally in Appendix 5.A. Hence, we have obtained another HPP with $m+n-1$ involved (d-dimensional) unitary black-box gates.

To give an example, we can consider the simplest HPP in Table 5.1 with two involved unitaries. Let U_0 and $\tilde{U}_1$ be the unitaries for which it is promised that they either commute ($y_1=0$) or anticommute ($y_1=1$). The permutations $\Pi^{(1)}_{x_1}$ read then:

$$\Pi^{(1)}_{x_1=0} = \tilde{U}_1 U_0 \tag{5.7}$$

$$\Pi^{(1)}_{x_1=1} = U_0\tilde{U}_1 = (-1)^{y_1}\cdot\tilde{U}_1 U_0\,. \tag{5.8}$$

Now we can take another instance of the same HPP with $\Pi^{(2)}_{x_2=0} = U_2U_1$ and $\Pi^{(2)}_{x_2=1} = U_1U_2$ such that the two unitaries U_1 and U_2 again either commute ($y_2=0$) or anticommute ($y_2=1$):

$$U_1U_2 = (-1)^{y_2}\ U_2U_1\,. \tag{5.9}$$

Replacing now $\tilde{U}_1$ in both of the permutations $\Pi^{(1)}_{x_1=0} = \tilde{U}_1 U_0$ and $\Pi^{(1)}_{x_1=1} = U_0 \tilde{U}_1$ once with $\Pi^{(2)}_{x_2=0} = U_2 U_1$ and once with $\Pi^{(2)}_{x_2=1} = U_1 U_2$, we obtain in total four permutations for which the following promises hold:

$$\Pi_{(0,0)} = U_2 U_1 U_0 \,, \tag{5.10}$$

$$\Pi_{(0,1)} = U_1 U_2 U_0 = (-1)^{y_2} \; U_2 U_1 U_0 \,, \tag{5.11}$$

$$\Pi_{(1,0)} = U_0 U_2 U_1 = (-1)^{y_1} \; U_2 U_1 U_0 \,, \tag{5.12}$$

$$\Pi_{(1,1)} = U_0 U_1 U_2 = (-1)^{y_1+y_2} \; U_2 U_1 U_0 \,. \tag{5.13}$$

We illustrate in Table 5.2 that these relations form indeed a 4 × 4 Hadamard matrix.

(x_1, x_2)	$x =$	$x =$	$x =$	$x =$	Examples		
(y_1, y_2)	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$	U_0	U_1	U_2
$y = (0,0)$	1	1	1	1	σ_x	σ_x	$\mathbb{1}$
$y = (0,1)$	1	1	-1	-1	σ_x	$\frac{\sigma_y+\sigma_z}{\sqrt{2}}$	$\frac{\sigma_y-\sigma_z}{\sqrt{2}}$
$y = (1,0)$	1	-1	1	-1	σ_y	σ_x	$\mathbb{1}$
$y = (1,1)$	1	-1	-1	1	σ_y	$\frac{\sigma_y+\sigma_z}{\sqrt{2}}$	$\frac{\sigma_y-\sigma_z}{\sqrt{2}}$

Table 5.2.: The Hadamard matrix for the HPP given in (5.10)-(5.13) (for short: $\Pi_{(x_1,x_2)} = (-1)^{x_1 \cdot y_1 + x_2 \cdot y_2} \; \Pi_{(0,0)}$). For every possible combination of the parameters $y = (y_1, y_2)$ a set of unitaries that satisfy the promise is given.

To obtain a HPP with four gates, one could use the method again and replace, for instance, U_0 with another pair of commuting or anticommuting unitaries U_3 and U_4 (hence $U_4 U_3 = (-1)^{y_3} \; U_3 U_4$). In this way, we have to replace U_0 in each of the four permutations $\Pi_{(x_1,x_2)}$ once with $U_4 U_3$ ($x_3 = 0$) and once with $U_3 U_4$ ($x_3 = 1$) which leads to eight permutations $\Pi_{(x_1,x_2,x_3)}$ in total. Following this, we obtain a HPP for every number of unitary black-box gates n with $n_x = 2^{n-1}$ permutations. Note, however, that we are not restricted to split a unitary into a pair of commuting or anticommuting unitaries, but replacing a unitary by any set of permutations that form a HPP by themselves is possible.

To show that these tasks are indeed realisable, one has to prove that unitaries that satisfy these promises exist. It turns out that for many tasks of this class, this can be done by a straightforward approach. For instance, we obtained the examples in Table 5.2 by simply replacing the examples of $U_1 = \sigma_x$ in Table 5.1 with a pair of unitaries that either commute (if $y_2 = 0$) or anticommute (if $y_2 = 1$) and whose product is proportional to the original unitary $U_1 = \sigma_x$:

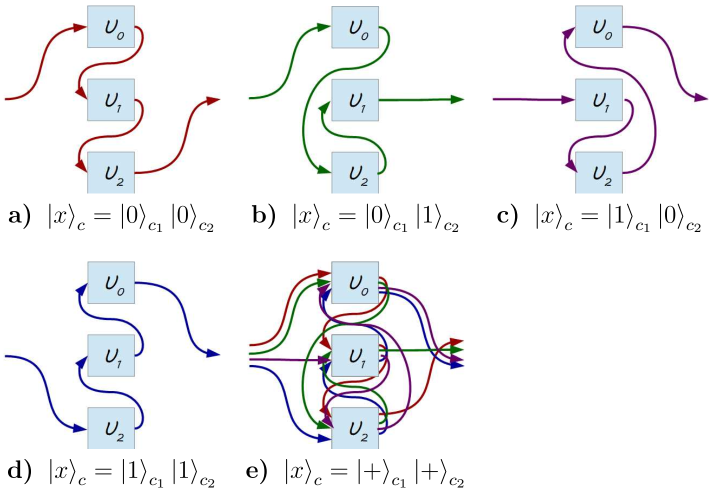
$$U_1 = \sigma_x \xrightarrow{y_2=0} U_1 = \sigma_x \qquad\qquad U_2 = \mathbb{1} \tag{5.14}$$

$$U_1 = \sigma_x \xrightarrow{y_2=1} U_1 = \frac{\sigma_y + \sigma_z}{\sqrt{2}} \qquad\qquad U_2 = \frac{\sigma_y - \sigma_z}{\sqrt{2}} \tag{5.15}$$

In this sense, we obtain the examples for the task with $n + 1$ unitaries from the examples for the task with n unitaries. Since there are some subtleties with this procedure, we discuss this further in Appendix 5.B.

Albeit all statements in this work hold for an arbitrary target space dimension d, the existence of qubit gates ($d = 2$) for any task in this class is important for experiments. This is different for Fourier promise problems [27], where the target space dimension has to grow with the number of gates. For the interested reader, we give a more detailed comparison between these two classes in Appendix 5.C. However, in the contrary to (discrete) Fourier matrices, Hadamard matrices can only exist in dimension 1, 2 and multiples of 4 [185]. Since the size of the Hadamard matrix always equals the number of involved permutations, this implies for example that there is no HPP that uses all six permutations in the case of $n = 3$. At the same time, a smaller number of involved permutations is also an advantage for future experiments, since it requires less control about the degrees of freedom that implement the different gate orderings.

5.4. Solution with the quantum-n-switch



a) $|x\rangle_c = |0\rangle_{c_1} |0\rangle_{c_2}$ **b)** $|x\rangle_c = |0\rangle_{c_1} |1\rangle_{c_2}$ **c)** $|x\rangle_c = |1\rangle_{c_1} |0\rangle_{c_2}$

d) $|x\rangle_c = |1\rangle_{c_1} |1\rangle_{c_2}$ **e)** $|x\rangle_c = |+\rangle_{c_1} |+\rangle_{c_2}$

$$\frac{1}{2}\sum_{x_i=0}^{1} |x_1\rangle_{c_1} |x_2\rangle_{c_2} |\Psi_t\rangle \xrightarrow{S_3} \frac{1}{2}\sum_{x_i=0}^{1} |x_1\rangle_{c_1} |x_2\rangle_{c_2} \Pi_{(x_1,x_2)} |\Psi_t\rangle$$
$$= H\,|y_1\rangle_{c_1} H\,|y_2\rangle_{c_2} \Pi_{(0,0)} |\Psi_t\rangle$$

Figure 5.1.: Solving the HPP in Table 5.2 with the 3-switch: The state of the control system $|x\rangle_c = |x_1\rangle_{c_1} |x_2\rangle_{c_2}$ determines in which order the gates are applied on the target system. If the control system is initialized in a superposition, the quantum-3-switch can be used to solve this HPP by calling each unitary U_i only once.

As pointed out in Ref. [28], every HPP (independent of whether it is constructed using our method or otherwise) can be solved with the quantum-n-switch and a single call to each gate. The quantum-n-switch is denoted here as S_n. It is the quantum gate that applies the permutation Π_x on the target system $|\Psi_t\rangle$ whenever the control system is in the state $|x\rangle$:

$$\forall x \in \{0, 1, ..., n_x - 1\} : \; S_n \left|x\right\rangle_c \otimes \left|\Psi_t\right\rangle = \left|x\right\rangle_c \otimes \Pi_x \left|\Psi_t\right\rangle \; . \tag{5.16}$$

Moreover, to every Hadamard matrix $s(x, y)$ we associate the corresponding unitary transformation H_{n_x} that is defined as:

$$\forall y \in \{0, 1, ..., n_x - 1\} : \; H_{n_x} \left|y\right\rangle = \frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} s(x, y) \left|x\right\rangle \; . \tag{5.17}$$

To solve HPPs, the n_x-dimensional control system is first transformed into an equal superposition of all states $x \in \{0, 1, ..., n_x - 1\}$, usually by applying a Hadamard transformation to all control qubits. Meanwhile, the target system $|\Psi_t\rangle$ is initialized in an arbitrary d-dimensional state:

$$\left(\frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} \left|x\right\rangle_c \right) \otimes \left|\Psi_t\right\rangle \; . \tag{5.18}$$

Now, if the n-switch is applied, depending on the state $|x\rangle$ of the control system, the permutation Π_x is applied on the target system $|\Psi_t\rangle$ (see Fig. 5.1 for an illustration of the map for the case of $n = 3$):

$$S_n \left(\frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} \left|x\right\rangle_c \right) \otimes \left|\Psi_t\right\rangle = \frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} \left|x\right\rangle_c \otimes \Pi_x \left|\Psi_t\right\rangle \; . \tag{5.19}$$

With the promise $\Pi_x = s(x, y) \cdot \Pi_0$, this state can be rewritten into:

$$\frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} \left|x\right\rangle_c \otimes \Pi_x \left|\Psi_t\right\rangle = \left(\frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} s(x, y) \left|x\right\rangle_c \right) \otimes \Pi_0 \left|\Psi_t\right\rangle \; . \tag{5.20}$$

In this way, the target system always ends up in the state $\Pi_0 |\Psi_t\rangle$ (independent of x) and factorizes out. Observe that the final state of the control system is precisely $H_{n_x} |y\rangle_c$. Hence, applying the inverse (transposed) Hadamard transform $H_{n_x}^{-1}$ on the control system, we obtain:

$$H_{n_x}^{-1} \left(\frac{1}{\sqrt{n_x}} \sum_{x=0}^{n_x - 1} s(x, y) \left|x\right\rangle_c \right) \otimes \Pi_0 \left|\Psi_t\right\rangle = \left|y\right\rangle_c \otimes \Pi_0 \left|\Psi_t\right\rangle \; . \tag{5.21}$$

In this way, the solution y can be read out by a measurement of the control system in the computational basis. In the n-switch each unitary is called exactly once. Hence, the total query complexity of this algorithm is n.

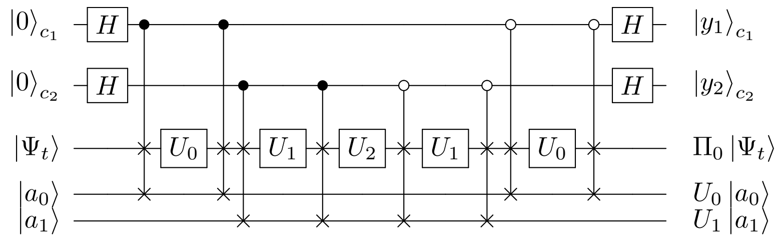


Figure 5.2.: Simulation of the four permutations $U_2U_1U_0$, $U_1U_2U_0$, $U_0U_2U_1$, $U_0U_1U_2$ involved in the HPP given in Table 5.2 with the smallest possible number of used black-box gates. A measurement of the control qubits at the end reveals the solution $y = (y_1, y_2)$.

5.5. Solution with causal quantum algorithms

It is possible to simulate the quantum-n-switch with a causal algorithm and $O(n^2)$ calls to the black-box gates. Since every HPP can be solved with the quantum-n-switch, every simulation thereof (or more precisely the simulation of all involved permutations) can solve the same task as well. For a detailed study of the simulation of the quantum-n-switch we refer to Ref. [174] (but also Ref. [173, 27, 184]). For example, all permutations involved in the HPP given in Table 5.2 can be simulated with the algorithm in Fig. 5.2. This is also the shortest possible solution since such an algorithm can be used to determine for each pair of the unitaries U_0, U_1 and U_2 whether the pair of unitaries commute or anticommute (by setting the remaining gate to $\mathbb{1}$). Such a causal algorithm requires to call at least two of the three unitaries twice, hence at least five gates are called in total.

The idea can be extended to HPPs with a set of n unitary gates. Each such problem contains as a subproblem the task of deciding for each pair of gates whether that pair commutes or anticommutes. This later problem requires a minimum number of queries and thus also determines a lower bound on the number of queries for the original problem. This is specified by the following lemma.

Lemma 5.1. *Consider the class of all problems that can be generated from the HPP in Table 5.1 with the method introduced in Section 5.3. For every HPP (with a set of n different gates) in that class a solution with a causal quantum algorithm has to call at least $2n-1$ unitary gates.*

Proof. Appendix 5.D.1 □

However, we believe that for most tasks in that class a causal solution has to call more than $O(n)$ gates. To motivate our conjecture, we want to point out that a similar argument as above holds for a very simple HPP that contains only two permutations and is defined by:

$$\Pi_0 := U_{n-1}U_{n-2}...U_2U_1U_0\,, \tag{5.22}$$

$$\Pi_1 := U_0U_1U_2...U_{n-2}U_{n-1}\,. \tag{5.23}$$

It is promised that $\Pi_1 = (-1)^y \, \Pi_0$ and the task is to determine y. A solution to that HPP is able to determine for every pair of unitaries U_j and U_k whether they commute or anticommute. More precisely, if we set all remaining unitaries to $\mathbb{1}$, the two permutations reduce to $\Pi_0 = U_k U_j$ and $\Pi_1 = U_j U_k$ (given that w.l.o.g. $j < k$) from which the statement follows. In general, however, a HPP with n gates that is generated with our method contains many more permutations (in fact 2^{n-1}) and is able to determine much more structure between the unitaries.

Therefore, we conjecture that, for small n, a simulation of all involved permutations is the most efficient causal solution. For larger n, methods similar to the ones introduced in Ref. [184] can be used to find more efficient solutions. Indeed, we show in Appendix 5.D.2 that all HPPs that we can generate with our method can be solved with a causal quantum algorithm and $O(n \log_2(n))$ calls to the black-box gates and we conjecture that there are tasks in this class for which no better causal solution exists.[2]

5.6. Conclusion

Indefinite causal structures can be used to solve certain tasks more efficiently than any causally ordered quantum algorithm. In this work, we generalized a specific class of problems that provide an advantage of using a superposition of different gate orderings in the asymptotic limit. These tasks are constructed for an arbitrary number of gates and are suitable for an experimental demonstration of this computational advantage since only low-dimensional target systems (qubits) are required. We showed that, while all of these tasks can be solved with the quantum-n-switch and a single call to each gate, causal algorithms require more calls to the black-box unitaries. Possible platforms for the demonstration of this advantage are tabletop experiments involving photon polarization [168, 165, 169, 170, 28, 161], orbital angular momentum [171, 162] or time-bin degrees of freedom [172] for the target system, and either path [168, 165, 169, 170, 28, 161, 172] or polarization [171, 162] for the control system. The scalability to more gates has been already demonstrated in a photonic platform by Taddei et al. [28] but also other experiments using for example trapped ions are possible [186].

Furthermore, we found that all of these tasks can be solved with a causal algorithm and $O(n \log_2(n))$ calls to the black-box gates. We want to point out that currently there is no known task for which the advantage in the number of gates that has to be called is larger than $O(n)$ (for indefinite causal structures) versus $O(n \log_2(n))$ (for causal quantum circuits). This raises the important challenge of finding computational tasks for which indefinite causal structures provide a more significant advantage.

[2] There are other known techniques to solve the same tasks. They are discussed in Ref. [28] and it is argued there that they require more calls to the black-box gates than a simulation of all permutations.

Acknowledgements

We thank two anonymous referees for many valuable comments and suggestions. We acknowledge financial support from the Austrian Science Fund (FWF) through BeyondC (F7103-N38), the project no. I-2906, as well as support by the John Templeton Foundation through grant 61466, The Quantum Information Structure of Spacetime (qiss.fr), the Foundational Questions Institute (FQXi) and the research platform TURIS. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation. Furthermore, we are thankful for the tutorial on Q-circuit [183] that helped a lot to create the quantum circuits in LaTeX.

5.A. The product of two Hadamard matrices is another Hadamard matrix

Lemma 5.2. *If $s_1(x_1, y_1)$ and $s_2(x_2, y_2)$ are the entries of an $m_x \times m_x$ and $n_x \times n_x$ Hadamard matrix, then $s((x_1, x_2), (y_1, y_2)) := s_2(x_2, y_2) \cdot s_1(x_1, y_1)$ forms an $(m_x \cdot n_x) \times (m_x \cdot n_x)$ Hadamard matrix.*

Proof. Since $s_1(x_1, y_1)$ and $s_2(x_2, y_2)$ form Hadamard matrices, we know that $s_1(x_1, y_1) \in \{+1, -1\}$ and $s_2(x_2, y_2) \in \{+1, -1\}$ from which we conclude that $s((x_1, x_2), (y_1, y_2)) \in \{+1, -1\}$. Furthermore, since $s_1(x_1, y_1)$ and $s_2(x_2, y_2)$ form orthogonal matrices, we know:

$$\forall y_1, y_1' \in \{0, 1, ..., m_x - 1\} : \sum_{x_1=0}^{m_x-1} s_1(x_1, y_1) \cdot s_1(x_1, y_1') = m_x \cdot \delta_{y_1, y_1'} \,. \tag{5.24}$$

For $s_2(x_2, y_2)$ the analog expression holds. From this we can calculate directly that $s((x_1, x_2), (y_1, y_2))$ forms an orthogonal matrix as well. In fact, two rows are orthogonal to each other:

$$\begin{aligned}
&\forall (y_1, y_2), (y_1', y_2') \in \{0, 1, ..., m_x - 1\} \times \{0, 1, ..., n_x - 1\} : \\
&\sum_{x_1=0}^{m_x-1} \sum_{x_2=0}^{n_x-1} s((x_1, x_2), (y_1, y_2)) \cdot s((x_1, x_2), (y_1', y_2')) \\
&\qquad = \sum_{x_1=0}^{m_x-1} \sum_{x_2=0}^{n_x-1} s_1(x_1, y_1) \cdot s_2(x_2, y_2) \cdot s_1(x_1, y_1') \cdot s_2(x_2, y_2') \\
&\qquad = \left(\sum_{x_1=0}^{m_x-1} s_1(x_1, y_1) \cdot s_1(x_1, y_1') \right) \cdot \left(\sum_{x_2=0}^{n_x-1} s_2(x_2, y_2) \cdot s_2(x_2, y_2') \right) \\
&\qquad = m_x \cdot n_x \cdot \delta_{y_1, y_1'} \cdot \delta_{y_2, y_2'} \\
&\qquad = (m_x \cdot n_x) \cdot \delta_{(y_1, y_2), (y_1', y_2')} \,.
\end{aligned} \tag{5.25}$$

Hence $s((x_1, x_2), (y_1, y_2))$ forms an orthogonal matrix whose entries are either $+1$ or -1, a Hadamard matrix. □

5.B. Existence of unitaries that satisfy the promise

As already mentioned in the main text, given that examples of unitaries for the task with n gates exist, examples for the task with $n + 1$ unitaries (in which one unitary is replaced by a pair of either commuting or anticommuting unitaries) can be found. More formally, if the unitary U_i in the original HPP is of the form $U \sigma_z U^\dagger$ (for an arbitrary 2-dimensional unitary U) and should be replaced by a pair of commuting unitaries, one can choose for instance $U \sigma_z U^\dagger$ and $\mathbb{1}$:

$$U \sigma_z U^\dagger = \mathbb{1} \cdot U \sigma_z U^\dagger = U \sigma_z U^\dagger \cdot \mathbb{1} \,. \tag{5.26}$$

On the other hand, if U_i should be replaced with two anticommuting unitaries, one can choose for example $U\sigma_x U^\dagger$ and $U(i\sigma_y)U^\dagger$ since:

$$U\sigma_z U^\dagger = U\sigma_x U^\dagger \cdot U(i\sigma_y)U^\dagger = -U(i\sigma_y)U^\dagger \cdot U\sigma_x U^\dagger \,. \tag{5.27}$$

Note that such a replacement is not unique, since we can choose $U\left(\frac{\sigma_x+\sigma_y}{\sqrt{2}}\right)U^\dagger$ and $U\left(\frac{\sigma_x-\sigma_y}{\sqrt{2}}\right)U^\dagger$ as well. More precisely, replacing $U\sigma_z U^\dagger$ by $UV\sigma_x V^\dagger U^\dagger$ and $UV(i\sigma_y)V^\dagger U^\dagger$ with $V\sigma_z V^\dagger = \sigma_z$ is allowed. (Intuitively speaking, V is a rotation in the x-y-plane that leaves the z direction invariant.)

Nevertheless, for certain parameter combinations a problem appears with this method. Take for instance the step in which $U\sigma_z U^\dagger$ is replaced by $U\sigma_z U^\dagger$ and $\mathbb{1}$. In a next step, it is impossible to replace $\mathbb{1}$ by a pair of anticommuting unitaries since there are no 2×2 unitaries U_1 and U_2 such that:

$$\mathbb{1} = U_1 \cdot U_2 = -U_2 \cdot U_1 \,. \tag{5.28}$$

(However, replacing $\mathbb{1}$ by two commuting unitaries is clearly possible, for example $U\sigma_z U^\dagger$ and $U\sigma_z U^\dagger$.) Therefore, if we are only using combinations of the most simplest HPP in Table 5.1 (replacing a unitary step by step with pairs of commuting or anticommuting unitaries), for certain parameter combinations no examples of 2-dimensional unitaries can be found. For the task itself this only implies that certain solutions y are impossible. For the other parameter combinations, unitaries that satisfy the promises can still be found. We are not giving a thorough analysis of which parameter combinations are impossible since this also depends on the details of the HPP and which specific unitary U_i is replaced. However, the impossibility of finding examples for certain solutions seems to be rare (especially for small n) and we present in the next subsection a way to circumvent this issue by changing the underlying HPP.

5.B.1. Changing the underlying HPP

x / y	$x = 0$	$x = 1$	Examples		
	$(\Pi_0 = \Pi_0)$	$(\Pi_1 = (-1)^y\, \Pi_0)$	U_0	U_1	U_2
$y = 0$	1	1	σ_y	σ_z	σ_z
$y = 1$	1	-1	σ_x	σ_y	σ_z

Table 5.3.: The Hadamard matrix for the HPP with $\Pi_0 := U_2U_1U_0$ and $\Pi_1 := U_0U_1U_2$. It is promised that $\Pi_x = (-1)^{x \cdot y}\, \Pi_0$ and the task is to find y. Using our method we obtain HPPs for an arbitrary (odd) number of unitary qubit gates. Note that one can also combine this HPP with the one in Table 5.1. For example, by replacing one of the three unitaries with a pair of commuting or anticommuting unitaries, we obtain a HPP with four unitary gates and four permutations.

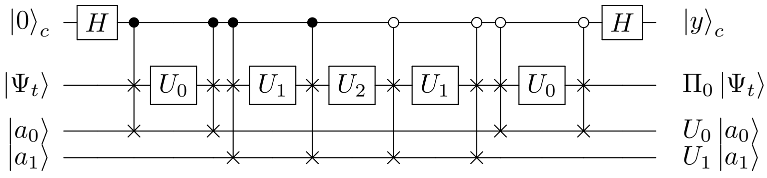


Figure 5.3.: Most efficient causal solution of the HPP in Table 5.3 based on the simulation of the two permutations $\Pi_0 = U_2U_1U_0$ and $\Pi_1 = U_0U_1U_2$. In comparison with the algorithm in Fig. 5.2 of the main text, this algorithm uses only one control qubit instead of two which might be interesting for an experimental realisation. This causal solution is also the one with the smallest number of black-box calls since such an algorithm must be able to determine for each pair of unitary gates whether that pair commutes or anticommutes (by setting the remaining gate to $\mathbb{1}$). This requires to call at least two of the three gates twice (see also Sec. 5.5 of the main text).

Consider the HPP with three unitaries in which $\Pi_0 := U_2U_1U_0$ and $\Pi_1 := U_0U_1U_2$ such that the two permutations satisfy the promise $\Pi_1 = (-1)^y \cdot \Pi_0$ (see Table 5.3). We show that for every HPP that can be generated out of that HPP using our method, it is possible to find a set of 2-dimensional unitaries that satisfy the promise for every parameter combination. As one can see in Table 5.3, for the original HPP with three gates one can find examples of unitary gates that are only of the form $U\sigma_zU^\dagger$. Replacing a unitary of that form by three unitaries that satisfy the promise of the same HPP for $y = 0$ is always possible. We can take for instance $(U\sigma_zU^\dagger)$, $(U\sigma_xU^\dagger)$ and $(U\sigma_xU^\dagger)$:

$$U\sigma_zU^\dagger = (U\sigma_zU^\dagger) \cdot (U\sigma_xU^\dagger) \cdot (U\sigma_xU^\dagger) = (U\sigma_xU^\dagger) \cdot (U\sigma_xU^\dagger) \cdot (U\sigma_zU^\dagger) . \tag{5.29}$$

Similar if the three unitaries shall satisfy the promise for $y = 1$ we can replace $U\sigma_zU^\dagger$ by $\mathbb{1}$, $U\sigma_xU^\dagger$ and $U(i\sigma_y)U^\dagger$:

$$U\sigma_zU^\dagger = \mathbb{1} \cdot (U\sigma_xU^\dagger) \cdot (U(i\sigma_y)U^\dagger) = -(U(i\sigma_y)U^\dagger) \cdot (U\sigma_xU^\dagger) \cdot \mathbb{1} . \tag{5.30}$$

The difference is now that it is also possible to replace a unitary of the form $\mathbb{1}$ into three unitaries that satisfy the promise by themselves. For $y = 0$, we can take for instance $U\sigma_xU^\dagger$, $U\sigma_xU^\dagger$ and $\mathbb{1}$:

$$\mathbb{1} = (U\sigma_xU^\dagger) \cdot (U\sigma_xU^\dagger) \cdot \mathbb{1} = \mathbb{1} \cdot (U\sigma_xU^\dagger) \cdot (U\sigma_xU^\dagger) . \tag{5.31}$$

For the case of $y = 1$, we can replace $\mathbb{1}$ by $U\sigma_xU^\dagger$, $U(i\sigma_y)U^\dagger$ and $U\sigma_zU^\dagger$:

$$\mathbb{1} = (U\sigma_xU^\dagger) \cdot (U(i\sigma_y)U^\dagger) \cdot (U\sigma_zU^\dagger) = -(U\sigma_zU^\dagger) \cdot (U(i\sigma_y)U^\dagger) \cdot (U\sigma_xU^\dagger) . \tag{5.32}$$

Note, however, that $U\sigma_xU^\dagger$ and $U(i\sigma_y)U^\dagger$ are again of the form $U'\sigma_zU'^\dagger$ for an appropriate choice of U' and in a next step these unitaries can be replaced again with three unitaries that satisfy the promise of that HPP by themselves. Therefore, the existence of examples for all possible solutions y of the HPP with n unitaries implies, by induction, the existence of examples for all solutions y of the resulting HPP with $n + 2$ gates.

5.C. Comparison with Fourier promise problems

In the main text, we stated that Hadamard matrices can only exist in dimension 1, 2 and multiples of 4. Furthermore, the existence of a Hadamard matrix for every dimension that is a multiple of 4 is an open problem [185]. However, we use in our generalization of HPPs the fact that Hadamard matrices are simple to construct for powers of 2. This follows from Lemma 5.2 which states that the (tensor) product of two Hadamard matrices is another Hadamard matrix. Since the size of the Hadamard matrix always equals the number of involved permutations, this implies that for example in the case of the 3-switch (see Fig. 5.1 in the main text), we can never make use of all six permutations for this class of problems. Furthermore, there is no known closed form of a Hadamard matrix of dimension $n!$ (which might exist for $n \neq 3$). This fact limits our tasks to use only a subset of all permutations (for example $2^{n-1} < n!$ for the class of all problems that can be generated with our method from the most simplest HPP in Table 5.1).

Nevertheless, one can ask if tasks exists that make use of all permutations. The answer is affirmative and they are called Fourier promise problems [27]. They are defined very similar to HPPs (compare with Eq. (5.1) of the main text): A set of d-dimensional unitary gates $\{U_i\}_0^{n-1}$ is given. The entire set of permutations Π_x is chosen and labeled with $x \in \{0, 1, ..., n! - 1\}$. Furthermore, it is promised that for some $y \in \{0, 1, ..., n! - 1\}$ the following relations hold:

$$\forall x \in \{0, 1, ..., n! - 1\} : \ \Pi_x = \omega^{x \cdot y} \cdot \Pi_0 \,. \tag{5.33}$$

Here, $\omega = e^{2\pi i/n!}$ and the task is to find out y. The crucial difference is that the coefficients $\omega^{x \cdot y}$ form a (discrete) Fourier matrix which exists and can be easily stated for all dimensions.

However, it turns out that the minimal Hilbert space dimension of the target systems must be at least $d = n!$, which makes it impractical for an experimental demonstration. To see this, we can consider $\Pi_1 = \omega^y \, \Pi_0$ and take the determinant on both sides:

$$\det(\Pi_1) = \omega^{y \cdot d} \cdot \det(\Pi_0) \,. \tag{5.34}$$

Since Π_0 and Π_1 are products of the same unitaries in different order, we obtain $\det(\Pi_1) = \det(\Pi_0)$ and therefore $\omega^{y \cdot d} = 1$. Therefore, a solution for every $y \in \{0, 1, ..., n! - 1\}$ can only exist if $d \geq n!$ (remember that $\omega = e^{2\pi i/n!}$) and no examples with qubit gates ($d = 2$) can be found for $n \geq 3$.[3] For further details, we refer to Ref. [27, 28, 184].

Another disadvantage for the class of FPPs is the fact that all involved unitaries satisfy certain pairwise commutation relations (as shown in Ref. [184]):

$$\forall j, k \in \{0, 1, ..., n - 1\} \ \exists \ \alpha_{jk} \in \mathbb{C} \ s.t. : \ U_j U_k = \alpha_{jk} \, U_k U_j \,. \tag{5.35}$$

In Ref. [184] this property is used to find causal solutions for FPPs that call only $O(n \log_2(n))$ black-box unitaries and it was conjectured that all problems of this class

[3] On the contrary, a similar calculation for Hadamard matrices leads to $s(x, y)^d = 1$ where $s(x, y) = \pm 1$ and solutions exist even for $d = 2$.

can be solved with the same efficiency. This property is not (necessarily) satisfied for HPPs, as one can already see in the example for $y = (1,1)$ given in Table 5.2 of the main text, in which for $U_0 = \sigma_y$ and $U_1 = \frac{\sigma_y + \sigma_z}{\sqrt{2}}$ no such relation exists. The intuition for this fact is, roughly speaking, that using only a subset of all permutations implies less promises (there is one promise for every chosen permutation) and therefore less constraints on the structure of the unitary gates. For the case of the HPPs that we generate with our method it turns out that a causal algorithm that calls $O(n \log_2(n))$ gates exists as well. Nevertheless, it shows that for the construction of future tasks that offer a more significant advantage by using indefinite causal structures, considering only a subset of all permutations might be useful.

5.D. Solutions with causal quantum algorithms

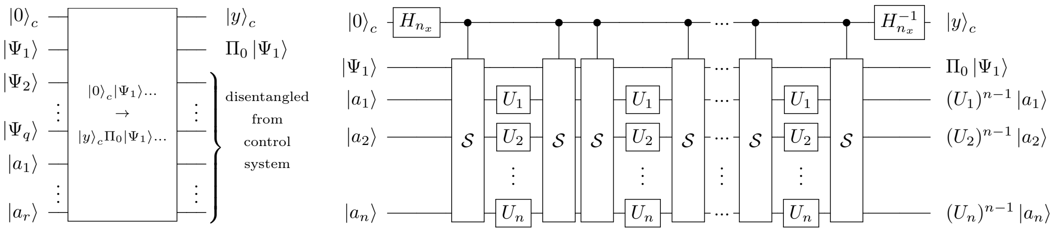
5.D.1. Proof of a lower bound

Lemma 5.1. *Consider the class of all problems that can be generated from the HPP in Table 5.1 with the method introduced in Section 5.3 of the main text. For every HPP (with a set of n different gates) in that class a solution with a causal quantum algorithm has to call at least $2n-1$ unitary gates.*

Proof. We can show by induction, that every solution to that task must be able to determine for every pair of unitary gates whether that pair commutes or anticommutes, when we set all remaining gates to $\mathbb{1}$. For the base case of $n = 2$, we note that there is only the HPP given by Table 5.1 itself for which the statement is clearly correct.

For the induction step, remember that any task with $n+1$ unitary gates is obtained by replacing one unitary U_i from a task with n gates with two unitaries that we denote here as $U_i^{(1)}$ and $U_i^{(2)}$. We can check that a solution to the new task must be able to determine for every pair of gates whether that pair commutes or anticommutes: (1) If the solution for the task with n gates is able to determine for every pair of unitaries whether they commute or anticommute, a solution to the new task with $n+1$ gates is able to determine for every pair U_j and U_k with $j,k \neq i$ this property when we set $U_i^{(1)} = U_i^{(2)} = \mathbb{1}$. (2) Similar, a solution to the new task is able to determine whether $U_i^{(1)}$ and U_j (for every $j \neq i$) commute or anticommute when we set $U_i^{(2)} = \mathbb{1}$. The analog argument holds for $U_i^{(2)}$ and every U_j with $j \neq i$. (3) For the remaining pair of $U_i^{(1)}$ and $U_i^{(2)}$, this follows by construction of the task since part of the solution of the new task is exactly to determine whether $U_i^{(1)}$ and $U_i^{(2)}$ commute or anticommute. This proves the induction hypothesis.

Since a causal algorithm that is able to determine whether two gates commute or anticommute has to call at least one of the two gates twice [26], this requires, in total, to call at least $n-1$ gates twice. Therefore, at least $2n-1$ gates have to be called in total. □



a) Structure of our solutions

b) Solving every HPP with the simulation of the n-switch

Figure 5.4.: The structure of the algorithms we construct here is given in Fig. 5.4 a): All target and auxiliary systems are initialized in an arbitrary d-dimension state. After the algorithm is applied, the control system ends up in the state $|y\rangle_c$ from which the solution can be read out by a measurement in the computational basis. In addition, it is important for our proof that the first target system ends up in the state $\Pi_0|\Psi_1\rangle$ where Π_0 is the identity permutation of the corresponding HPP. One such algorithm is based on the simulation of the quantum-n-switch and given in Fig. 5.4 b): The Hadamard transform H_{n_x} maps the initial state of the control system to an equal superposition of all considered permutations. Afterwards, the permutation $\Pi_x = U_{\sigma_x(n)}...U_{\sigma_x(2)}U_{\sigma_x(1)}$ is applied on $|\Psi_1\rangle$ by swapping the target system $|\Psi_1\rangle$ in each step $i = 1, 2, ..., n$ with the corresponding auxiliary system $\left|a_{\sigma_x(i)}\right\rangle$. Since each auxiliary system $|a_i\rangle$ is swapped with the target system exactly once, the gate U_i acts on $|a_i\rangle$ exactly $n-1$ times and ends up in the state $(U_i)^{n-1}|a_i\rangle$, independent of the state of the control system. Due to the promise $\Pi_x = s(x, y)\,\Pi_0$, the final state of the control and target system can be rewritten into $|y\rangle_c \otimes \Pi_0|\Psi_1\rangle$ (same calculation as in Eq. (5.20) and Eq. (5.21) in the main text) as required for the algorithm in Fig. 5.4 a). (To avoid confusion, we want to mention that we label the unitaries in this section (for convenience) with $1, 2, ..., n$.)

5.D.2. Proof of an upper bound

In this section, we will show that all HPPs that we can generate with our method from a finite set of HPPs (we call them "fundamental" here) can be solved with a causal quantum algorithm and $O(n\log_2(n))$ calls to the black-box gates. The fundamental HPP can be for example only the one given in Table 5.1 and then the HPP given in Table 5.2 of the main text is an example of a (non-fundamental) task in that class. One can also consider the class of all tasks generated by the two fundamental HPPs given in Table 5.1 *and* Table 5.3 which contains more tasks. Also other HPPs, not explicitly stated in this work, can be included.

Theorem 5.1. *A finite set of HPPs is given and we consider the class of problems that can be generated from these fundamental HPPs with the method introduced in Section 5.3 of the main text. Let k_{max} be the number of unitary gates contained in the fundamental HPP with the most gates and let $C := 2\cdot(k_{max}-1)$. For every problem (with a set of n*

unitary gates) in that class there exists a causal quantum algorithm that solves this task by calling at most $C \cdot n \cdot \log_2(n)$ gates.

Proof. We construct for any task in that class a causal algorithm that has the form given in Fig. 5.4 a) and calls at most $C \cdot n \cdot \log_2(n)$ gates. More precisely, given that such an algorithm exists for every problem in that class with at most $n-1$ unitary gates, we construct a solution for the task with n gates and the hypothesis follows by induction.

Base case:

For the base case, we consider all HPPs in that class that contain not more than k_{max} gates (hence $2 \leq n \leq k_{max}$). For these tasks, there exists a solution with a causal algorithm that calls n^2 gates in total. In fact, one can use the simulation of the quantum-n-switch given in Fig. 5.4 b). The hypothesis holds since $C \cdot n \cdot \log_2(n) = 2 \cdot (k_{max}-1) \cdot n \cdot \log_2(n) \geq 2 \cdot (k_i - 1) \cdot n \geq n^2$ (note that $k_{max} \geq n \geq 2$ and $\log_2(n) \geq 1$).

Induction step:

Consider any HPP with n involved black-box gates and denote the permutations involved in this task as Π_x (with $x \in \{0, 1, ..., n_x - 1\}$) and the promise as $\Pi_x = s(x,y)\ \Pi_0$. Since the HPP is constructed with our method, there is a fundamental HPP from which everything starts. Let's denote the identity permutation of this HPP as $\tilde{\Pi}_0$ and let $k \leq k_{max}$ be the number of unitaries in that HPP (note that, for convenience, we label these unitaries with $1, 2, ..., k$ instead of $0, 1, ..., k-1$):

$$\tilde{\Pi}_0 = \tilde{U}_k \tilde{U}_{k-1} ... \tilde{U}_2 \tilde{U}_1 \,. \tag{5.36}$$

The permutations $\tilde{\Pi}_{\tilde{x}} = \tilde{U}_{\tilde{\sigma}_{\tilde{x}}(k)} ... \tilde{U}_{\tilde{\sigma}_{\tilde{x}}(2)} \tilde{U}_{\tilde{\sigma}_{\tilde{x}}(1)}$ of that starting HPP are permutations of the unitaries $\tilde{U}_i$ and satisfy the following relations:

$$\tilde{\Pi}_{\tilde{x}} = \tilde{s}(\tilde{x}, \tilde{y})\ \tilde{\Pi}_0 \,. \tag{5.37}$$

Now, by applying our method, each unitary $\tilde{U}_i$ is replaced, step by step, with the permutations of other HPPs. It is important to note that these permutations form by themselves a HPP of the same class (but with less unitaries $n_i < n$):

$$\forall i \in \{1, 2, ..., k\} : \tilde{U}_i \to \Pi^{(i)}_{x_i} = s_i(x_i, y_i)\ \Pi^{(i)}_0 \,. \tag{5.38}$$

Therefore, we can write the label x as $(\tilde{x}, x_1, x_2, ..., x_k)$ and the permutation Π_x is exactly obtained by taking $\tilde{\Pi}_{\tilde{x}}$ and replacing each unitary $\tilde{U}_i$ with the corresponding permutation $\Pi^{(i)}_{x_i}$. In this way, we obtain:

$$\Pi_x = \Pi_{(\tilde{x}, x_1, x_2, ..., x_k)} = \tilde{s}(\tilde{x}, \tilde{y}) \cdot \left(\prod_{i=1}^{k} s_i(x_i, y_i) \right) \Pi_{0,0,0,...,0} \tag{5.39}$$

$$\implies s(x,y) = \tilde{s}(\tilde{x}, \tilde{y}) \cdot \left(\prod_{i=1}^{k} s_i(x_i, y_i) \right) \,. \tag{5.40}$$

Hence, solving the task is equivalent to find all values of $\tilde{y}, y_1, y_2, ..., y_{k-1}$ and y_k. Since the permutations $\Pi^{(i)}_{x_i} = s_i(x_i, y_i)\ \Pi^{(i)}_0$ form a HPP of the same class with $n_i \leq n-1$ involved unitaries, there is, by the induction hypothesis, for each i a causal algorithm that finds y_i with $C \cdot n_i \cdot \log_2(n_i)$ queries.

To find the remaining value $\tilde{y}$, more work is required. Let j be the index of the block that contains the most unitaries ($n_j = \max\limits_{1\leq i\leq k} \{n_i\}$). If this index is not unique, one can choose one of them. Since $\Pi_{\tilde{x},0,0,...,0}$ is a permutation of the blocks $\Pi^{(1)}_0$, $\Pi^{(2)}_0$, ..., $\Pi^{(k-1)}_0$, $\Pi^{(k)}_0$ and $\Pi_{\tilde{x},0,0,...,0} = \tilde{s}(\tilde{x}, \tilde{y})\ \Pi_{0,0,0,...,0}$ we are able to find $\tilde{y}$, when we are able to simulate all permutations of the blocks $\Pi^{(1)}_0$, $\Pi^{(2)}_0$, ..., $\Pi^{(k)}_0$. This is achieved in the upper part of the algorithm in Figure 5.5 by a particular simulation of the quantum-k-switch build out of two simulations of the quantum-$(k-1)$-switch. Here, in each step i, depending on the state of the control system $|\tilde{x}\rangle$, the corresponding block $\Pi^{(\tilde{\sigma}_{\tilde{x}}(i))}_0$ (with $\tilde{\sigma}_{\tilde{x}}(i) \neq j$) is applied on the target system $\left|\Psi^{(j)}_1\right\rangle$ by swapping $\left|\Psi^{(j)}_1\right\rangle$ with the corresponding auxiliary system $\left|\tilde{a}_{\tilde{\sigma}_{\tilde{x}}(i)}\right\rangle$. At the point where the block $\Pi^{(j)}_0$ shall be applied on $\left|\Psi^{(j)}_1\right\rangle$, the algorithm in the middle is used to realize this transformation. The first part requires at most $k-1$ steps since there are at most $k-1$ blocks in $\Pi_{\tilde{x},0,0,...,0}$ before $\Pi^{(j)}_0$. Afterwards the same procedure is used to simulate all blocks $\Pi^{(i)}_0$ that appear after $\Pi^{(j)}_0$, which requires again at most $k-1$ steps. Since each auxiliary system $|\tilde{a}_i\rangle$ is swapped exactly once, it ends up in the state $(\Pi^{(i)}_0)^{2(k-1)-1}\,|\tilde{a}_i\rangle$, independent of the state of the control system $|\tilde{x}\rangle$.

In total, this algorithm consumes

$$\sum_{i=1}^{k} C \cdot n_i \cdot \log_2 n_i + \sum_{i=1, i\neq j}^{k} 2 \cdot (k-1) \cdot n_i \leq C \cdot n \cdot \log_2 n \tag{5.41}$$

queries. The first term corresponds to the algorithms that find all values of y_i and the second term comes from the simulation of all permutations of the blocks $\Pi^{(1)}_0$, $\Pi^{(2)}_0$, ..., $\Pi^{(k)}_0$ (each block contains n_i black-box gates and appears $2 \cdot (k-1)$ times, except $\Pi^{(j)}_0$ which does not appear at all). To see that this expression is smaller than $C \cdot n \cdot \log_2 n$, we observe that:

$$C \cdot n \cdot \log_2 n - \sum_{i=1}^{k} C \cdot n_i \cdot \log_2 n_i = C \cdot \left(\sum_{i=1}^{k} n_i\right) \cdot \log_2 n - C \cdot \sum_{i=1}^{k} n_i \cdot \log_2 n_i \tag{5.42}$$

$$= C \cdot \left(\sum_{i=1}^{k} n_i \cdot \log_2 \frac{n}{n_i}\right) \geq C \cdot \left(\sum_{i=1, i\neq j}^{k} n_i \cdot \log_2 \frac{n}{n_i}\right) \geq C \cdot \left(\sum_{i=1, i\neq j}^{k} n_i \cdot \log_2 2\right) \tag{5.43}$$

$$= 2 \cdot (k_{max} - 1) \cdot \left(\sum_{i=1, i\neq j}^{k} n_i\right) \geq \sum_{i=1, i\neq j}^{k} 2 \cdot (k-1) \cdot n_i\,. \tag{5.44}$$

Here, we have used that $\frac{n}{n_i} \geq 2$ for every $i \neq j$ (since $n_j \geq n_i$ and $n \geq n_j + n_i$). This concludes the proof. □

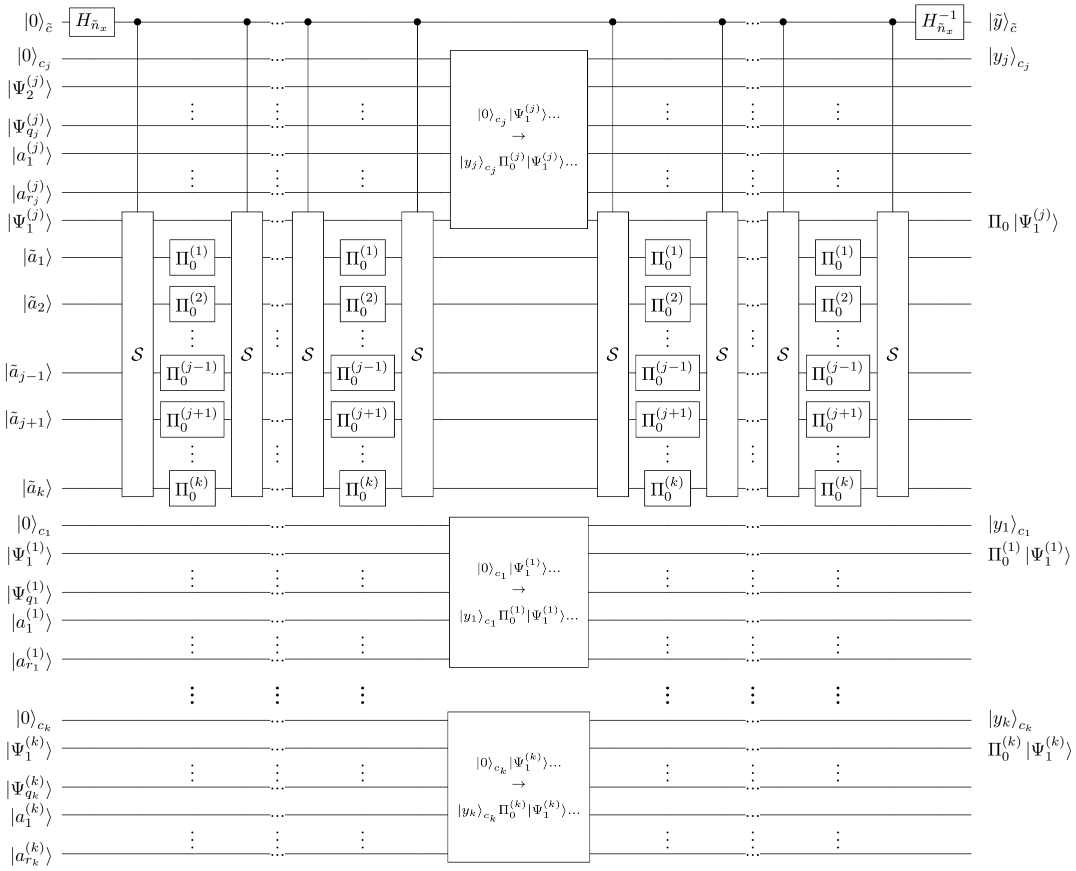

$|0\rangle_{\tilde{c}}$
$H_{\tilde{n}_x}$
$|0\rangle_{c_j}$
$|\Psi_2^{(j)}\rangle$
$|\Psi_{q_j}^{(j)}\rangle$
$|a_1^{(j)}\rangle$
$|a_{r_j}^{(j)}\rangle$
$|\Psi_1^{(j)}\rangle$
$|\tilde{a}_1\rangle$
$|\tilde{a}_2\rangle$
$|\tilde{a}_{j-1}\rangle$
$|\tilde{a}_{j+1}\rangle$
$|\tilde{a}_k\rangle$
$|0\rangle_{c_1}$
$|\Psi_1^{(1)}\rangle$
$|\Psi_{q_1}^{(1)}\rangle$
$|a_1^{(1)}\rangle$
$|a_{r_1}^{(1)}\rangle$
$|0\rangle_{c_k}$
$|\Psi_1^{(k)}\rangle$
$|\Psi_{q_k}^{(k)}\rangle$
$|a_1^{(k)}\rangle$
$|a_{r_k}^{(k)}\rangle$
$\mathcal{S}$
$\Pi_0^{(1)}$
$\Pi_0^{(2)}$
$\Pi_0^{(j-1)}$
$\Pi_0^{(j+1)}$
$\Pi_0^{(k)}$
$|0\rangle_{c_j}|\Psi_1^{(j)}\rangle\ldots \rightarrow |y_j\rangle_{c_j}\Pi_0^{(j)}|\Psi_1^{(j)}\rangle\ldots$
$|0\rangle_{c_1}|\Psi_1^{(1)}\rangle\ldots \rightarrow |y_1\rangle_{c_1}\Pi_0^{(1)}|\Psi_1^{(1)}\rangle\ldots$
$|0\rangle_{c_k}|\Psi_1^{(k)}\rangle\ldots \rightarrow |y_k\rangle_{c_k}\Pi_0^{(k)}|\Psi_1^{(k)}\rangle\ldots$
$H_{\tilde{n}_x}^{-1}$
$|\tilde{y}\rangle_{\tilde{c}}$
$|y_j\rangle_{c_j}$
$\Pi_0|\Psi_1^{(j)}\rangle$
$|y_1\rangle_{c_1}$
$\Pi_0^{(1)}|\Psi_1^{(1)}\rangle$
$|y_k\rangle_{c_k}$
$\Pi_0^{(k)}|\Psi_1^{(k)}\rangle$

Figure 5.5.: The algorithm that finds $y = (\tilde{y}, y_1, y_2, ..., y_k)$: The values y_i for $i \neq j$ are found in the lower part of the algorithm completely independent of the rest (the index j is skipped in the lower part). In the upper part, we simulate all possible permutations of the blocks $\Pi_0^{(1)}$, $\Pi_0^{(2)}$, ..., $\Pi_0^{(k)}$ (which is sufficient to determine $\tilde{y}$) by a construction that simulates the quantum-k-switch with two simulations of the quantum-$(k-1)$-switch. Here, in each step i and depending on the state of the control system $|\tilde{x}\rangle$ the corresponding block $\Pi_0^{(\tilde{\sigma}_{\tilde{x}}(i))}$ (with $\tilde{\sigma}_{\tilde{x}}(i) \neq j$) is applied on $\left|\Psi_1^{(j)}\right\rangle$ by swapping that target system with $\left|\tilde{a}_{\tilde{\sigma}_{\tilde{x}}(i)}\right\rangle$. The block $\Pi_0^{(j)}$ is applied in the middle step. In this way, the target system $\left|\Psi_1^{(j)}\right\rangle$ ends up in the state $\Pi_0\left|\Psi_1^{(j)}\right\rangle$ and the solution $\tilde{y}$ can be read out in the control system.

6. Conclusion

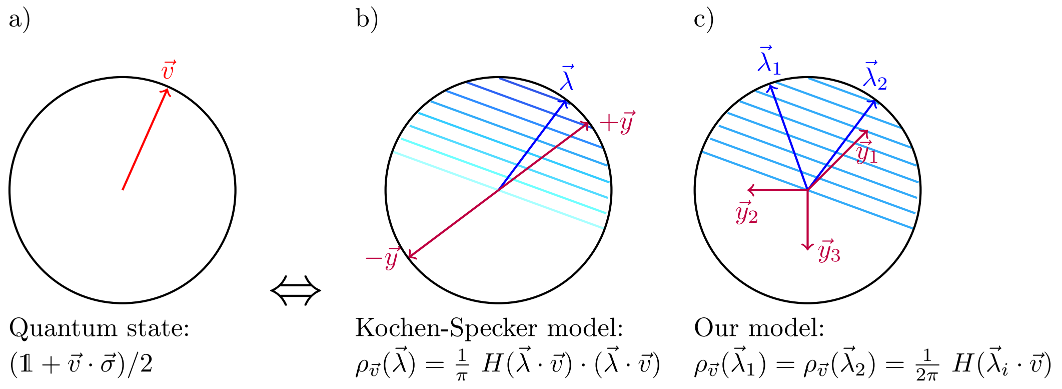
In the first part of this thesis, we investigate the classical cost of reproducing the quantum correlations arising from local measurements on two-qubit states. First, we studied the prepare-and-measure scenario and found tight bounds on the communication cost. In fact, we proved that two bits are necessary and sufficient to simulate all qubit strategies. We used this simulation technique to show that arbitrary local measurements on any entangled two-qubit state can be simulated by the same amount, namely two classical bits. In our protocols, each party can choose an arbitrary measurement and we do not require any knowledge of what the other party is going to measure on the other qubit. In some special cases, we have shown that the communication cost is even smaller. Namely, in Chapter 2 we proved that for the restricted class of projective measurements, already a single trit is sufficient. In addition, if the state is either weakly or maximally entangled, the communication cost reduces further to just a single bit. One important open problem is to decide whether all two-qubit states can be simulated with a single bit of classical communication. For this conjecture, a neural network approach provides strong numerical evidence that this is indeed the case [112]. In principle, it is also possible that even when we consider the most general class of measurements (POVMs) just one classical bit of communication might be sufficient to simulate all correlations.

There are several other ways this result could be extended. Our protocols apply only to the most fundamental scenarios involving at most two qubits. However, the question of how much classical communication is required to simulate quantum correlations can be asked in every scenario, involving arbitrary quantum systems and an arbitrary number of parties. The communication cost for many of these scenarios is unknown. Already if the qubit in the prepare-and-measure scenario is replaced by a qutrit, a three-level quantum system, it is not known whether these correlations can be simulated with a finite classical message. Albeit, they can be reproduced with a communication cost that is finite on average [48], in the worst case, they might require an infinite communication cost. These problems are relevant in many ways, for instance, for the question of bounding the quantum to classical advantage in communication tasks. In this work, we proved that all correlations that can be achieved by sending a qubit can also be reproduced by sending two classical bits of communication. Hence, in any such communication task, two classical bits perform at least as good as one qubit. Can we find similar bounds for high-dimensional quantum systems? It is known that higher dimensional quantum systems can perform exponentially better in certain communication tasks compared to classical systems [12, 13]. However, in principle, such an advantage could be even larger. Can we find another communication task, in which quantum systems provide an even larger advantage than classical systems? Or, on the other hand, can we achieve the correlations obtained by these systems with a classical message of a certain length?

Other important directions to explore are related scenarios involving several parties. In Chapter 1 we consider prepare-and-measure scenarios, where one party sends a quantum state to the other party, which measures the received state. How does the communication cost change, when there is another party in the middle, that can apply an arbitrary transformation on the state before it is sent to the party, that measures the state? Our simulation protocols in Chapter 2 apply to two-qubit states. About multipartite qubit states, it is known that equatorial measurements on a Greenberger-Horne-Zeilinger state can be simulated with three bits of communication [55]. How much communication is necessary when general three-qubit states and general qubit measurements are considered, or even a multipartite state of a larger local dimension? Again, it is known that these scenarios can be simulated with a finite communication cost on average [58]. However, it is unknown if the communication cost remains finite in the worst case.

It is important to recognize that our simulation techniques are directly linked to hidden variable models of the qubit. In fact, the simulation techniques we developed in Chapter 2 make direct use of the Kochen-Specker model [110]. More precisely, a simulation of the correlations reduces to an efficient way of sampling the Kochen-Specker model for a given qubit state. However, this model can only reproduce the correlations of projective measurements. As a byproduct of our work, we obtain a new and even simple model of the qubit capable of reproducing the outcomes of generalized measurements. Namely, in Chapter 1 we discovered that two vectors distributed uniformly on the positive hemisphere centered around the Bloch vector of the given quantum state (see Fig. 1.2 and Fig. 6.1) are sufficient to simulate the effect of general POVMs applied to that state. This is a natural extension of the Kochen-Specker model to the most general class of quantum measurements. We believe this model has applications beyond the task of simulating quantum correlations.

In Chapter 3, we turned our attention to a closely related question, the development of local models for entangled quantum states. The seminal work of Werner [17] showed that there are quantum states that are entangled but do not violate any Bell inequality for local projective measurements. He proves this by providing an explicit local hidden variable model that can reproduce the quantum correlations for all projective measurements. In Chapter 3, we extend this model to generalized quantum measurements (POVMs). It is worth noting that our model determines the exact threshold for which two-qubit Werner states can demonstrate quantum steering. Interestingly, our result implies that generalized measurements are not more powerful than projective measurements to demonstrate quantum steering for these states. Namely, whenever a two-qubit Werner state can demonstrate quantum steering, this effect can be demonstrated by using only projective measurements. It is still an important open problem if there exists a quantum state that admits a local hidden state model for all projective measurements but can demonstrate quantum steering when generalized measurements are considered. The analog problem for Bell nonlocality is also open. Does there exist a quantum state that admits a local hidden variable model for all projective measurements but violates a Bell inequality when generalized measurements (POVMs) are considered?



Quantum state:
$(\mathbb{1} + \vec{v} \cdot \vec{\sigma})/2$

Kochen-Specker model:
$\rho_{\vec{v}}(\vec{\lambda}) = \frac{1}{\pi}\, H(\vec{\lambda} \cdot \vec{v}) \cdot (\vec{\lambda} \cdot \vec{v})$

Our model:
$\rho_{\vec{v}}(\vec{\lambda}_1) = \rho_{\vec{v}}(\vec{\lambda}_2) = \frac{1}{2\pi}\, H(\vec{\lambda}_i \cdot \vec{v})$

Figure 6.1.: **a)** An arbitrary qubit state represented by its Bloch vector $\vec{v}$. **b)** The Kochen-Specker model associates a hidden variable $\vec{\lambda}$ to this quantum state according to a certain distribution $\rho_{\vec{v}}(\vec{\lambda})$ ($H(x)$ denotes the Heaviside function). This model can simulate arbitrary projective measurements applied to that state. **c)** In Chapter 1, we introduce a model capable of reproducing the statistics for arbitrary qubit measurements (POVMs). The model consists of two vectors $\vec{\lambda}_i$ distributed independently and uniformly in the positive hemisphere centered around $\vec{v}$.

In the second part of this thesis, we explored quantum computation using indefinite causal structures. It was known that the quantum-controlled gate orderings, which are an instance of indefinite causal structures can boost the capabilities of quantum computing even further. For instance, they can solve certain oracle tasks, such as determining whether two gates commute or anticommute, by making fewer calls to the oracles than causally ordered quantum algorithms [26]. Since then other tasks were found that were expected to generalize this effect to an arbitrary number of gates. This is the topic of the second part of this thesis. In Chapter 4, we studied a class of problems that are called Fourier Promise Problems, introduced by Araújo, Costa, and Brukner [27]. They were expected to provide a quadratic advantage of using the quantum switch compared to casually ordered quantum algorithms. However, we found more efficient causally ordered algorithms for problems of this class, hence showing that the advantage is smaller than previously expected.

In Chapter 5, we study another closely related class of problems called Hadamard Promise Problems [28]. They can also be solved efficiently by using the quantum switch and were expected to provide an advantage compared to causally ordered algorithms. However, we showed that for these problems the advantage is smaller than previously expected. We introduced a general procedure to obtain problem instances of this class and proved that they can be solved by using a causally ordered algorithm and calling each gate $O(\log n)$ times. This raises the question of whether other tasks might offer a larger advantage by using indefinite causal structures. At the same time, it is also interesting to explore whether these tasks can be used for other information-processing tasks involving indefinite causal structures. For instance, Guérin et al. [156] showed that the problem of determining whether two gates commute or anticommute can be translated into a communication task that provides an exponential advantage compared to causally ordered protocols. It is then interesting to ask, whether the problems we introduce in Chapter 5 can lead to a similar advantage for communication tasks involving several parties. Connected to that, one could also ask if these rather abstract oracle problems can be translated into a more practical quantum algorithm capable of solving more natural problems. Usually, oracle problems are introduced to study the power of a new computational scheme. For instance, the power of quantum compared to classical computation was first demonstrated by certain oracle problems, including Deutsch [187], Deutsch-Josza [188], Simon [189], and Bernstein-Vazirani [190]. Later, they were used to find quantum algorithms for more useful tasks. Most famously, Shor was inspired by Simon's problem to find his polynomial quantum algorithm that can find the prime factors of large numbers [24]. Do similar real-world tasks also exist for the rather abstract Fourier and Hadamard Promise Problem?

Bibliography

[1] Werner Heisenberg. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik*, 43:172–198, March 1927.

[2] Erwin Schrödinger. Die gegenwärtige Situation in der Quantenmechanik. *Naturwissenschaften*, 23(48):807–812, November 1935.

[3] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47(10):777–780, May 1935.

[4] John S. Bell. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1(3):195–200, November 1964.

[5] Artur K. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67(6):661–663, August 1991.

[6] Stefano Pironio, Antonio Acín, Serge Massar, A. Boyer de La Giroday, Dzmitry N. Matsukevich, Peter Maunz, Steven Olmschenk, David Hayes, Le Luo, T. Andrew Manning, and Christopher Monroe. Random numbers certified by Bell's theorem. *Nature*, 464:1021–1024, April 2010.

[7] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, October 2018.

[8] Tim Maudlin. Bell's inequality, information transmission, and prism models. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 1992(1):404–417, 1992.

[9] Gilles Brassard, Richard Cleve, and Alain Tapp. Cost of Exactly Simulating Quantum Entanglement with Classical Communication. *Physical Review Letters*, 83(9):1874–1877, August 1999.

[10] Ben F. Toner and Dave Bacon. Communication Cost of Simulating Bell Correlations. *Physical Review Letters*, 91(18):187904, October 2003.

[11] André A. Méthot. Simulating POVMs on EPR pairs with 5.7 bits of expected communication. *European Physical Journal D*, 29(3):445–446, June 2004.

[12] Ziv Bar-Yossef, Thathachar S. Jayram, and Iordanis Kerenidis. Exponential separation of quantum and classical one-way communication complexity. In *Proceedings*

of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC '04, page 128–137, New York, NY, USA, 2004. Association for Computing Machinery.

[13] Ran Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 358–367, May 1999.

[14] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

[15] Nicolas J. Cerf, Nicolas Gisin, and Serge Massar. Classical Teleportation of a Quantum Bit. *Physical Review Letters*, 84(11):2521–2524, March 2000.

[16] Nicolas Gisin. Bell's inequality holds for all non-product states. *Physics Letters A*, 154(5-6):201–202, April 1991.

[17] Reinhard F. Werner. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. *Physical Review A*, 40(8):4277–4281, October 1989.

[18] Howard M. Wiseman, Steve J. Jones, and Andrew C. Doherty. Steering, Entanglement, Nonlocality, and the Einstein-Podolsky-Rosen Paradox. *Physical Review Letters*, 98(14):140402, April 2007.

[19] Roope Uola, Ana C. S. Costa, H. Chau Nguyen, and Otfried Gühne. Quantum steering. *Reviews of Modern Physics*, 92(1):015001, January 2020.

[20] Paul Busch. Unsharp reality and joint measurements for spin observables. *Physical Review D*, 33(8):2253–2261, April 1986.

[21] Lucien Hardy. Probability theories with dynamic causal structure: A new framework for quantum gravity. *arXiv:gr-qc/0509120*, September 2005.

[22] Ognyan Oreshkov, Fabio Costa, and Časlav Brukner. Quantum correlations with no causal order. *Nature Communications*, 3:1092, October 2012.

[23] Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti, and Benoit Valiron. Quantum computations without definite causal structure. *Physical Review A*, 88(2):022318, August 2013.

[24] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[25] 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, page 212–219, New York, NY, USA, July 1996. Association for Computing Machinery.

[26] Giulio Chiribella. Perfect discrimination of no-signalling channels via quantum superposition of causal structures. *Physical Review A*, 86(4):040301, October 2012.

[27] Mateus Araújo, Fabio Costa, and Časlav Brukner. Computational advantage from quantum-controlled ordering of gates. *Physical Review Letters*, 113(25):250402, December 2014.

[28] Márcio M. Taddei, Jaime Cariñe, Daniel Martínez, Tania García, Nayda Guerrero, Alastair A. Abbott, Mateus Araújo, Cyril Branciard, Esteban S. Gómez, Stephen P. Walborn, Leandro Aolita, and Gustavo Lima. Computational advantage from the quantum superposition of multiple temporal orders of photonic gates. *PRX Quantum*, 2(1):010320, February 2021.

[29] Richard Cleve and Harry Buhrman. Substituting quantum entanglement for communication. *Physical Review A*, 56(2):1201–1204, August 1997.

[30] Harry Buhrman, Richard Cleve, and Wim van Dam. Quantum entanglement and communication complexity. *SIAM Journal on Computing*, 30(6):1829–1841, January 2001.

[31] Toby S. Cubitt, Debbie Leung, William Matthews, and Andreas Winter. Improving Zero-Error Classical Communication with Entanglement. *Physical Review Letters*, 104(23):230503, June 2010.

[32] Armin Tavakoli, Jef Pauwels, Erik Woodhead, and Stefano Pironio. Correlations in entanglement-assisted prepare-and-measure scenarios. *PRX Quantum*, 2(4):040357, December 2021.

[33] Charles H. Bennett, Peter W. Shor, John A. Smolin, and Ashish V. Thapliyal. Entanglement-assisted capacity of a quantum channel and the reverse shannon theorem. *IEEE Transactions on Information Theory*, 48(10):2637–2655, 2002.

[34] Amélie Piveteau, Jef Pauwels, Emil Hâkansson, Sadiq Muhammad, Mohamed Bourennane, and Armin Tavakoli. Entanglement-assisted quantum communication with simple measurements. *Nature Communications*, 13:7878, December 2022.

[35] Charles H. Bennett and Stephen J. Wiesner. Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. *Physical Review Letters*, 69(20):2881–2884, November 1992.

[36] Harry Buhrman, Richard Cleve, Serge Massar, and Ronald de Wolf. Nonlocality and communication complexity. *Reviews of Modern Physics*, 82(1):665–698, January 2010.

[37] Gilles Brassard. Quantum communication complexity. *Foundations of Physics*, 33(11):1593–1616, November 2003.

[38] Thomas Vidick and Stephanie Wehner. Does Ignorance of the Whole Imply Ignorance of the Parts? Large Violations of Noncontextuality in Quantum Theory. *Physical Review Letters*, 107(3):030402, July 2011.

[39] Armin Tavakoli, Alley Hameedi, Breno Marques, and Mohamed Bourennane. Quantum Random Access Codes Using Single d-Level Systems. *Physical Review Letters*, 114(17):170502, May 2015.

[40] Miguel Navascués and Tamás Vértesi. Bounding the Set of Finite Dimensional Quantum Correlations. *Physical Review Letters*, 115(2):020501, July 2015.

[41] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, July 2002.

[42] Andris Ambainis, Debbie Leung, Laura Mancinska, and Maris Ozols. Quantum Random Access Codes with Shared Randomness. *arXiv:0810.2937*, October 2008.

[43] Rodrigo Gallego, Nicolas Brunner, Christopher Hadley, and Antonio Acín. Device-Independent Tests of Classical and Quantum Dimensions. *Physical Review Letters*, 105(23):230501, December 2010.

[44] Marcin Pawłowski and Nicolas Brunner. Semi-device-independent security of one-way quantum key distribution. *Physical Review A*, 84(1):010302, July 2011.

[45] Hong-Wei Li, Zhen-Qiang Yin, Yu-Chun Wu, Xu-Bo Zou, Shuang Wang, Wei Chen, Guang-Can Guo, and Zheng-Fu Han. Semi-device-independent random-number expansion without entanglement. *Physical Review A*, 84(3):034301, September 2011.

[46] Erik Woodhead and Stefano Pironio. Secrecy in Prepare-and-Measure Clauser-Horne-Shimony-Holt Tests with a Qubit Bound. *Physical Review Letters*, 115(15):150501, October 2015.

[47] Armin Tavakoli, Jędrzej Kaniewski, Tamás Vértesi, Denis Rosset, and Nicolas Brunner. Self-testing quantum states and measurements in the prepare-and-measure scenario. *Physical Review A*, 98(6):062307, December 2018.

[48] Serge Massar, Dave Bacon, Nicolas J. Cerf, and Richard Cleve. Classical simulation of quantum entanglement without local hidden variables. *Physical Review A*, 63(5):052305, May 2001.

[49] Arun K. Pati. Minimum classical bit for remote preparation and measurement of a qubit. *Physical Review A*, 63(1):014302, December 2000.

[50] Nicolas Gisin and Bernard Gisin. A local hidden variable model of quantum correlation exploiting the detection loophole. *Physics Letters A*, 260(5):323–327, September 1999.

[51] Michael Steiner. Towards quantifying non-local information transfer: finite-bit non-locality. *Physics Letters A*, 270(5):239–244, June 2000.

[52] Julien Degorre, Sophie Laplante, and Jérémie Roland. Simulating quantum correlations as a distributed sampling problem. *Physical Review A*, 72(6):062314, December 2005.

[53] Julien Degorre, Sophie Laplante, and Jérémie Roland. Classical simulation of traceless binary observables on any bipartite quantum state. *Physical Review A*, 75(1):012309, January 2007.

[54] Oded Regev and Ben F. Toner. Simulating quantum correlations with finite communication. *SIAM Journal on Computing*, 39(4):1562–1580, 2010.

[55] Cyril Branciard and Nicolas Gisin. Quantifying the Nonlocality of Greenberger-Horne-Zeilinger Quantum Correlations by a Bounded Communication Simulation Protocol. *Physical Review Letters*, 107(2):020401, July 2011.

[56] Cyril Branciard, Nicolas Brunner, Harry Buhrman, Richard Cleve, Nicolas Gisin, Samuel Portmann, Denis Rosset, and Mario Szegedy. Classical Simulation of Entanglement Swapping with Bounded Communication. *Physical Review Letters*, 109(10):100401, September 2012.

[57] Gilles Brassard, Luc Devroye, and Claude Gravel. Exact classical simulation of the quantum-mechanical ghz distribution. *IEEE Transactions on Information Theory*, 62(2):876–890, February 2016.

[58] Gilles Brassard, Luc Devroye, and Claude Gravel. Remote Sampling with Applications to General Entanglement Simulation. *Entropy*, 21(1):92, January 2019.

[59] Giacomo Mauro D'Ariano, Paoloplacido Lo Presti, and Paolo Perinotti. Classical randomness in quantum measurements. *Journal of Physics A: Mathematical and General*, 38(26):5979–5991, June 2005.

[60] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[61] I.D. Ivanovic. How to differentiate between non-orthogonal states. *Physics Letters A*, 123(6):257–259, August 1987.

[62] Asher Peres. How to differentiate between non-orthogonal states. *Physics Letters A*, 128(1):19, March 1988.

[63] Konrad Banaszek, Giacomo Mauro D'ariano, Matteo G. Paris, and Massimiliano F. Sacchi. Maximum-likelihood estimation of the density matrix. *Physical Review A*, 61(1):010304, December 1999.

[64] Joseph M. Renes. Spherical-code key-distribution protocols for qubits. *Physical Review A*, 70(5):052314, November 2004.

[65] Tamas Vértesi and Erika Bene. Two-qubit Bell inequality for which positive operator-valued measurements are relevant. *Physical Review A*, 82(6):062115, December 2010.

[66] N. Bent, H. Qassim, A. A. Tahir, D. Sych, G. Leuchs, L. L. Sánchez-Soto, E. Karimi, and R. W. Boyd. Experimental realization of quantum tomography of photonic qudits via symmetric informationally complete positive operator-valued measures. *Physical Review X*, 5(4):041006, October 2015.

[67] Antonio Acín, Stefano Pironio, Tamás Vértesi, and Peter Wittek. Optimal randomness certification from one entangled bit. *Physical Review A*, 93(4):040102, April 2016.

[68] Florian J. Curchod, Markus Johansson, Remigiusz Augusiak, Matty J. Hoban, Peter Wittek, and Antonio Acín. Unbounded randomness certification using sequences of measurements. *Physical Review A*, 95(2):020102, February 2017.

[69] Joonwoo Bae, Beatrix C. Hiesmayr, and Daniel McNulty. Linking entanglement detection and state tomography via quantum 2-designs. *New Journal of Physics*, 21(1):013012, January 2019.

[70] Armin Tavakoli. Semi-device-independent certification of independent quantum state and measurement devices. *Physical Review Letters*, 125(15):150503, October 2020.

[71] Armin Tavakoli, Máté Farkas, Denis Rosset, Jean-Daniel Bancal, and Jedrzej Kaniewski. Mutually unbiased bases and symmetric informationally complete measurements in bell experiments. *Science Advances*, 7(7):eabc3847, February 2021.

[72] Miguel Navascués, Adrien Feix, Mateus Araújo, and Tamás Vértesi. Characterizing finite-dimensional quantum behavior. *Physical Review A*, 92(4):042117, October 2015.

[73] Armin Tavakoli, Massimiliano Smania, Tamás Vértesi, Nicolas Brunner, and Mohamed Bourennane. Self-testing nonprojective quantum measurements in prepare-and-measure experiments. *Science Advances*, 6(16):eaaw6664, April 2020.

[74] Piotr Mironowicz and Marcin Pawłowski. Experimentally feasible semi-device-independent certification of four-outcome positive-operator-valued measurements. *Physical Review A*, 100(3):030301, September 2019.

[75] Jonathan Steinberg, H. Chau Nguyen, and Matthias Kleinmann. Minimal scheme for certifying three-outcome qubit measurements in the prepare-and-measure scenario. *Physical Review A*, 104(6):062431, December 2021.

[76] Daniel Martínez, Esteban S. Gómez, Jaime Cariñe, Luciano Pereira, Aldo Delgado, Stephen P. Walborn, Armin Tavakoli, and Gustavo Lima. Certification of a non-projective qudit measurement using multiport beamsplitters. *Nature Physics*, 19(2):190–195, December 2022.

[77] Nicolas Brunner, Daniel Cavalcanti, Stefano Pironio, Valerio Scarani, and Stephanie Wehner. Bell nonlocality. *Reviews of Modern Physics*, 86(2):419–478, April 2014.

[78] Ernesto F. Galvão and Lucien Hardy. Substituting a Qubit for an Arbitrarily Large Number of Classical Bits. *Physical Review Letters*, 90(8):087902, February 2003.

[79] Joseph Bowles, Marco Túlio Quintino, and Nicolas Brunner. Certifying the Dimension of Classical and Quantum Systems in a Prepare-and-Measure Scenario with Independent Devices. *Physical Review Letters*, 112(14):140407, April 2014.

[80] Joseph Bowles, Flavien Hirsch, Marco Túlio Quintino, and Nicolas Brunner. Local Hidden Variable Models for Entangled Quantum States Using Finite Shared Randomness. *Physical Review Letters*, 114(12):120401, March 2015.

[81] Julio I. de Vicente. Shared randomness and device-independent dimension witnessing. *Physical Review A*, 95(1):012340, January 2017.

[82] Ram Krishna Patra, Sahil Gopalkrishna Naik, Edwin Peter Lobo, Samrat Sen, Tamal Guha, Some Sankar Bhattacharya, Mir Alimuddin, and Manik Banik. Classical analogue of quantum superdense coding and communication advantage of a single quantum system. *Quantum*, 8:1315, April 2024.

[83] Péter E. Frenkel and Mihály Weiner. Classical Information Storage in an n-Level Quantum System. *Communications in Mathematical Physics*, 340(2):563–574, December 2015.

[84] Michele Dall'Arno, Sarah Brandsen, Alessandro Tosini, Francesco Buscemi, and Vlatko Vedral. No-Hypersignaling Principle. *Physical Review Letters*, 119(2):020401, July 2017.

[85] Jonathan Barrett. Nonsequential positive-operator-valued measurements on entangled mixed states do not always violate a Bell inequality. *Physical Review A*, 65(4):042302, April 2002.

[86] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[87] Mateus Araújo, Flavien Hirsch, and Marco Túlio Quintino. Bell nonlocality with a single shot. *Quantum*, 4:353, October 2020.

[88] Jessica Bavaresco, Mio Murao, and Marco Túlio Quintino. Strict Hierarchy between Parallel, Sequential, and Indefinite-Causal-Order Strategies for Channel Discrimination. *Physical Review Letters*, 127(20):200504, November 2021.

[89] Wikipedia. https://en.wikipedia.org/wiki/snub_cube, 2022. (accessed 20/06/2022).

[90] Armin Tavakoli and Nicolas Gisin. The Platonic solids and fundamental tests of quantum mechanics. *Quantum*, 4:293, July 2020.

[91] Wikipedia. https://en.wikipedia.org/wiki/thomson_problem, 2022. (accessed 20/06/2022).

[92] Marco T. Quintino. https://github.com/mtcq, 2022.

[93] Avshalom C. Elitzur, Sandu Popescu, and Daniel Rohrlich. Quantum nonlocality for each pair in an ensemble. *Physics Letters A*, 162(1):25–28, January 1992.

[94] Jonathan Barrett, Adrian Kent, and Stefano Pironio. Maximally Nonlocal and Monogamous Quantum Correlations. *Physical Review Letters*, 97(17):170409, October 2006.

[95] Jef Pauwels, Armin Tavakoli, Erik Woodhead, and Stefano Pironio. Entanglement in prepare-and-measure scenarios: many questions, a few answers. *New Journal of Physics*, 24(6):063015, June 2022.

[96] Péter E. Frenkel and Mihály Weiner. On entanglement assistance to a noiseless classical channel. *Quantum*, 6:662, March 2022.

[97] Antonio Acín, Nicolas Brunner, Nicolas Gisin, Serge Massar, Stefano Pironio, and Valerio Scarani. Device-Independent Security of Quantum Cryptography against Collective Attacks. *Physical Review Letters*, 98(23):230501, June 2007.

[98] Umesh Vazirani and Thomas Vidick. Fully device-independent quantum key distribution. *Physical Review Letters*, 113(14):140501, September 2014.

[99] Ivan Šupić and Joseph Bowles. Self-testing of quantum systems: a review. *Quantum*, 4:337, September 2020.

[100] Dave Bacon and Ben F. Toner. Bell Inequalities with Auxiliary Communication. *Physical Review Letters*, 90(15):157904, April 2003.

[101] Katherine Maxwell and Eric Chitambar. Bell inequalities with communication assistance. *Physical Review A*, 89(4):042108, April 2014.

[102] Emmanuel Zambrini Cruzeiro and Nicolas Gisin. Bell Inequalities with One Bit of Communication. *Entropy*, 21(2):171, February 2019.

[103] Martin J. Renner, Armin Tavakoli, and Marco Túlio Quintino. Classical Cost of Transmitting a Qubit. *Physical Review Letters*, 130(12):120801, March 2023.

[104] Nicolas Brunner, Nicolas Gisin, and Valerio Scarani. Entanglement and non-locality are different resources. *New Journal of Physics*, 7:88, April 2005.

[105] Nicolas J. Cerf, Nicolas Gisin, Serge Massar, and Sandu Popescu. Simulating maximal quantum entanglement without communication. *Physical Review Letters*, 94(22):220403, June 2005.

[106] Philippe H. Eberhard. Background level and counter efficiencies required for a loophole-free einstein-podolsky-rosen experiment. *Physical Review A*, 47(2):R747, February 1993.

[107] Adán Cabello and Jan-Åke Larsson. Minimum Detection Efficiency for a Loophole-Free Atom-Photon Bell Experiment. *Physical Review Letters*, 98(22):220402, June 2007.

[108] Nicolas Brunner, Nicolas Gisin, Valerio Scarani, and Christoph Simon. Detection Loophole in Asymmetric Bell Experiments. *Physical Review Letters*, 98(22):220403, June 2007.

[109] Mateus Araújo, Marco Túlio Quintino, Daniel Cavalcanti, Marcelo França Santos, Adán Cabello, and Marcelo Terra Cunha. Tests of Bell inequality with arbitrarily low photodetection efficiency and homodyne measurements. *Physical Review A*, 86(3):030101, September 2012.

[110] Simon Kochen and Ernst P. Specker. The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics*, 17(1):59–87, July 1967.

[111] Samuel Portmann, Cyril Branciard, and Nicolas Gisin. Local content of all pure two-qubit states. *Physical Review A*, 86(1):012104, July 2012.

[112] Peter Sidajaya, Aloysius Dewen Lim, Baichu Yu, and Valerio Scarani. Neural Network Approach to the Simulation of Entangled States with One Bit of Communication. *Quantum*, 7:1150, October 2023.

[113] Nicolas Gisin and Florian Fröwis. From quantum foundations to applications and back. *Philosophical Transactions of the Royal Society of London Series A*, 376(2123):20170326, July 2018.

[114] Valerio Scarani. Local and nonlocal content of bipartite qubit and qutrit correlations. *Physical Review A*, 77(4):042112, April 2008.

[115] Cyril Branciard, Nicolas Gisin, and Valerio Scarani. Local content of bipartite qubit correlations. *Physical Review A*, 81(2):022103, February 2010.

[116] Arthur Fine. Hidden variables, joint probability, and the bell inequalities. *Physical Review Letters*, 48:291–295, February 1982.

[117] Arthur Fine. Joint distributions, quantum correlations, and commuting observables. *Journal of Mathematical Physics*, 23(7):1306–1310, July 1982.

[118] Michael M. Wolf, David Perez-Garcia, and Carlos Fernandez. Measurements Incompatible in Quantum Theory Cannot Be Measured Jointly in Any Other No-Signaling Theory. *Physical Review Letters*, 103(23):230402, December 2009.

[119] Claudio Carmeli, Teiko Heinosaari, and Alessandro Toigo. Quantum random access codes and incompatibility of measurements. *Europhysics Letters*, 130(5):50001, June 2020.

[120] Debashis Saha, Debarshi Das, Arun Kumar Das, Bihalan Bhattacharya, and A. S. Majumdar. Measurement incompatibility and quantum advantage in communication. *Physical Review A*, 107(6):062210, June 2023.

[121] Claudio Carmeli, Teiko Heinosaari, and Alessandro Toigo. Quantum Incompatibility Witnesses. *Physical Review Letters*, 122(13):130402, April 2019.

[122] Roope Uola, Tristan Kraft, Jiangwei Shang, Xiao-Dong Yu, and Otfried Gühne. Quantifying Quantum Resources with Conic Programming. *Physical Review Letters*, 122(13):130404, April 2019.

[123] Paul Skrzypczyk, Ivan Šupić, and Daniel Cavalcanti. All Sets of Incompatible Measurements give an Advantage in Quantum State Discrimination. *Physical Review Letters*, 122(13):130403, April 2019.

[124] Teiko Heinosaari, Takayuki Miyadera, and Mário Ziman. An invitation to quantum incompatibility. *Journal of Physics A: Mathematical and Theoretical*, 49(12):123001, February 2016.

[125] Otfried Gühne, Erkka Haapasalo, Tristan Kraft, Juha-Pekka Pellonpää, and Roope Uola. Colloquium: Incompatible measurements in quantum information science. *Reviews of Modern Physics*, 95(1):011003, February 2023.

[126] Qiongyi He and Margaret D. Reid. Genuine Multipartite Einstein-Podolsky-Rosen Steering. *Physical Review Letters*, 111(25):250403, December 2013.

[127] Joseph Bowles, Tamás Vértesi, Marco Túlio Quintino, and Nicolas Brunner. One-way Einstein-Podolsky-Rosen Steering. *Physical Review Letters*, 112(20):200402, May 2014.

[128] Daniel Cavalcanti and Paul Skrzypczyk. Quantum steering: a review with focus on semidefinite programming. *Reports on Progress in Physics*, 80(2):024001, February 2017.

[129] Pavel Sekatski, Florian Giraud, Roope Uola, and Nicolas Brunner. Unlimited One-Way Steering. *Physical Review Letters*, 131(11):110201, September 2023.

[130] Marco Túlio Quintino, Tamás Vértesi, and Nicolas Brunner. Joint Measurability, Einstein-Podolsky-Rosen Steering, and Bell Nonlocality. *Physical Review Letters*, 113(16):160402, October 2014.

[131] Roope Uola, Tobias Moroder, and Otfried Gühne. Joint Measurability of Generalized Measurements Implies Classicality. *Physical Review Letters*, 113(16):160403, October 2014.

[132] Roope Uola, Costantino Budroni, Otfried Gühne, and Juha-Pekka Pellonpää. One-to-One Mapping between Steering and Joint Measurability Problems. *Physical Review Letters*, 115(23):230402, December 2015.

[133] Wikipedia. https://en.wikipedia.org/wiki/sic-povm, 2022. (accessed 21/09/2023).

[134] Tamas Hausel, Endre Makai Jr., and Andras Szűcs. Inscribing cubes and covering by rhombic dodecahedra via equivariant topology. *Mathematika*, 47(1-2):371–397, December 2000.

[135] Freeman J. Dyson. Continuous functions defined on spheres. *Annals of Mathematics*, 54(3):534–536, November 1951.

[136] George R. Livesay. On a Theorem of F. J. Dyson. *Annals of Mathematics*, 59(2):227–229, March 1954.

[137] Edwin E. Floyd. Real-valued mappings of spheres. *Proceedings of the American Mathematical Society*, 6:957–959, June 1955.

[138] Karol Borsuk. Drei Sätze über die n-dimensionale euklidische Sphäre. *Fundamenta Mathematicae*, 20(1):177–190, 1933.

[139] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, October 1969.

[140] Tamas Vértesi. More efficient Bell inequalities for Werner states. *Physical Review A*, 78(3):032112, September 2008.

[141] Antonio Acín, Nicolas Gisin, and Benjamin Toner. Grothendieck's constant and local models for noisy entangled quantum states. *Physical Review A*, 73(6):062105, June 2006.

[142] Sébastien Designolle, Gabriele Iommazzo, Mathieu Besançon, Sebastian Knebel, Patrick Gelß, and Sebastian Pokutta. Improved local models and new Bell inequalities via Frank-Wolfe algorithms. *Physical Review Research*, 5(4):043059, October 2023.

[143] Flavien Hirsch, Marco Túlio Quintino, Tamás Vértesi, Miguel Navascués, and Nicolas Brunner. Better local hidden variable models for two-qubit Werner states and an upper bound on the Grothendieck constant $K_G(3)$. *Quantum*, 1:3, April 2017.

[144] Michał Oszmaniec, Leonardo Guerini, Peter Wittek, and Antonio Acín. Simulating Positive-Operator-Valued Measures with Projective Measurements. *Physical Review Letters*, 119(19):190501, November 2017.

[145] Marco Túlio Quintino, Tamás Vértesi, Daniel Cavalcanti, Remigiusz Augusiak, Maciej Demianowicz, Antonio Acín, and Nicolas Brunner. Inequivalence of entanglement, steering, and Bell nonlocality for general measurements. *Physical Review A*, 92(3):032107, September 2015.

[146] Jessica Bavaresco, Marco Túlio Quintino, Leonardo Guerini, Thiago O. Maciel, Daniel Cavalcanti, and Marcelo Terra Cunha. Most incompatible measurements for robust steering tests. *Physical Review A*, 96(2):022110, August 2017.

[147] H. Chau Nguyen, Antony Milne, Thanh Vu, and Sania Jevtic. Quantum steering with positive operator valued measures. *Journal of Physics A: Mathematical and Theoretical*, 51(35):355302, August 2018.

[148] H. Chau Nguyen, Huy-Viet Nguyen, and Otfried Gühne. Geometry of Einstein-Podolsky-Rosen Correlations. *Physical Review Letters*, 122(24):240401, June 2019.

[149] Open quantum problems, iqoqi vienna: Steering bound for qubits and povms, 2023. (accessed 12/07/2023).

[150] Reinhard F. Werner. Steering, or maybe why einstein did not go all the way to bells argument. *Journal of Physics A: Mathematical and Theoretical*, 47(42):424008, October 2014.

[151] H. Chau Nguyen and Otfried Gühne. Some quantum measurements with three outcomes can reveal nonclassicality where all two-outcome measurements fail to do so. *Physical Review Letters*, 125(23):230402, December 2020.

[152] Mafalda L. Almeida, Stefano Pironio, Jonathan Barrett, Géza Tóth, and Antonio Acín. Noise Robustness of the Nonlocality of Entangled Quantum States. *Physical Review Letters*, 99(4):040403, July 2007.

[153] Yujie Zhang and Eric Chitambar. Exact Steering Bound for Two-Qubit Werner States. *Physical Review Letters*, 132(25):250201, June 2024.

[154] Martin J. Renner. https://github.com/martinjrenner, 2024.

[155] Magdalena Zych, Fabio Costa, Igor Pikovski, and Časlav Brukner. Bell's theorem for temporal order. *Nature Communications*, 10:3772, August 2019.

[156] Philippe Allard Guérin, Adrien Feix, Mateus Araújo, and Časlav Brukner. Exponential communication complexity advantage from quantum superposition of the direction of communication. *Physical Review Letters*, 117(10):100502, September 2016.

[157] Jessica Bavaresco, Mio Murao, and Marco Túlio Quintino. Strict hierarchy between parallel, sequential, and indefinite-causal-order strategies for channel discrimination. *Physical Review Letters*, 127(20):200504, November 2021.

[158] Daniel Ebler, Sina Salek, and Giulio Chiribella. Enhanced communication with the assistance of indefinite causal order. *Physical Review Letters*, 120(12):120502, March 2018.

[159] Sina Salek, Daniel Ebler, and Giulio Chiribella. Quantum communication in a superposition of causal orders. *arXiv:1809.06655*, September 2018.

[160] Giulio Chiribella, Manik Banik, Some Sankar Bhattacharya, Tamal Guha, Mir Alimuddin, Arup Roy, Sutapa Saha, Sristy Agrawal, and Guruprasad Kar. Indefinite causal order enables perfect quantum communication with zero capacity channels. *New Journal of Physics*, 23(3):033039, March 2021.

[161] Yu Guo, Xiao-Min Hu, Zhi-Bo Hou, Huan Cao, Jin-Ming Cui, Bi-Heng Liu, Yun-Feng Huang, Chuan-Feng Li, Guang-Can Guo, and Giulio Chiribella. Experimental transmission of quantum information using a superposition of causal orders. *Physical Review Letters*, 124(3):030502, January 2020.

[162] Kaumudibikash Goswami, Y. Cao, Gerardo A. Paz-Silva, Jacquiline Romero, and Andrew G. White. Increasing communication capacity via superposition of order. *Physical Review Research*, 2(3):033292, August 2020.

[163] Alastair A. Abbott, Julian Wechs, Dominic Horsman, Mehdi Mhalla, and Cyril Branciard. Communication through coherent control of quantum channels. *Quantum*, 4:333, September 2020.

[164] Philippe Allard Guérin, Giulia Rubino, and Časlav Brukner. Communication through quantum-controlled noise. *Physical Review A*, 99(6):062317, June 2019.

[165] Giulia Rubino, Lee A. Rozema, Daniel Ebler, Hlér Kristjánsson, Sina Salek, Philippe Allard Guérin, Alastair A. Abbott, Cyril Branciard, Časlav Brukner, Giulio Chiribella, and Philip Walther. Experimental quantum communication enhancement by superposing trajectories. *Physical Review Research*, 3(1):013093, January 2021.

[166] Mateus Araújo, Philippe Allard Guérin, and Ämin Baumeler. Quantum computation with indefinite causal structures. *Physical Review A*, 96(5):052315, November 2017.

[167] Ämin Baumeler and Stefan Wolf. Computational tameness of classical non-causal models. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170698, January 2018.

[168] Lorenzo M. Procopio, Amir Moqanaki, Mateus Araújo, Fabio Costa, Irati Alonso Calafell, Emma G. Dowd, Deny R. Hamel, Lee A. Rozema, Časlav Brukner, and Philip Walther. Experimental superposition of orders of quantum gates. *Nature Communications*, 6:7913, August 2015.

[169] Giulia Rubino, Lee A. Rozema, Adrien Feix, Mateus Araújo, Jonas M. Zeuner, Lorenzo M. Procopio, Časlav Brukner, and Philip Walther. Experimental verification of an indefinite causal order. *Science Advances*, 3(3):e1602589, March 2017.

[170] Giulia Rubino, Lee A. Rozema, Francesco Massa, Mateus Araújo, Magdalena Zych, Časlav Brukner, and Philip Walther. Experimental entanglement of temporal order. *Quantum*, 6:621, January 2022.

[171] Kaumudibikash Goswami, Christina Giarmatzi, Michael Kewming, Fabio Costa, Cyril Branciard, Jacquiline Romero, and Andrew G. White. Indefinite causal order in a quantum switch. *Physical Review Letters*, 121(9):090503, August 2018.

[172] Kejin Wei, Nora Tischler, Si-Ran Zhao, Yu-Huai Li, Juan Miguel Arrazola, Yang Liu, Weijun Zhang, Hao Li, Lixing You, Zhen Wang, Yu-Ao Chen, Barry C. Sanders, Qiang Zhang, Geoff J. Pryde, Feihu Xu, and Jian-Wei Pan. Experimental quantum switching for exponentially superior quantum communication complexity. *Physical Review Letters*, 122(12):120504, March 2019.

[173] Timoteo Colnaghi, Giacomo Mauro D'Ariano, Stefano Facchini, and Paolo Perinotti. Quantum computation with programmable connections between gates. *Physics Letters A*, 376(45):2940–2943, October 2012.

[174] Stefano Facchini and Simon Perdrix. Quantum circuits for the unitary permutation problem. *Theory and Applications of Models of Computation*, page 324–331, April 2015.

[175] Daniel J. Kleitman and David J. Kwiatkowski. A lower bound on the length of a sequence containing all permutations as subsequences. *Journal of Combinatorial Theory, Series A*, 21(2):129 – 136, September 1976.

[176] Malcolm Newey. Notes on a problem involving permutations as subsequences. *Technical Report 340, Stanford University*, March 1973.

[177] Eugen Zalinescu. Shorter strings containing all k-element permutations. *Information Processing Letters*, 111:605–608, June 2011.

[178] Sasa Radomirovic. A construction of short sequences containing all permutations of a set as subsequences. *The Electronic Journal of Combinatorics*, 19:P31, November 2012.

[179] Mateus Araújo, Adrien Feix, Fabio Costa, and Časlav Brukner. Quantum circuits cannot control unknown operations. *New Journal of Physics*, 16(9):093026, September 2014.

[180] Benjamin P. Lanyon, Marco Barbieri, Marcelo P. Almeida, Thomas Jennewein, Timothy C. Ralph, Kevin J. Resch, Geoff J. Pryde, Jeremy L. O'Brien, Alexei Gilchrist, and Andrew G. White. Simplifying quantum logic using higher-dimensional hilbert spaces. *Nature Physics*, 5(2):134–140, February 2009.

[181] Xiao-Qi Zhou, Timothy C. Ralph, Pruet Kalasuwan, Mian Zhang, Alberto Peruzzo, Benjamin P. Lanyon, and Jeremy L. O'Brien. Adding control to arbitrary unknown quantum operations. *Nature Communications*, 2:413, August 2011.

[182] Xiao-Qi Zhou, Pruet Kalasuwan, Timothy C. Ralph, and Jeremy L. O'Brien. Calculating unknown eigenvalues with a quantum algorithm. *Nature Photonics*, 7(3):223–228, March 2013.

[183] Bryan Eastin and Steven T. Flammia. Q-circuit tutorial. *arXiv:quant-ph/0406003*, August 2004.

[184] Martin J. Renner and Časlav Brukner. Reassessing the computational advantage of quantum-controlled ordering of gates. *Physical Review Research*, 3(4):043012, October 2021.

[185] Kathy J. Horadam. Hadamard matrices and their applications. *Princeton University Press*, 2007.

[186] Nicolai Friis, Antony R. Lee, and Jorma Louko. Scalar, spinor, and photon fields under relativistic cavity motion. *Physical Review D*, 88(6):064028, September 2013.

[187] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400:97–117, July 1985.

[188] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 439:553–558, October 1992.

[189] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[190] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, October 1997.