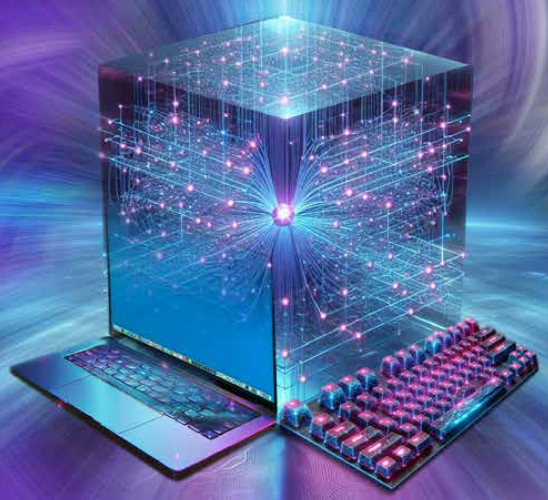
Adaptive Quantum Computers: decoding and state preparation - Niels M. P. Neumann

ISBN: 978-90-824947-8-5

Adaptive Quantum Computers

decoding and state preparation



Niels M. P. Neumann

Adaptive Quantum Computers
decoding and state preparation

ILLC Dissertation Series DS-2025-05

Institute for Logic, Language and Computation

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

The research for and publication of this doctoral thesis received financial assistance from the Netherlands Organisation for Applied Scientific Research (TNO).

Cover design by Marc Hoeijmans.
ISBN: 978-90-824947-8-5

Adaptive Quantum Computers
decoding and state preparation

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op vrijdag 2 mei 2025, te 13.00 uur

door Niels Martinus Philippe Neumann
geboren te Tilburg

Promotiecommissie

Promotores:	prof. dr. H.M. Buhrman	Universiteit van Amsterdam
	prof. dr. F. Phillipson	TNO
Copromotores:	dr. J. Briët	Centrum Wiskunde & Informatica
Overige leden:	prof. dr. C.J.M. Schoutens	Universiteit van Amsterdam
	dr. J.M.M. van de Wetering	Universiteit van Amsterdam
	prof. dr. G. Schaefer	Universiteit van Amsterdam
	prof. dr. S.M. Girvin	Yale University
	dr. F. Speelman	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

Part One: Decoding

Part Two: LAQCC

Chapter 1

Introduction

This chapter describes the emerging technology of quantum computers, by relating them to more general computational devices and problems. We also discuss advances in quantum algorithms and the search for a (provable) quantum advantage. Afterwards, the special role Fourier analysis plays in many quantum algorithms is discussed. Next, we consider some practicalities that arise when implementing quantum algorithms on quantum hardware. This chapter concludes with the contributions of this work.

1.1 Computational devices

For centuries, humans have used tools to perform computations. For a long time, we only had relatively simple tools: Our hands for counting or an abacus to perform arithmetic operations. With the advent of computers, we gained access to a new tool. These computers could carry out similar operations as we have long performed, but with the benefit of doing so automatically.

In the 19th century, Babbage was the first to propose a steam-engine powered computer-like device [Bab82]. Others later also proposed and even built devices to perform computations. It was not until the 1930s that Turing laid the theoretical foundation for a universal computer [Tur37]. Von Neumann later introduced the von Neumann architecture, where the data and the programs are both stored in the same memory [Neu45]. This architecture enables users to easily reprogram the computers and use them for different applications.

Modern computers still heavily rely on the foundational work of Turing and von Neumann. Today, computers work by manipulating computational units called

bits in a precise order. The speed at which bits are manipulated largely determines a computer's computational power. Modern day computers use transistors to manipulate bits. The number of transistors roughly translates to the computational power of the whole device. Moore first noted an apparent exponential growth in the number of transistors on a chip, which he then predicted as a trend for future growth [Moo65].

Moore's law, as this predicted growth was quickly called, effectively states that the number of transistors per integrated circuit roughly doubles every two years. As a result, the size of the transistors decreases by roughly the same factor. This size decrease implies a limit to the scaling of the computational power resulting from the number of transistor per chip [Spe18]. At some point, the transistors become so small that they no longer obey the laws of conventional physics. Instead, quantum physics must be used to describe their behavior [Lan12].

Quantum physics extends the laws of conventional physics as developed by Newton and others [New78]. With quantum physics, events can happen with certain probability, instead of with certainty. As an undesirable effect, electrons in transistors can 'tunnel' through barriers, thereby resulting in unpredictable behavior of the transistors.

1.2 Computational problems

As computers can be programmed to automatically perform operations and thus solve problems, it is natural to ask what problems these computers can solve. The field of computer science tries to group problems based on the asymptotic computational resources required to solve them. We call these groups complexity classes.

The most well-known class is P, the class of decision problems solvable in polynomial time[1]. We call problems in P efficiently solvable and we call algorithms that solve a problem in P efficient. For an algorithm f to be of polynomial time means that there exist positive constants A and c such that on binary input x of length n, the algorithm computes $f(x)$ in at most An^c steps.

As both A and c can be any nonzero positive constant, the total number of operations required can vary widely. It might therefore seem counterintuitive to call *all* polynomial-time algorithms efficient. The practical applicability of algorithms significantly depends on the exact value of the constant. Luckily, the constants for most polynomial-time algorithms used in practice are small [Wig19, Section 3].

[1]Formally, the class P is defined in terms of languages that decide a decision problem in polynomial time. We will instead follow the definition in terms of polynomial-time algorithms.

Polynomial-time algorithms have the added benefit that their sum, product, and composition remain polynomial. This property is desired as it allows an efficient algorithm to call another efficient algorithm as subroutine. Their composition remains efficient. Subroutines are extensively used in programming languages.

The best known algorithms for problems without known efficient solutions typically have running times that are exponential in the input length. As exponential functions grow significantly faster than polynomials, even for small input sizes, the running times of these inefficient algorithms become too large.

Examples of problems that admit efficient solutions are finding the shortest route between different stops [Dij59] and testing the primality of integers [AKS04]. Yet, we have not yet found efficient algorithms for the closely related problems of finding the shortest cycle between different stops (the Traveling Salesman Problem) [Rob49] and finding the prime factors of an integer. Many believe no efficient algorithm exists for these problems. However, these two problems have efficiently verifiable solutions; Computing the product of different primes is simple, as is comparing it with the original integer. This gives the complexity class NP, consisting of problems that can be verified efficiently (that is, in polynomial time).

Naturally, every problem in P is also in NP, as finding a solution is more difficult than verifying one. However, it is still unknown whether there are any problems in NP that are not in P. Many believe this to be the case, but it remains one of the main open questions in the field.

As one might expect, there are problems that admit no efficiently verifiable solution and are outside of NP. A prime example is the Halting problem, which tries to decide if a computer program terminates or run indefinitely. Turing proved that this problem is in fact undecidable [Tur37], meaning that no single algorithm can decide for every program if it will ever terminate. Aaronson, Kuperberg, and Granade provide an extensive overview of many complexity classes, including known inclusions and separations between certain classes, as well as well-known extensions [AKG05].

If we have a complexity class, we can often associate it with a class of circuits. Figure 1.1 gives a graphical example of a circuit. The input of length n is shown at the bottom, and time flows from the bottom to the top. Each square represents an elementary operation, often referred to as a gate. We call the number of inputs to a gate the fan-in and the number of outputs the fan-out of a gate. Examples of these operations include the AND-gate, which outputs 1 precisely if all inputs are 1, and the OR-gate, which outputs 1 precisely if at least one of the inputs is 1. We can group the gates in layers, such that every gate in a layer is independent of the other gates in the layer. In the figure, the dotted box denotes a layer. The width of a single layer equals the number of operations in that layer and the width of the circuit as a whole equals the maximum width across all layers. The

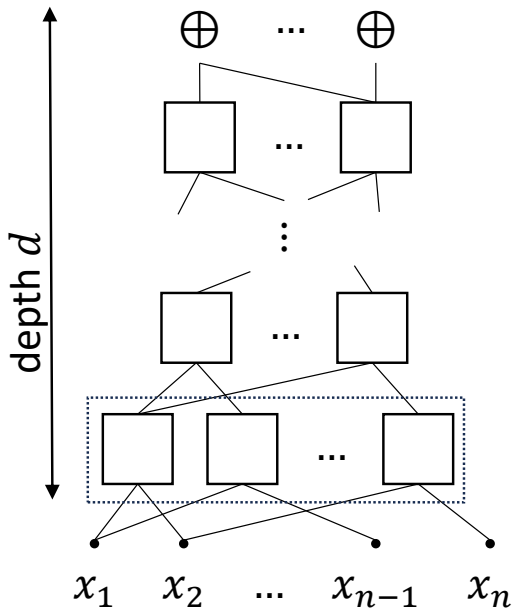


Figure 1.1: An example of a circuit. The n inputs are shown at the bottom and every square box represents an elementary operation. Time flows from the bottom to the top. The dotted box denotes a layer and the number of numbers equals the depth of the circuit.

depth of a circuit is defined as the minimum number of layers required.

In terms of circuits, the class P consists of circuits where both the width and depth of the circuit are polynomial in n. In the remainder of this work we will mainly focus on circuits. For a complexity class X, we refer to the corresponding circuits as X-circuits. In the remainder of this work, we will interpret the complexity classes in terms of these circuits. Differences between complexity classes translate into differences in the allowed gates, the fan-in or fan-out of gates, or the depth or width of the circuits.

An example of a complexity class we will revisit in this work is the class NC^k. This class consists of all circuits of polynomial size in the input length n and depth $\mathcal{O}((\log n)^k)$ using only bounded-fan-in AND- and OR-gates[2]. The closely related class AC^k consists of circuits of the same depth and size, but uses unbounded-fan-in AND- and OR-gates. The class TC^k consists of circuits that additionally have access to unbounded-fan-in Threshold$_t$-gates. These gates evaluate to 1 precisely if the inputs sum to at least t. These three classes admit a natural hierarchy:

$$\mathsf{NC}^k \subseteq \mathsf{AC}^k \subseteq \mathsf{TC}^k \subseteq \mathsf{NC}^{k+1},$$

[2]We refer to the List of Symbols for a definition of the $\mathcal{O}$-, o-, Ω-, ω-, and Θ-notation. We can add a subscript to these notations to indicate a dependency on some fixed constant.

where it is unknown if all inclusions are strict, or whether for some k, the inclusion is actually an equality. The first two inclusions follow directly from the definition. The third inclusion follows as any gate with unbounded-fan-in can be implemented with bounded-fan-in gates in logarithmic depth.

Another class that uses bounded-fan-in AND- and OR-gates is L. This class consists of all circuits of width $\mathcal{O}(\log n)$ and has no restrictions on the circuit depth. Johnson showed the inclusion $\mathsf{L} \subseteq \mathsf{AC}^1 \subseteq \mathsf{TC}^1$ [Joh90].

1.3 From two problems to a single solution

We have now encountered two problems: 1) The size of transistors presents a natural barrier beyond which quantum mechanical laws have to be taken into account; 2) For some problems we have no efficient algorithms. Where some see a problem, others see a new possibility of using quantum mechanical effects to perform computations.

Inspired by work of others [Ben73; Ben80; Ben82], Feynman popularized the idea of a programmable quantum device to simulate physical systems [Fey82]. Instead of fighting against unwanted quantum effects in transistors, quantum computers embrace the quantum effects and use them to perform computations. Shortly after, Deutsch proposed a model for a universal quantum computer [Deu85], in line with the work for conventional computers by Turing.

1.3.1 Quantum states

Quantum computers use the properties of superposition, measurement, entanglement, and interference to perform computations on quantum bits, often called qubits. In the following sections, we briefly discuss these properties.

Qubits and superposition

Quantum computers and conventional computers are closely related. Conventional computers operate on bits, each with 2 (orthogonal) computational basis states, which we will denote by $|0\rangle$ and $|1\rangle$. With n bits in total, a conventional computer is in one of the 2^n possible computational basis states.

Qubits are the quantum-mechanical analogue of the conventional bit. A *qubit* $|\psi\rangle$ in a quantum computer additionally has the property that it can be in any *superposition* of the two computational basis states

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle .$$

In some sense, a superposition means that a qubit is in all computational basis states simultaneously. More rigorously, a superposition is a complex linear com-

bination of the computational basis states, such that α_i is the complex amplitude of the state $|i\rangle$ and $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

The computational basis states of a qubit form a basis for the Hilbert space $\mathcal{H} = \mathbb{C}^2$. A Hilbert space is a vector space with an inner product, such that the resulting metric space is complete. We will use the operator norm given by

$$\|U\|_{op} = \inf\{c > 0 \mid \|U|\psi\rangle\|_2 \leq c\||\psi\rangle\|_2, \forall |\psi\rangle \in \mathcal{H}\}, \tag{1.1}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and gives the length of a vector.

Measurements

Observing a quantum state, even those that are in superposition, will return precisely one computational basis state. Before measuring, it is unknown which computational basis state is found. All we can say is that we observe state $|i\rangle$ with probability $|\alpha_i|^2$.

Measurements also alter the measured quantum state: Only the measured state remains, whereas the rest of the superposition is destroyed.

Entanglement

The composition of two qubits is given by the tensor product: Given two qubits $|\psi\rangle_A \in \mathcal{H}_A$ and $|\phi\rangle_B \in \mathcal{H}_B$, their composition is given by

$$|\psi\rangle \otimes |\phi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B.$$

The composed Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ has as basis all possible combinations of the computational basis states of $\mathcal{H}_A$ and $\mathcal{H}_B$.

If we take the tensor product of n qubits, we can write the joint state as a tensor product of these n single qubits:

$$\sum_{i_1,\ldots,i_n \in \{0,1\}} \alpha_{i_1 \ldots i_n} |i_1\rangle \otimes \ldots \otimes |i_n\rangle .$$

We can equivalently interpret the n qubits as a 2^n-level system, described by

$$\sum_{i=0}^{2^n-1} \alpha_i |i\rangle .$$

Both notations are used interchangeably in quantum computer literature. The used representation usually follows from the context.

The composed Hilbert space contains more states than just the product between the states of the two individual Hilbert spaces. We call a quantum state in

$|\varphi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ *separable* if there exist $|\psi\rangle_A \in \mathcal{H}_A$ and $|\phi\rangle_B \in \mathcal{H}_B$ such that $|\varphi\rangle = |\psi\rangle_A \otimes |\phi\rangle_B$. In all other cases, we call $|\varphi\rangle$ *entangled*. Most quantum states in the composed Hilbert space are entangled.

A prime example of an entangled state between two parties A and B is

$$\frac{1}{\sqrt{2}} \left(|0\rangle_A |0\rangle_B + |1\rangle_A |1\rangle_B \right), \tag{1.2}$$

often called the EPR-state after Einstein, Podolsky, and Rosen who first analyzed the state and the correlations between the measurement results [EPR35]. Measuring the qubit of A locally directly determines the value of the other qubit, independent of the physical distance between the two qubits.

Bell showed that the locally measured entangled variables are strictly more correlated than locally measured variables based on shared randomness. Clauser et al. introduced the CHSH game, a specific two-player game, where the players either share randomness or share an EPR-state. With shared randomness, the players win the game with probability at most 0.75, whereas they do so with probability $\cos^2(\pi/8) \approx 0.85$ if they share an EPR-state [Cla+69]. This so-called Bell inequality violation was later also realized experimentally [Bel64; Hen+15].

Interference

Contrary to what one might believe, quantum computing is more than simply performing computations using probability distributions. The complex amplitudes α_i can be negative. Different quantum states can therefore cancel or reinforce each other, an effect called *interference*. One way to let quantum states interfere is via quantum gates.

1.3.2 Quantum gates

Quantum gates can manipulate quantum states. Quantum gates are unitary operators $U : \mathcal{H} \to \mathcal{H}$ mapping $|\phi\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle \mapsto |\psi\rangle = \sum_{i=0}^{N-1} \alpha_i U |i\rangle$. The unitary property of quantum gates implies that they are reversible.

In this work, we will only consider quantum equivalents of the conventional circuits shown in Figure 1.1. Quantum devices that perform operations represented by such circuits are often called gate-based quantum computers. Linear algebra provides the mathematical foundation that describes gate-based quantum computing. In this context, we can interpret quantum states as unit vectors, and quantum gates as unitary matrices. Note that unitary matrices preserve the length of vectors as desired.

Quantum circuits are often depicted with time running from left to right, as shown in Figure 1.2. This circuit prepares the EPR-state shown in Equation (1.2). Each

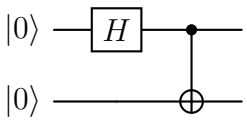


Figure 1.2: An example quantum circuit. Each line represents a qubit. The shown quantum circuit prepares the quantum state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

line represents a qubit, the square denotes the single-qubit Hadamard gate, and the black circle connected to the $\oplus$-symbol represents the CNOT-gate (short for controlled-NOT-gate). The black circle represents a conditional control: The NOT-operation on the second qubit is only applied if the controlling qubit is in the $|1\rangle$-state. For computational basis states $|x\rangle$ and $|y\rangle$, the Hadamard gate H implements the map $|x\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x |1\rangle)$ and the CNOT-gate implements the map $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus x\rangle$, where $\oplus$ denotes addition modulo 2.

Other common quantum gates include the three *Pauli matrices*,

$$X : |x\rangle \mapsto |1 \oplus x\rangle , \quad Y : |x\rangle \mapsto -i(-1)^x |1 \oplus x\rangle , \quad Z : |x\rangle \mapsto (-1)^x |x\rangle . \tag{1.3}$$

From the Pauli matrices we arrive at the Pauli group P_n consisting of all n-qubit combinations of Pauli matrices:

$$P_n = \{cp_1 \otimes \ldots \otimes p_n \mid p_i \in \{I, X, Y, Z\}, c \in \{\pm 1, \pm i\}\}, \tag{1.4}$$

where I is the identity matrix that leaves all states the same.

We also have parametrized versions of the Pauli matrices:

$$R_X(\theta) = e^{-i\theta X/2} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{1.5}$$

$$R_Y(\theta) = e^{-i\theta Y/2} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{1.6}$$

$$R_Z(\theta) := e^{i\theta/2} e^{-i\theta Z/2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}. \tag{1.7}$$

In the last line we used that global phases have no measurable effect in quantum states to simplify the expression for the R_Z-gate.

Physical realizations of quantum computers often have access to a small gate set. Gates outside this gate set have to be constructed using the available gates. However, not all gate sets are equal. Consider for instance the *Clifford group* C_n, which consists of all gates that normalize the Pauli group:

$$C_n = \{V \in U^{2^n \times 2^n} \mid VPV^\dagger \in P_n \, \forall P \in P_n\}. \tag{1.8}$$

Quantum circuits consisting solely of gates from the Clifford group can be efficiently simulated [Got98], whereas it is expected that general quantum circuits do not admit efficient simulations by conventional computers.

We call a gate set that can approximate any quantum circuit to arbitrary precision $\varepsilon > 0$ *universal*. If two quantum circuits are close with respect to the operator norm (Equation (1.1)), the quantum states they prepare are close with respect to the Euclidean norm. A common universal gate set is the set generated by the CNOT-gate, the Hadamard gate and the $R_Z(\pi/4)$-gate, often called the T-gate. By the Solovay-Kitaev theorem, the overhead of this approximation is $\mathcal{O}(\log^c(1/\varepsilon))$ for some small constant c [Kit97; DN06, Theorem 1].

1.3.3 Quantum complexity theory

Using these definitions, we can generalize the conventional complexity classes to quantum complexity classes. From NC^k we now obtain QNC^k, the class consisting of quantum circuits of depth $\mathcal{O}((\log n)^k)$ consisting of single and two-qubit gates.

The class P generalizes to BQP (for bounded-error polynomial time), the class of polynomial-sized quantum circuits that output the correct answer with probability at least 2/3. The bounded error allows for more flexibility in the algorithm design and naturally generalizes the zero-error setting of P. In this work, we deviate slightly from the class BQP and instead consider the class QPoly consisting of all polynomial-sized circuits that use single- and two-qubit quantum gates.

1.3.4 Different computational paradigms

Conventional computing knows two computational paradigms: digital computing and analog computing. Similarly, quantum computing knows different computational paradigms. For information purposes, we briefly discuss some of them.

The first is adiabatic quantum computing, which works by adiabatically evolving the quantum state using a problem-specific Hamiltonian. Adiabatic evolution requires a slow enough evolution at low temperatures. A measurement of the final state after evolution will give the answer to the problem. If the evolution is too fast or occurs at too high temperatures, an incorrect answer is found with high probability. Adiabatic quantum computing is proven to be equivalent to gate-based quantum computing [Aha+07].

The second closely related paradigm is quantum annealing. Quantum annealing is based on simulated annealing, an optimization technique that sometimes considers a worse solution to escape local optima [KN98]. Quantum annealing algorithms evolve the system fast and the system might return the wrong answer. As a result, the conditions imposed on the quantum hardware are less stringent compared to adiabatic quantum computers. To compensate the possible wrong

answers returned by the quantum annealing process, the system is evolved, measured and reinitialized multiple times. The resulting probability distribution can then be used to tackle optimization problems where often a *good* solution suffices, while the *best* solution might be hard to find or might not even exist.

Measurement-based quantum computing, a universal form of quantum computing, is the third computational paradigm [GC99; RB01; Joz06; CJL08; Bri+09]. In this paradigm, a quantum state is prepared and operations are performed by measuring qubits in different bases, depending on previous measurements and intermediate computations. This method is sometimes also called one-way quantum computing because of the measurements.

Fourth, we have photonic quantum computing, which uses photons as quantum states. Photonic quantum computers use beam splitters, mirrors, and polarizers to manipulate the photons [AC98]. Photonic systems using two-level systems exist, as well as those using multilevel systems. The latter case uses the number of photons in a state as the computational unit. Both heuristic [AA13] and universal [KLM01] versions of photonic quantum computing have been proposed.

1.4 Towards new computational devices

The actual use of these quantum effects requires physical implementations of quantum computers. Conventional computers operate under the laws of Newtonian mechanics and after years of development, manufacturing these devices is relatively well understood. Quantum computers on the other hand operate under the laws of quantum mechanics and the quantum states in a quantum computers are usually fragile and easily disturbed. As a result, manufacturing quantum computers raises new engineering challenges [Alm+17].

Even today, multiple hardware technologies are used to build quantum computers, some examples include trapped ions [CZ95], neutral atoms [Dum+02], spin systems [Ima+99] and superconducting circuits [KLO04; Hou+09; Bar+13; Kja+20]. Moreover, among the various quantum hardware technologies currently under development are postulated options such as skyrmion qubits [PP21] and topological qubits based on anyons [Nay+08]. One might even wonder if at some point one single hardware technology will become dominant, or that instead multiple hardware technologies will remain, each with their own specific applications in mind.

DiVincenzo laid out different requirements a quantum system should adhere to, independent of the underlying technology used [DiV00]. He describes requirements needed to perform computations such as the ability to initialize qubits in a known state and later read them out, and the access to a universal gate set.

One of the main challenges in manufacturing quantum computers is isolating

them from the environment to minimize the effect of noise and decoherence. Due to this difficulty, current quantum hardware still suffers greatly from the effects of noise. Preskill saw this trend and called the current devices part of *Noisy Intermediate-Scale Quantum* (NISQ) technology [Pre18].

The NISQ devices form a stepping-stone towards general Fault-Tolerant quantum computers. In these fault-tolerant devices, we assume that qubits remain coherent throughout the whole execution of the algorithm and that algorithms produce correct answers. One step towards fault-tolerant devices is reducing error rates through the use of error-correcting codes [Sho95].

Error-correcting codes impose a code on a group of qubits, such that together, the group acts as a single qubit with lower error rate. The individual qubits are often called physical qubits and the qubits used in quantum algorithms logical qubits. Simply comparing the number of qubits a quantum device has only paints a partial picture, as the error rates of these qubits and how they can interact also affect the capabilities of quantum devices in practice.

The number of physical qubits in quantum hardware has been steadily rising over the last few years, as Figure 1.3 shows. The error rates of the (physical) qubits in these devices remain high. Only recently, the first results showed improved error rates by using error-correcting codes [Goo23; Sil+24; Ach+24]. Earlier, the error rates of physical qubits were too high for error-correcting codes to be able to work.

We can also concatenate error-correcting codes to achieve an exponential suppression of the error-rates [DMN13]. Different groups of qubits are each encoded with an error-correcting code, after which the groups themselves are also encoded using a (possibly different) error-correcting code.

1.5 Developments in quantum algorithms

Shortly after Feynman called for the development of quantum computers, the first theoretical results showed the potential power of these future devices. Deutsch and later Deutsch and Jozsa presented an oracular promise problem with provable exponential speedup [Deu85; DJ92]. They consider the problem to decide if a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is constant or balanced, provided one of the two is the case. Their quantum algorithm answers the problem with certainty using a single query to f, whereas any conventional algorithm has to make an exponential number of queries to f in the worst case.

Later, Bernstein and Vazirani presented a quantum algorithm to learn the hidden string $s \in \{0,1\}^n$ in the inner-product function $f(x) = \langle x, s \rangle \bmod 2$ [BV97]. Their algorithm again provides an exponential advantage over conventional algorithms, requiring only a single query to f, instead of n.

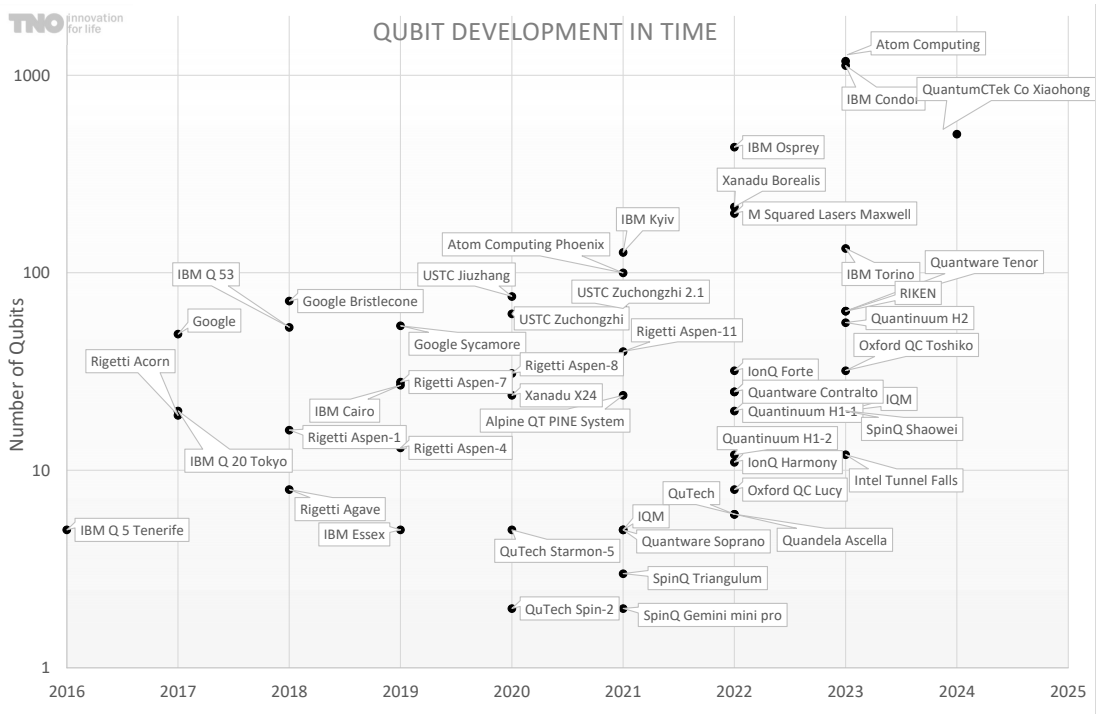


Figure 1.3: Overview of different quantum computers with the available number of qubits over the years.

Grover developed an algorithm for a practical problem: searching an unstructured database for a specific item [Gro96]. In the worst case, you have to check every item in your database before you find the correct one. Grover's algorithm, achieves the same task quadratically faster. The difference between the quantum and conventional algorithm becomes clear in the number of queries to the database. Conventional algorithms can only check a single item at a time. Quantum algorithms can query the database "in superposition", which roughly means checking the entire database at once. The successive queries are needed for constructive interference to give the correct answer with high probability.

Each of these algorithms uses an oracle, a black-box function that queries some function and returns the results in superposition (see also Section 1.6.2 and Equation (1.10)). Oracles allow to reason about the effectiveness of algorithms. In practice, we have to implement the oracle. This implementation step might diminish, or even nullify, the speedups offered by a quantum algorithm.

Shor provided another quantum algorithm for a practical problem: Breaking an often-used encryption protocol for which all known conventional algorithms take at least sub-exponential time. Shor presented a quantum algorithm to efficiently determine the period of a modular exponentiation function [Sho94; Sho97]. This computation relates to finding the prime factors of integers, which is the problem

underlying the often-used RSA-encryption protocol [RSA78].

Grover's algorithm offers a provable quadratic advantage over conventional algorithms. In contrast, Shor's algorithm provides an exponential advantage over all known conventional algorithms. As a result, the impact of Shor's algorithm is larger: It provides a polynomial-time algorithm for a problem for which no efficient conventional algorithm is known.

Note that it is still open whether the discrete logarithm problem and prime factorization, the problems solved by Shor's algorithm, are indeed hard. However, many believe this to be true.

Recently, more attention has been given to variational algorithms, where a function is optimized to solve a specific problem [FGG14; Per+14; Had+19; Cer+21]. These variational algorithms show potential for near-term applications as they can be implemented relatively easily on NISQ hardware. Variational quantum algorithms use a circuit with parametrized gates. These gate parameters are optimized with respect to some objective function.

Variational algorithms have the potential advantage that they can 'learn' the noise present in NISQ devices. Specifically, unwanted bias in the hardware can be learned as systematic noise and thus be mitigated. Noise resulting from decoherence will be biased around the expected outcome in a noiseless situation. Averaging repeated measurements therefore gives a correct answer with good probability.

Research into NISQ algorithms also triggered a new trend: optimizing costly quantum resources. The simplest approach in this research is optimizing the implementation of an algorithm with respect to the number of gates or with respect to the number of T-gates used.

These optimization objectives make sense, as applying quantum gates will introduce more noise than doing nothing. Hence, if we can implement the same operation using fewer gates, noise rates will typically decrease [KA22; Pér+23]. Similar arguments hold for optimizing the number of T-gates [Rei+17; GE21].

The second part of this work extends this line of research by optimizing with respect to the circuit depth. We present state preparation protocols that have higher success probabilities that protocols using fewer gates. See Chapter 9 for more details.

1.6 Achieving quantum advantage

The advent of NISQ devices allowed for the first time to actually implement quantum algorithms on a large scale. The next step was to show that quantum

computers can solve specific tasks faster than conventional computers in practice. This event is known as quantum advantage.

The first claim for quantum advantage was made in 2019: a quantum computer performed a sampling task in 200 seconds, whereas conventional computers would take 10,000 years to complete the task [Aru+19]. Shortly after, it was shown that conventional supercomputers could perform the task in only 2.5 days [Ped+19; PCZ22]. Still, the result was groundbreaking and opened the way to other results showing a quantum advantage [Zho+20; Mad+22; Ach+24].

However, these quantum advantage experiments do not provide a *provable* quantum advantage. The results for conventional computing are often based on extrapolation from the running times for smaller problem instances. To prove a quantum advantage we need to shift our focus to either low-depth circuits or to query complexity. The first part of this work extends this line of research in both directions.

1.6.1 Constant-depth quantum advantage

Constant-depth circuits, both quantum and conventional, recently enjoyed renewed interest in the context of provable separations. Moreover, constant-depth circuits are likely easier to implement in practice. Furthermore, constant-depth circuits are one of the few settings currently amenable to provable lower bounds. Our current techniques fail for unbounded circuit depth.

In light of provable separations, a common extension of the classes NC^0 and AC^0 is the unbounded-fan-in parity gate, which computes the parity of all input bits, denoted by $\mathsf{NC}^0[\oplus]$ and $\mathsf{AC}^0[\oplus]$, respectively. Note that this specific example is a true extension, as parity cannot be computed by AC^0 and NC^0 is a proper subset of AC^0 [AB09]. Furthermore, the classes AC^0 and $\mathsf{NC}^0[\oplus]$ are incomparable since $\mathsf{NC}^0[\oplus]$ cannot compute the n-bit AND function; indeed, $\mathsf{NC}^0[\oplus]$ circuits can compute only constant-degree polynomials over $\mathbb{F}_2$ (see also Section 3.1), whereas AND has degree n.

Moore and Moore and Nilsson were the first to consider the quantum versions of these classes [Moo99; MN01]. In contrast with their conventional analogues, the classes $\mathsf{QNC}^0[\oplus]$ and QAC^0 are known to be equivalent [Gre+02; HŠ05; Moo99]. Furthermore, the works of Moore and Green et al. showed that

$$\mathsf{QAC}^k[\oplus] = \mathsf{QAC}^k[q],$$

where $\mathsf{QAC}^k[q]$ extends QAC^k with additional modulo-q gates [Moo99; Gre+02]. For an integer $q \geq 2$, a modulo-q gate evaluates to 1 if the sum of its inputs equals 0 mod q, and evaluates to 0 otherwise. In contrast, the classes $\mathsf{AC}^0[p]$ and $\mathsf{AC}^0[q]$ are incomparable if p and q are powers of distinct primes [Raz87; Smo87].

Høyer and Špalek proved a quantum advantage, by showing that $\mathsf{QNC}^0[\oplus]$ circuits can approximate with polynomially small error functions such as OR, AND, and Majority [HŠ05]. Later, Takahashi and Tani extended these results to exact quantum circuits. Key in their work was the use of a quantum fanout gate

$$\text{Fanout} : |\mathrm{x}\rangle\,|\mathrm{y}_1\rangle \ldots |\mathrm{y}_\mathrm{n}\rangle \mapsto |\mathrm{x}\rangle\,|\mathrm{y}_1 \oplus \mathrm{x}\rangle \ldots |\mathrm{y}_\mathrm{n} \oplus \mathrm{x}\rangle\,. \tag{1.9}$$

This gate is typically not supported by the native gate set of a quantum computers. However, using intermediate conventional processing, we can still implement this gate efficiently.

For a long time, research on provable quantum advantages lay still. The breakthrough work by Bravyi, Gosset, and König restarted this line of research into provable quantum advantages leading to multiple new results and ideas:

- The 2D-Hidden Linear Function problem can be solved exactly by a QNC^0-circuit, while any AC^0-circuit succeeds with exponentially small probability under a certain input distribution [Ben+19]; this strengthened the main result of [BGK18] showing that this problem separates QNC^0 from NC^0 in the worst case.

- The Relaxed Parity Halving problem can be solved exactly by a QNC^0-circuit while any AC^0-circuit succeeds with probability at most $\frac{1}{2}+\exp(-n^\varepsilon)$ for some $\varepsilon > 0$ under the uniform input distribution [Ben+19].

- The Parallel Parity Bending problem can be solved with probability $1-o(1)$ by a $\mathsf{QNC}^0/\mathsf{qpoly}$-circuit, while any $\mathsf{AC}^0[\oplus]/\mathsf{rpoly}$-circuit succeeds with probability at most $\mathcal{O}(n^{-\varepsilon})$ [Ben+19].

- The problem of simulating correlations obtained from measuring graph states separates QNC^0 and NC^0, even in the average case [Le 19].

- The 1D Magic Square problem separates noisy QNC^0-circuits from NC^0-circuits, provided that the noise levels are below some threshold value independent of the input size [Bra+20].

Similar separations based on other relational and sampling-based problems were proven in [CSV21; GS20; BP23]. A common feature of all these problems is that they were specifically designed to prove a separation between shallow quantum and conventional circuits.

Chapters 3 and 4 extend this line of research to a problem that occurs naturally in computer science: decoding heavily corrupted error-correcting codes using limited resources. This problem is well studied in the context of conventional complexity theory, where shallow circuits endowed with parity gates are often considered.

1.6.2 Query-based quantum advantage

Another stylized model that admits provable separations is the query model. Here, black-box access is given to some function, and algorithms are compared in terms of how often the function is queried. The first exponential separations were shown already a few decades ago [Deu85; DJ92; BV97; Sim97].

Let $f : A \to \{0,1\}$, for some set A. Conventional computers can query f by inputting x and obtaining $f(x)$ from the oracle. Quantum queries are coherent, in the sense that the input can be any superposition over elements in A and the output is stored in an auxiliary bit. For any $x \in A$ and $b \in \{0,1\}$, a quantum query to f is given by the map

$$\frac{1}{\sqrt{|A|}} \sum_{x \in A} |x\rangle\,|b\rangle \mapsto \frac{1}{\sqrt{|A|}} \sum_{x \in A} |x\rangle\,|b \oplus f(x)\rangle\,. \tag{1.10}$$

We refer to this oracle as a bit-flip oracle, as it flips the state of the auxiliary bit.

If we initialize the auxiliary qubit in the $|1\rangle$-state, we can transform the oracle query into a phase query by conjugating the oracle call with Hadamard gates on the auxiliary qubit. A phase oracle thus gives the state

$$\frac{1}{\sqrt{|A|}} \sum_{x \in A} (-1)^{f(x)} |x\rangle\,|b\rangle\,.$$

Note that a bit-flip oracle generalizes a phase oracle, as in the latter, we cannot distinguish between f and the function $1 - f$, as they differ only by a global phase.

Results based on query complexity can be translated into the circuit model. This translation requires a reversible implementation of the oracle as a quantum circuit. For some problems, this implementation of the oracle might prove difficult and thereby reduce the advantage over a conventional approach. The query model remains an interesting model for proving a quantum advantage.

Aaronson showed the largest query-based separation known between quantum and conventional algorithms. He introduced a problem called Forrelation that, given two functions, asks if the first function correlates with the Fourier transform of the second [Aar10]. A quantum algorithm only needs a single query to solve the Forrelation problem. Aaronson and Ambainis showed that on input length N, a conventional query algorithm requires $\widetilde{\Omega}\big(\sqrt{N}\big)$ queries [AA15]. They also showed that this separation is optimal for a single query and conjectured it to hold for any constant number of queries. Bansal and Sinha later proved this conjecture in the standard query model [BS21]. Sherstov, Storozhenko, and Wu independently obtained a similar result for randomized queries [SSW23].

Montanaro obtained a smaller quantum advantage in the query model, but did so for a more practical problem [Mon12]. Montanaro considered learning arbitrary polynomials of degree d, a task that requires $\Omega(n^{d-1})$ quantum queries, while conventional algorithms require $\Omega(n^d)$ queries. His algorithm generalizes the Bernstein-Vazirani algorithm, which can be interpreted as learning a polynomial of degree 1. Montanaro also showed the optimality of this query lower bound in the noiseless case using Fano's inequality [FH61]. However, the algorithm by Montanaro fails if the queries can be corrupted.

Chapter 5 extends the work by Montanaro to also work in case of corrupted query calls. In Chapter 5, we revisit a result in higher-order Fourier analysis and use a quantum subroutine to improve the query complexity.

1.7 Fourier analysis: a tool for analyzing algorithms

Many quantum algorithms, most notably that of Shor, rely on an efficient implementation of the Fourier transform on quantum computers. Conventional computers can implement a Fourier transform on n elements in time polynomial in n. Coppersmith showed how quantum computers can implement a quantum Fourier transform exponentially faster in time $\operatorname{poly}\log(n)$ [Cop02].

In the remainder of this work, we only work with finite Abelian groups. Two prime examples we consider are $\mathbb{F}_p^n$, the finite n-dimensional vector space over the field with p elements for some prime p, and $\mathbb{Z}_n$, the cyclic group over the first n nonnegative integers. In this section, we present definitions only with respect to (one of) these two groups. They do naturally translate to other groups. Part One of this work considers applications of Fourier analysis and generalization higher-order Fourier analysis.

1.7.1 The Fourier transform

Fourier transforms prove vital to find patterns in data. A Fourier transform decomposes a function as a combination of character functions [Fou88]. As an example, let $f(t)$ represent some musical composition. The Fourier transform helps us to decompose the musical composition in individual notes (the characters) and their relative magnitude over time. In a sense, the Fourier transform indicates which notes are most prominent in the musical composition over time.

Now consider $\mathbb{F}_p^n$ for some prime p and let $\omega_p = e^{2\pi i/p}$. We then define its character functions as $\{\chi_y | \chi_y(x) = \omega_p^{\langle x,y\rangle}\}$, where $\langle x, y\rangle = x_1y_1 + \ldots x_ny_n$ denotes the inner product of x and y. Character functions carry the essential information of the

group elements and we often refer to them as *phase functions*. We can then define the Fourier transform in terms of the character functions:

1.7.1. Definition (Fourier transform). Let $f : \mathbb{F}_p^n \to \mathbb{C}$ be some function, then its Fourier transform $\widehat{f} : \mathbb{F}_p^n \to \mathbb{C}$ evaluated at $y \in \mathbb{F}_p^n$ is given by

$$\widehat{f}(y) = \mathbb{E}_{x \in \mathbb{F}_p^n} f(x) \omega_k^{-\langle x,y \rangle} = \frac{1}{p^n} \sum_{x \in \mathbb{F}_p^n} f(x) \omega_p^{-\langle x,y \rangle}.$$

The function f is often said to be in physical space and the Fourier transformed function $\widehat{f}$ is said to be in frequency space. Both are just naming conventions.

Both the Fourier-transformed function $\widehat{f}$ and the character functions themselves have interesting properties. For instance, every function f can be written uniquely as a sum of its Fourier coefficients, called the Fourier inversion formula:

$$f(x) = \sum_{y \in \mathbb{F}_p^n} \widehat{f}(y) \omega_p^{\langle x,y \rangle}. \tag{1.11}$$

This argument follows as the character functions form an orthonormal basis for $\mathbb{F}_p^n$. We have a total of p^n characters, one for every $y \in \mathbb{F}_p^n$, and they are all orthogonal:

$$\langle \chi_y, \chi_z \rangle = \mathbb{E}_{x \in \mathbb{F}_p^n} \omega_p^{\langle (y-z),x \rangle} = \begin{cases} 1 & \text{if } y = z \\ 0 & \text{otherwise.} \end{cases}$$

The last equation uses that the expected value over the whole group of any non-identity character function vanishes, as the next lemma shows.

1.7.2. Lemma. *Let χ_y be a non-identity character of $\mathbb{F}_p^n$. Then*

$$\mathbb{E}_{x \in \mathbb{F}_p^n} \chi_y(x) = 0. \tag{1.12}$$

Proof: As χ_y is not the identity element, there exists a $z \in \mathbb{F}_p^n$ such that $\chi_y(z) \neq 1$. Then we have

$$\begin{aligned} \mathbb{E}_{x \in \mathbb{F}_p^n} \chi_y(x) &= \mathbb{E}_{x \in \mathbb{F}_p^n} \chi_y(x+z) \\ &= \chi_y(z) \mathbb{E}_{x \in \mathbb{F}_p^n} \chi_y(x). \end{aligned}$$

In the first equality, we translated x to $x + z$, in the second equality we used the linearity of χ_y in its phase. □

This proof holds in greater generality for any non-identity character of any finite Abelian group.

An important property of Fourier transforms is that they preserve inner products, as shown by Parseval's identity from 1806 [Par06]:

1.7.3. PROPOSITION. *For any two function $f, g : \mathbb{F}_p^n \to \mathbb{C}$ it holds that*

$$\mathbb{E}_{x\in\mathbb{F}_p^n} f(x)\overline{g(x)} = \sum_{y\in\mathbb{F}_p^n} \widehat{f}(y)\overline{\widehat{g}(y)}.$$

The proof of Parseval's identity follows by applying the Fourier inversion formula twice and then use that characters are orthogonal. As Fourier transforms preserve inner products, they are unitary and thus valid quantum operations.

1.7.2 The quantum Fourier transform

Fourier analysis might seem like a complicated topic, yet it finds applications throughout quantum computing: The Bernstein-Vazirani algorithm [BV97], Simon's problem [Sim97], Shor's algorithm [Sho97], the HHL algorithm [HHL09] and many more use a quantum Fourier transformation over some group. The efficient implementation of a quantum Fourier transform by Coppersmith often gives the exponential separation between quantum and conventional circuits.

The algorithm by Shor works over the cyclic group $\mathbb{Z}_N$, with $N = 2^n$, instead of over $\mathbb{F}_2^n$. Luckily, a bijection between the two groups exist:

$$x = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_2^n \leftrightarrow y = \sum_{i=0}^{n-1} 2^i x_i \in \mathbb{Z}_N.$$

A quantum Fourier transform over $\mathbb{F}_2^n$ translates to a single layer of n Hadamard gates applied to every qubit. The quantum Fourier transform over $\mathbb{Z}_{2^n}$ looks slightly different and is given by

$$QFT : |x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{xy}{N}} |y\rangle . \tag{1.13}$$

As character functions we now have $\omega_{N,x}(y) = e^{2\pi i \frac{xy}{N}}$.

Implementing a conventional Fourier transform on N elements requires $\mathcal{O}(nN) = \mathcal{O}(n2^n)$ operations. By Coppersmith, we can implement an equivalent quantum Fourier transform using only $\mathcal{O}(n^2)$ quantum gates.

Figure 1.4 shows an implementation of a quantum Fourier transform on n qubits. The R_k-gates denote an $R_Z(\pi/2^k)$-gate. The quantum Fourier transform typically also swaps all qubits at the end of the circuit, we omitted that operation, as we can easily incorporate these operations in the rest of a quantum algorithm. The controlled-R_Z-gates have a simple decomposition into two CNOT-gates and two R_Z-gates, giving us a simple implementation of the quantum Fourier transform. As the circuit works for any computational basis state, it also works for arbitrary quantum state by linearity.

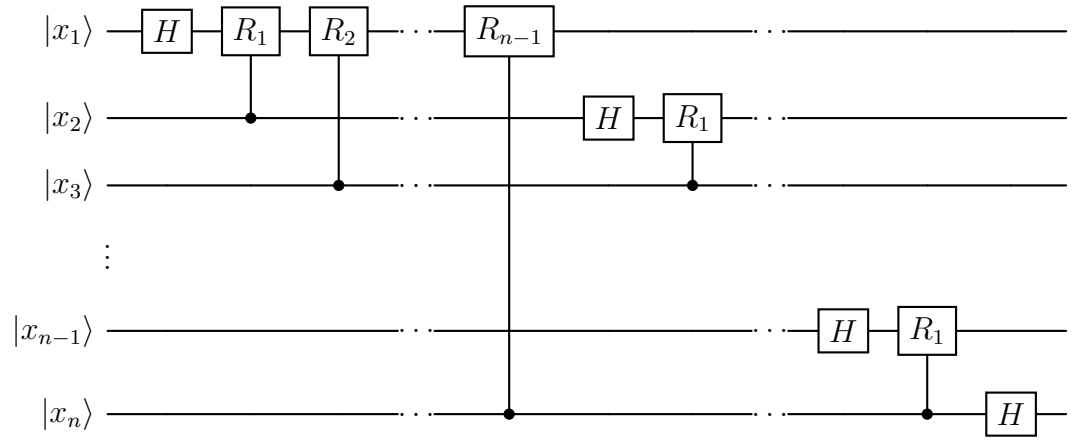


Figure 1.4: A quantum circuit that implements the quantum Fourier transform. The gates R_k correspond to $R_Z(\pi/2^k)$-gate.

We now define $(0.x_j \dots x_n)$ as the binary fraction $\sum_{i=j}^{n} x_i/2^{i-j+1}$, to rewrite the state $|x\rangle$ after the quantum Fourier transform as

$$QFT\,|x\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i(0.x_1 \dots x_n)}\,|1\rangle)(|0\rangle + e^{2\pi i(0.x_2 \dots x_n)}\,|1\rangle) \dots (|0\rangle + e^{2\pi i(0.x_n)}\,|1\rangle).$$

This representation is often used for quantum arithmetic algorithms that work with the Fourier-transformed state [RG17; NW22].

1.7.3 Higher-order Fourier analysis

Fourier analysis is an important technique for counting linear patterns in subsets of integers and other Abelian groups. For instance, in the field of additive combinatorics, Roth used Fourier analysis to show that any sufficiently large subset of positive integers contains a triple (a, b, c) such that the difference between two successive numbers is the same [Rot53]. We call such triples three-term arithmetic progressions.

Counting longer arithmetic progressions amounts to counting quadratic or higher-order patterns in subsets of integers. However, Fourier analysis appears insufficient to find correlations with polynomials of degree larger than one.

An important realization is that taking the derivative of a function lowers its degree. For a function f over a finite Abelian group, we define its multiplicative derivative in the direction h as $\Delta_h f(x) = f(x+h)\overline{f(x)}$. For a polynomial $P \in \mathbb{F}_p[x_1, \dots, x_n]$, we often use the additive derivative instead, given by $\Delta_h P(x) = P(x+h) - P(x)$.

For a long time, a generalization of Fourier analysis was unknown, until the seminal work by Gowers [Gow01]. He introduced the notion of higher-order Fourier

analysis, which examines correlations with higher-order functions. Gowers also introduced his uniformity norms based on the multiplicative derivatives of a function, which we will use extensively in Chapter 5 of this work.

1.7.4. Definition. Let $f : \mathbb{F}_p^n \to \mathbb{C}$ be any function. The Gowers U^d-norm is defined by

$$\|f\|_{U^d} = \left(\mathbb{E}_{x,h_1,\ldots,h_d \in \mathbb{F}_p^n} \Delta_{h_d} \ldots \Delta_{h_1} f(x)\right)^{1/2^d}. \tag{1.14}$$

For $d = 1$, the Gowers U^d-norm is actually a seminorm, as

$$\|f\|_{U^1} = |\mathbb{E}_{x \in \mathbb{F}_p^n} f(x)|$$

can vanish for nonzero functions, such as $f(x) = \omega_p^x$. For $d \geq 2$, the Gowers U^d-norm is actually a norm. For $d = 2$ specifically, we can use the Fourier inversion formula to see that

$$\begin{aligned} \|f\|_{U^2} &= \left(\mathbb{E}_{x,h_1,h_2 \in \mathbb{F}_p^n} f(x)\overline{f(x+h_1)f(x+h_2)}f(x+h_1+h_2)\right)^{1/4} \\ &= \left(\sum\nolimits_{a \in \mathbb{F}_p^n} \left|\widehat{f}(a)\right|^4\right)^{1/4} = \|\hat{f}\|_{L^4}. \end{aligned} \tag{1.15}$$

Hence, the Gowers U^2-norm of a function can be completely described in terms of the L^4 norm of the Fourier transform of that function.

For larger d, we have no such explicit relation. However, we can reformulate the Gowers U^d-norms inductively:

$$\|f\|_{U^{d+1}}^{2^{d+1}} = \mathbb{E}_{h \in \mathbb{F}_p^n} \|\Delta_h f\|_{U^d}^{2^d}. \tag{1.16}$$

Repeatedly applying the Cauchy-Schwarz inequality on this inductive expressions gives the *nesting property* [TV06, pp. 420]

$$\|f\|_{U^1} \leq \|f\|_{U^2} \leq \cdots, \tag{1.17}$$

a useful tool in many algorithms.

In general, Gowers norms quantify structure in complex functions by identifying correlations of these complex functions with polynomials. If a function correlates with a degree-d function, then taking $d+1$ derivatives of that function results in an approximately zero function, which in turn correlates with an approximately constant function. As a result, the Gowers U^{d+1}-norm is large for such a function.

Interestingly, a series of works by Green, Tao, and Ziegler showed that the converse also holds [GT08; TZ11; GTZ12]. These results are known as the inverse theorem for the Gowers U^d-norm. Gowers norms now play a fundamental role within higher-order Fourier analysis and find many applications in theoretical computer science.

In Chapter 5, we will restrict ourselves to specific functions called *polynomial phase functions*: $f : \mathbb{F}_p^n \to \mathbb{D}$, with $\mathbb{D}$ being the complex unit disc, such that $f = \omega_p^{P(x)}$ for P some n-variate polynomial over $\mathbb{F}_p$ such that every variable has degree at most $p-1$. The degree of a polynomial phase function is defined as the degree of the polynomial P. Given a quantum state $|x\rangle$, we define a phase-query to f as the map $|x\rangle \mapsto f(x)\,|x\rangle$.

Since the $(d+1)$-fold multiplicative derivatives of a degree-d polynomial phase f function equal 1, it follows from the nesting property that the correlation between f and any function $g : \mathbb{F}_p^n \to \mathbb{D}$ is bounded from above as

$$\left|\mathbb{E}_{x\in\mathbb{F}_p^n} g(x)\overline{f(x)}\right| \leq \|g\|_{U^{d+1}}.$$

We refer the interested reader to [HHL19] for an elaborate overview of higher-order Fourier analysis and its applications.

1.7.4 Higher-order Fourier analysis applied to quantum computing

The Fourier transform is widely used and also underlies multiple famous quantum algorithms. Higher-order Fourier analysis has found many applications in theoretical computer science. However, the subfield of higher-order Fourier analysis used in quantum algorithms is relatively unexplored.

Rötteler presented a quantum algorithm to learn a hidden shift in functions with a large Gowers U^3-norm [Röt09]. Later, Montanaro presented a quantum algorithm to learn multilinear polynomials with improved query complexity over conventional algorithms [Mon12]. In the context of non-local games, higher-order Fourier analysis helps to prove a relation between the bias of a quantum strategy and the bias of a conventional strategy [Ban+19].

Only very recently, the application of higher-order Fourier analysis in quantum computing algorithms found new applications with the work of Arunachalam and Dutt in the context of stabilizer testing [AD24; BDH24; ABD24; Che+24]. These results consider the task of determining if a given unknown quantum state is close to or far from a stabilizer state.

Chapter 5 continues this line of research by providing a quantum analog of the quadratic Goldreich-Levin algorithm. With this algorithm, one can find a Reed-Muller codeword of degree at most 2 that correlates with the input, provided the input has large Gowers U^3-norm.

A key subroutine in that chapter is the Fourier sampling routine, which generalizes sampling the Fourier spectrum of a function using standard Fourier analytic

techniques. This Fourier sampling routine generalizes the celebrated Bernstein-Vazirani algorithm [BV97], which returns a string $a \in \mathbb{F}_p^n$ with probability $|\widehat{f}(a)|^2$ using a single query to a polynomial phase function $f : \mathbb{F}_p^n \to \mathbb{D}$. In particular, if $f = \omega_p^{\langle a,x\rangle+b}$ is a linear phase function given by some vector $a \in \mathbb{F}_p^n$ and scalar $b \in \mathbb{F}_p$, then the algorithm returns a with probability 1.

The Fourier sampling routine samples the Fourier spectrum of the multiplicative derivates of f, instead of the Fourier spectrum of f itself.

1.7.5. Lemma (Fourier sampling). *There is a p-query quantum algorithm that, given query access to a polynomial phase function $f : \mathbb{F}_p^n \to \mathbb{D}$ and input $h \in \mathbb{F}_p^n$, returns $a \in \mathbb{F}_p^n$ with probability exactly $|\widehat{\Delta_h f}(a)|^2$.*

Proof: Define the function $T^h f : \mathbb{F}_p^n \to \mathbb{D}$ by $T^h f(x) = f(x+h)$. Consider the unitary $S_h : (\mathbb{C}^p)^{\otimes n} \to (\mathbb{C}^p)^{\otimes n}$, defined by $|x\rangle \mapsto |x+h\rangle$. A query to $T^h f$ then corresponds to applying S_h, querying f, and applying S_{-h}. As one query to f introduces p-th roots of unity to the amplitudes, $p-1$ sequential queries to f are equivalent to a single query to $\overline{f}$.

We prove the lemma by outlining the algorithmic steps taken. Initialize a register $(\mathbb{C}^p)^{\otimes n}$ in the all-zeros state and create a uniform superposition over $\mathbb{F}_p^n$ by applying a quantum Fourier transform to the initial state. Next, query $T^h f$ and $\overline{f}$ to obtain the state

$$\frac{1}{\sqrt{p^n}} \sum_{x \in \mathbb{F}_p^n} f(x+h)\overline{f(x)}\,|x\rangle = \frac{1}{\sqrt{p^n}} \sum_{x \in \mathbb{F}_p^n} \Delta_h f(x)\,|x\rangle\,.$$

Finally, apply the inverse-quantum Fourier transform to obtain the state

$$\sum_{a \in \mathbb{F}_p^n} \left(\frac{1}{p^n} \sum_{x \in \mathbb{F}_p^n} \Delta_h f(x)\, \omega_p^{-\langle x,a\rangle} \right) |a\rangle = \sum_{a \in \mathbb{F}_p^n} \widehat{\Delta_h f}(a)\,|a\rangle\,.$$

Measuring in the computational basis gives the desired distribution. □

Note that this lemma implicitly uses quantum gates over p-level qudits, instead of over 2-level qubits. Section 4.7 briefly discusses a few of these gates.

1.8 From theory to practice

As quantum devices grow in computational power and more quantum algorithms are being developed, practical applications also come closer. With a quantum algorithm alone we however have not yet solved the computational problems encountered in practice.

Quantum computers usually solve only a small computationally hard problem in a larger sequence of algorithms, also called a workflow. As a result, quantum

computers have to be integrated into a larger (possibly existing) workflow. Furthermore, with an increasing number of quantum devices becoming available it might prove hard to choose the best device to solve your problem. Finally, multiple steps are required to obtain an implementation on quantum hardware, given a theoretical algorithm. This section briefly discusses these aspects relevant when performing quantum computations.

1.8.1 Integration in a large workflow: Hybrid quantum computing

Especially in the near-term, quantum computers will only solve a specific computationally hard problem. The outcome of the quantum algorithm is then processed by a conventional computer. For simple operations, conventional computers usually work significantly faster and quantum computers offer no benefit. Even on the long term, it is expected that quantum computers will only perform part of the computations of a larger workflow.

The interaction between quantum and conventional computers, and their integration in the same workflow, brought forward the term hybrid quantum computing [LU05]. However, this term is often used in varying situations, with its exact meaning differing each time [Per+14; Li+17; McC+20; Cer+21; De 21]. Often, the term hybrid quantum computing means that a quantum computer interacts with a conventional computer in some unspecified way.

Integrating conventional computers with quantum computers requires a clear definition of what is exactly meant. Such a definition benefits from an abstract point-of-view on the performed computations. Weder et al. provides such a view by considering workflows, a break-down of a computation in multiple high-level blocks [Wed+21]. An example workflow for clustering can for instance contain the high-level blocks *Retrieve data* and *Compute cluster*. Each block can represent a workflow itself.

Phillipson, Neumann, and Wezeman use this workflow approach to introduce new nomenclature to distinguish between various forms of hybrid quantum computing [PNW23]. They first make the distinction between vertical and horizontal hybrid computing. Vertical hybrid computing encompasses all activities required to control and operate a quantum circuit, independent of the actual algorithm. We distinguish between three different vertical hybrid forms of hybrid:

- Decomposition hybrid: All steps to decompose a high-level quantum routine into low-level instructions that correspond to the available gate set;
- Implementation hybrid: All steps to map low-level instructions to the quantum hardware, taking into account the scheduling of the gates and the hardware topology;

- Controlling hybrid: All steps to assure the quantum computer operates correctly and stays calibrated.

For horizontal hybrid computing, we distinguish between five different forms of hybrid:

- Processing hybrid: Only the pre- and post-processing steps are performed on a conventional device. Shor's algorithm is a prominent example of a processing hybrid algorithm;
- Macro/micro hybrid split: In the whole workflow, quantum and conventional computations alternate. The difference between macro and micro hybrid split depends mainly on the granularity of the workflow. Examples of a macro/micro hybrid splits originate typically in quantum machine learning, where data is prepared and a quantum circuit is run. We also find macro/micro hybrid splits in variational circuits, where a conventional computer optimizes the gate parameters of the quantum circuit;
- Parallel hybrid: Computations are solved in multiple independent branches, where each branch can employ its own algorithm. The workflow outputs the first (or best) solution found by the branches. The parallel hybrid computing is mainly used within current quantum annealing hardware [Wan+12b; Wan+12a; BRR17];
- Breakdown hybrid: A problem is decomposed in different smaller problems, each of which is solved independently, before being recombined. Divide-and-conquer approaches typically fall under breakdown hybrid [AS06; BSS16; Ara+21; Coj+21; Tom+23; Fan+24].

With hybrid quantum computing, also comes forth the topic of cloud-based quantum computing. Quantum computers will most likely act as a secondary processor to a conventional computer, similar to current graphical processing units (GPUs). Moreover, due to the often stringent operating conditions, quantum computers are unlikely to be hosted locally. Instead, we expect a few parties to host quantum computers, which others can use via remote connections. We call this type of computing cloud-based quantum computing.

Cloud-based computing is common with conventional computers, where large high-performance computers (HPCs) perform complex computations. With quantum computers, we are offered the opportunity to redefine the way we interact with cloud-based systems, based on experience from conventional cloud-based computing. Moreover, for a widespread adoption of quantum computing, cloud-based solutions should offer certain user-friendly functionalities.

Some functionalities such as appropriate device allocation, characterization of the quantum devices and automatic compilation of operations to hardware-aware

instructions are similar to conventional cloud-based systems. Others functionalities however are completely new, first and foremost the integration of quantum computers with conventional cloud-based services. Additionally, communication between various quantum computers via a quantum network allows for a significantly broader range of applications. Multiple initiatives are already underway to realize this ambition, both by companies [IBM25], as well as on a national and international level [Eur24].

Finally, these functionalities should also be tweaked based on user needs. Some users might want to have full control over the devices and the way operations are implemented, whereas others might only be interested in the final answer of a quantum algorithm. These different types of users translate into different services offered by a cloud-computing provider, as seen in conventional cloud-based computing [IDM18; NSS23].

Much work has already been done on how to set up a new cloud-based quantum platform [SS14; KSK19; NSS23]. Multiple cloud-based quantum computing platforms are already available. Most offer however only access to quantum computing hardware and do not offer integration with conventional computers or other quantum hardware.

1.8.2 Choosing quantum hardware: Quantum metrics

When using quantum devices, it is important to choose the one that best suits your goals. Number of qubits available is one way to choose a device, though other methods exist that might favor devices with fewer but better qubits. Quantum metrics provide insights in the performance of different quantum devices.

With varying interest in using quantum computers, also come various metrics to quantify their applicability. Typically, quantum metrics are grouped in one of three groups [Sch+23]:

- Component-level metrics;
- System-level metrics;
- Application-level metrics.

Component-level metrics focus on individual components of a quantum computer. This includes simple metrics such as the number of qubits available, and more elaborate metrics such as the coherence times of the qubits [Med+12; You20] and the quality of single-qubit and two-qubit quantum gates [MGE11; Nie+21]. These metrics are most suited for quantum hardware developers to tune the hardware and for low-level implementations.

System-level metrics focus on the system as a whole. A well-known system-level metric is the quantum volume [Cro+19], which quantifies how well quantum

computers can run random quantum circuits of some specific form. The Circuit Layer Operations Per Second (CLOPS) metric uses the quantum volume metric and also introduces a time component [Wac+21]. Another system-level metric is based on mirror circuits, where a circuit and its inverse are applied with an intermediate layer of random Pauli gates [Pro+21]. If we restrict the circuit to Clifford gates, then the inverse also consists of Clifford gates and the expected outcome and hence the device's performance is efficiently computed [Got98].

Application-level metrics consider the capabilities of a device to solve specific problems. These metrics focus on the actual use of a device in practice. As such, these metrics are typically agnostic to the hardware technology and, depending on the application considered, also allow to quantify the performance of heuristic methods.

Common application-level metrics include the QED-C metric, the Q-score, and QuAS. The first considers various tutorial algorithms, quantum subroutines and end-user applications to measure the performance of a device [Lub+23]. The Q-score originally considered the Max-Cut problem on random Erdös-Rényi $(N, \frac{1}{2})$-graphs [ER59] and compared these results to a random naive approach [MAA21]. The Q-score was later extended to include a time-aspect and different computational paradigms [Sch+22; Sch+24]. The QuAS metric combines elements from the Q-score and the QPack metric to evaluates the performance of quantum algorithms in different dimensions [MAM22; Mes+24].

As mentioned, different users require different types of metrics. Appropriate metrics give insight in which device works best for a given situation. It might even be that a conventional approach performs similar compared to a quantum approach. This comparison will however depend explicitly on the used algorithm, the used computational device and the problem instances.

1.8.3 From algorithm to implementation: Quantum programming

If we have a quantum algorithm and chosen a quantum device, we can start implementing the algorithm. This requires us to program our quantum algorithm and feed it to the computer, similar to programming a conventional computer.

Programming languages for conventional computers have been under development for decades. The first programming languages required the user to address the registers and program instruction close to the native hardware. Current conventional programming languages allow users to work with objects and classes. Thanks to compilers, users are not concerned with the exact underlying hardware.

Quantum programming languages have not seen this development yet and many steps are still to be taken [Pia+21]. Some quantum programming languages

already allow users to program somewhat higher-level instructions, such as applying a quantum Fourier transform on some qubits. However, most languages require users to program in a low-level hardware-agnostic manner using single- and two-qubit gates.

Eventually, compilers and transpilers running on conventional computers will help translate hardware-agnostic instructions into hardware-aware instructions. These hardware-aware instructions take into account the exact way the qubits are addressed, as well as the timing between different instructions. Ideally, compilers would also decompose high-level instruction into hardware-agnostic instructions. However, such decomposition techniques currently exist only in theory [VMS04; Zom+24].

As qubits easily decohere, quantum computers will require error-correction methods to mitigate these effects. There are results showing the reduction of error rates using these methods [Ach+24]. However, they are not yet applied automatically on hardware devices. In the long term, error-correction methods should be applied automatically, for instance, by the compiler and transpiler.

To utilize quantum solutions the soonest, developments in quantum algorithms and quantum hardware must go hand-in-hand with developments in integration of quantum solutions in larger workflows and developments in quantum programming languages.

1.9 Contributions

This work considers the added value of intermediate conventional computations for quantum computations. We refer to quantum computers that make use of intermediate conventional computations as *adaptive quantum computers*. We split the work into two parts. In the first part, we show that adaptive quantum computers are strictly stronger than conventional computers. In the second part, we show how adaptive quantum computations can help improve upon non-adaptive quantum computations. This section gives the papers on which this work was based and summarizes the contributions of this work together with references to the relevant sections or theorems.

The first part of this work considers a practical problem from computer science, namely, decoding corrupted error-correcting codes. We consider the setting where we restrict our circuits in depth, and then prove a quantum advantage with respect to this problem. Part One of this work is based on two papers:

- J. Briët, H. Buhrman, D. Castro-Silva, and N. M. P. Neumann. "Noisy Decoding by Shallow Circuits with Parities: Classical and Quantum (Extended Abstract)". In: *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Vol. 287. Leibniz International Proceedings in

Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. DOI: 10.4230/LIPIcs.ITCS.2024.21.

- J. Briët, D. Castro-Silva, and N. M. P. Neumann. "Quadratic Fourier analysis in quantum algorithms". Unpublished, to be submitted. 2025.

In this first part, we prove the following results:

- Polynomial maps of constant degree cannot decode any corrupted error-correcting code with constant success probability (Theorem 3.1.2). We conjecture that the found upper bound on the success probability can be improved to exponentially small, based on results for high field characteristics;
- From the result for polynomial maps we obtain that $\mathsf{NC}^0[\oplus]$-circuits cannot decode any corrupted error-correcting code with constant success probability (Theorem 3.1.1);
- We provide a family of quantum circuits to decode a corrupted Hadamard codeword with constant success probability (Theorem 4.1.1) and we extend this circuit to work in the high-error setting as well (Corollary 4.5.1);
- We obtain a separation between $\mathsf{NC}^0[\oplus]$ and $\mathsf{QNC}^0[\oplus]$, based on the result on polynomial maps and the quantum circuit for the Hadamard code (Theorem 4.5.3);
- From the $\mathsf{QNC}^0[\oplus]$-circuit, we construct a quantum circuit that computes Majority, a result known from literature, but presented without proof (Section 4.6);
- We provide a new algorithmic proof for the quadratic Goldreich-Levin theorem, using novel techniques from higher-order Fourier analysis. The algorithm can be used to retrieve a Reed-Muller codeword of degree 2 from a corrupted input with positive probability (Chapter 5). We additionally reduce the query complexity by omitting a Fourier estimation subroutine;
- We use a quantum Fourier sampling routine to further reduce the query complexity of obtaining the Reed-Muller codeword of degree 2 by a factor n (Theorem 5.1.1 and Section 5.7).

The second part of this work introduces a new computational model that alternates between quantum and conventional computing and limits the computational power of both. We then use that method to prepare quantum states and we conclude with an analysis of when such an adaptive approach is beneficial. Part Two of this work is based on two papers:

- H. Buhrman, M. Folkertsma, B. Loff, and N. M. P. Neumann. "State preparation by shallow circuits using feed forward". In: *Quantum* 8 (Dec.

2024), p. 1552. ISSN: 2521-327X. DOI: 10.22331/q-2024-12-09-1552. The second and fourth authors contributed equally to this result.

- H. Buhrman and N. M. P. Neumann. "Error-analysis of state preparation protocols using LAQCC". Unpublished work, to be submitted. 2025.

In this second part, we prove the following results:

- A new computational model *Local Alternating Quantum Classical Computations* (LAQCC), that alternates between quantum and conventional circuits. Our main interest is in a constant number of alternations between QNC^0-circuits and NC^1-circuits (Chapter 7). This instance of LAQCC is unlikely to be efficiently simulated (Section 7.5);
- We provide a LAQCC-circuit to prepare the uniform superposition over q states independent of the value of q (Theorem 8.2.1);
- We provide a LAQCC-circuit to prepare the W-state, based on a compress-uncompress method. This routine uses $\mathcal{O}(n\log(n)\log\log(n))$ qubits (Theorem 8.3.1);
- We extend this protocol to a LAQCC protocol to prepare the Dicke-(n,k) state, the uniform superposition over n-bitstrings of Hamming weight k for $k=\mathcal{O}(\sqrt{n})$. This routine uses $\mathcal{O}(n^3)$ qubits (Theorem 8.4.1);
- We use a different instance of LAQCC using a logarithmic number of alternations between QNC^0-circuits and NC^1-circuits to prepare the Dicke-(n,k) state for arbitrary k. This protocol uses $\mathcal{O}(\text{poly}(n))$ qubits and efficiently implements a map between different number representation systems (Theorem 8.5.1);
- We use a combinatorial worst-case error model to determine when a LAQCC-approach has a higher success probability in preparing the GHZ state and the W-state than a standard approach (Section 9.3 and Section 9.5).
- For the GHZ state, we implement both the standard approach and the LAQCC-approach and compare their success probabilities. We find that the standard approach performs better than the LAQCC-approach (Section 9.4).

Part One

Decoding

Chapter 2

Introduction to decoding error-correcting codes

This chapter provides an introduction to error-correcting codes and different error models. We also discuss the concept of list decoding, which becomes relevant in case of many errors. We briefly discuss circuits and query complexity, as in the remainder of this part, we will provide results with respect to those. We conclude with list-decoding algorithms beyond linear codes, and a brief discussion of deviation bounds that will be used to bound the probability of events happening.

2.1 Error-correcting codes

Error-correcting codes, introduced in Shannon's celebrated work [Sha48], protect digital signals from noise. For positive integers $n \geq k$, an error-correcting code over a finite alphabet Σ is a map $E : \Sigma^k \to \Sigma^n$, such that any message $x \in \Sigma^k$ can be decoded from the codeword $E(x)$ even if the codeword is partially corrupted. The magnitude of the corruption is measured by the Hamming distance $d(x, y)$, which counts the number of entries where x and y differ.

If too many errors occur, recovering the original message may become impossible. In such cases, one can instead resort to *list decoding*, an influential idea proposed in seminal works of Elias [Eli57] and Wozencraft [Woz58], which aims to give a small list of messages whose codewords are close to the received (corrupted) codeword. Complexity considerations appear naturally in this context, as encoding and decoding ideally allow for reliable communication with limited computational resources; they also appear because of the fundamental role played by error-correcting codes in computational complexity itself (see for instance [Tre04]).

Reed-Muller codes form a well-known class of error-correcting codes. Muller first proposed these codes and later Reed found an efficient decoding algorithm for them [Ree54; Mul54]. The popularity of Reed-Muller codes follows from their flexibility and as they generalize many codes, including Reed-Solomon codes [RS60]. A Reed-Muller code $RM(r, n)$ encodes a message of length $\sum_{i=0}^{r} \binom{n}{r}$ into a codeword of length 2^n. The codeword corresponds to the evaluation on all inputs $x \in \{0,1\}^n$ of a polynomial whose coefficients correspond to the input message. Polynomial interpolation helps to decode Reed-Muller codewords.

Reed-Muller codes also generalize the Hadamard code H, a basic but important example of a linear error-correcting code. The Hadamard code encodes k-bit messages into codewords of length $n = 2^k$ by evaluating all functions of degree at most 1: For message $x \in \mathbb{F}_2^k$, $H(x) = (\langle x, y \rangle)_{y \in \mathbb{F}_2^k}$, where $\langle x, y \rangle = \sum_{i=0}^{k-1} x_i y_i$.

2.2 Error models

In the error model considered by Shannon [Sha48], a codeword is corrupted by some random process. This process is described by the *symmetric channel*: for each coordinate of the codeword independently, the channel either transmits it unchanged with some probability ρ, or replaces it with a uniformly random element of Σ with probability $1 - \rho$; each coordinate is thus corrupted with probability $(1 - \rho)(1 - |\Sigma|^{-1})$. We refer to ρ as the *bias* of the channel. If $Z \in \Sigma^n$ is distributed according to the random outcome of the symmetric channel with bias ρ applied to a codeword $E(x)$, we write $Z \sim \mathcal{N}_\rho\big(E(x)\big)$. In this model, the goal is to correctly decode a corrupted codeword with good probability over the noise.

The combinatorial worst-case error model of Hamming [Ham50] instead assumes that the codeword is corrupted arbitrarily on at most some δ-fraction of the coordinates, for some *error parameter* $\delta \in [0, 1)$. In this setting, the number of errors that can be tolerated depends on the minimal Hamming distance between any pair of distinct codewords, or *minimal distance* of the code, denoted d_E. Since the Hamming ball of diameter $d_E - 1$ around any point $y \in \Sigma^n$ contains at most one codeword, a message can be retrieved if fewer than $d_E/2$ errors occur.

Every error-correcting code has a certain capacity, the theoretical maximum error rate below which messages can always be recovered. An interesting question is to develop an error-correcting code that actually achieves this capacity [Sha48].

2.3 List-decoding

If more errors occur, faithful decoding is no longer possible, and list decoding enters the picture. For error parameter $\delta \in [0, 1)$ and positive integer L, a code is (δ, L)-*list decodable* if for any point $y \in \Sigma^n$, the Hamming ball of radius δn

centered around y contains at most L codewords. It is well known that any (δ, L)-list decodable code satisfies $L \geq \Omega(1/\varepsilon^2)$ when $\delta = (1-\varepsilon)(1-|\Sigma|^{-1})$ [GV10]. If fewer than a δ-fraction of the codeword coordinates are corrupted, a random element from this list will give the correct message with probability at least $1/L$.

With growing error parameter δ, the list size also grows. However, as long as the error parameter remains below the capacity of the code, the list size remains constant. Yet, even list-decoding algorithms have their limits: If the error parameter increases too much, any list-decodable code has exponential list size. See Theorem 7.4.1 in Ref. [GRS22] for an exact definition and a proof.

In an influential paper in 1989, Goldreich and Levin gave the first efficient list-decoding algorithm [GL89]. They considered the Hadamard code. This code has minimal distance $n/2$ and is $(1/2-\varepsilon, O(1/\varepsilon^2))$-list decodable for any $\varepsilon \in (0, 1/2]$, which is known to be optimal for any code [GV10].

Under the symmetric channel, the Chernoff bound implies that unique decoding of the Hadamard code is possible with high probability for any constant bias $\rho > 0$. This result even holds for any code over a large enough alphabet [RU10]. This is due to the fact that, with high probability, the Hamming ball of radius $(1/4 - \rho/4)n$ around a corrupted version of a codeword C contains no other codewords than C itself.

For the worst-case Hamming model, Goldreich and Levin famously gave an efficient list decoding algorithm for the Hadamard code that runs in time $\mathrm{poly}(k, 1/\varepsilon)$, for error parameter $\delta = 1/2 - \varepsilon$ [GL89]. For fixed $\varepsilon > 0$, their algorithm gives a probabilistic AC^0-circuit that, on input length n, correctly returns the original message with probability $\Omega(1)$.

2.4 Constant-depth circuits

A well-studied problem is decoding corrupted error-correcting codes by constant-depth circuits, such as NC^0 or AC^0, for example in the context of black-box hardness amplification [STV99; Vio06; TV07]. Section 1.6.1 showed that these circuits are amenable to provable separations. It is worthwhile to study whether such a separation also exists for decoding highly-corrupted error-correcting codes.

As gates in NC^0-circuits have bounded-fan-in and the circuits have a constant depth, the outputs can only depend on a constant number of the inputs. However, the output of $\mathsf{NC}^0[\oplus]$-circuits can depend on all inputs, due to the unbounded-fan-in parity gate. Similarly, the output of $\mathsf{QNC}^0[\oplus]$-circuits can depend on the whole input.

In Chapter 3 and Chapter 4 we consider the classes $\mathsf{NC}^0[\oplus]$ and $\mathsf{QNC}^0[\oplus]$ and their capabilities in decoding corrupted error-correcting codes.

2.5 List-decoding beyond linear functions

The list-decoding algorithm by Goldreich and Levin works in greater generality to find large Fourier coefficients for any function.

Our quantum algorithm in Chapter 4 is an extension of the original Goldreich-Levin algorithm to a quantum setting, where we again sample the Fourier spectrum to find large Fourier coefficients. These large Fourier coefficients correspond precisely to possible original messages.

This quantum algorithm is inspired by the Bernstein-Vazirani algorithm [BV97], which in greater generality can be interpreted as a quantum algorithm to sample the Fourier spectrum of a function.

The Hadamard code is the simplest example of a Reed-Muller code. The next example would be to consider Reed-Muller codewords of degree at most 2, for which Tulsiani and Wolf gave a conventional list-decoding algorithm [TW14]. Their algorithm makes use of higher-order Fourier analysis. Furthermore, higher-order Fourier analysis helped to prove a qubic Goldreich-Levin algorithm [KLT23].

Only very few results are known that consider applications of higher-order Fourier analysis in quantum algorithms for decoding corrupted error-correcting codes. Montanaro took a first step in this direction by providing a quantum algorithm to learn uncorrupted multilinear polynomials of degree at most d using $\mathcal{O}(n^{d-1})$ quantum queries. However, this algorithm fails in case the queries can be corrupted. Chapter 5 extends this research line by considering learning functions that have large Gowers U^3-norm. This includes corrupted Reed-Muller codewords of degree at most 2.

2.6 Deviation bounds

In this part, we consider error models that corrupt codewords from some error-correcting codes. We interpret these codewords as strings with elements in a prime field. As we use probabilistic error models, we have to work with events that occur with certain probability. In most arguments, we will have to bound the probability of some event happening. Probability theory offers us the following two standard results which allow us to obtain these desired bounds.

The first result is by Markov, which gives an upper bound on the probability that some random variable is above some threshold.

2.6.1. Lemma (Markov's inequality). *Let $M > 0$ and let X be a random variable taking values in a finite set $S \subseteq (-\infty, M]$. Then, for any $a \in S$, we have that*

$$\Pr[X \geq a] \geq \frac{\mathbb{E}X - a}{M}.$$

Proof: We have

$$\begin{aligned}\mathbb{E}X &= \sum_{b\in S\cap(-\infty,a)} b\Pr[X=b] + \sum_{b\in S\cap[a,M]} b\Pr[X=b] \\ &\leq a\Pr[X\leq a] + M\Pr[X\geq a] \\ &\leq a + M\Pr[X\geq a].\end{aligned}$$

□

Additionally, Hoeffding's inequality upper bounds the probability that the sum of random variables differs from the expected value by more than a certain constant [Hoe63]. We will specifically use a complex version of Hoeffding's inequality. If all random variables are real valued, we can replace the 4 in the upper bound by a 2 in the lemma.

2.6.2. Lemma (Hoeffding's inequality). *Let $X_1,\ldots,X_n$ be independent $\mathbb{C}$-valued random variables such that $|X_i|\leq a_i$ for some $a_i>0$. Let $\overline{X}=n^{-1}(X_1+\cdots+X_n)$. Then, for any $\varepsilon>0$,*

$$\Pr\left[\left|\overline{X}-\mathbb{E}\overline{X}\right|>\varepsilon\right]\leq 4\exp\left(-\frac{2\varepsilon^2n^2}{\sum_{i=1}^n a_i^2}\right).$$

Proof: Let Y_i^0 and Y_i^1 be the real and complex parts of X_i, respectively. Then, $|Y_i^0|,|Y_i^1|\leq a_i$. For $b\in\{0,1\}$, let $\overline{Y^b}=n^{-1}(Y_1^b+\cdots+Y_n^b)$. By Hoeffding's inequality [Hoe63], for $b\in\{0,1\}$,

$$\Pr\left[\left|\overline{Y^b}-\mathbb{E}\overline{Y^b}\right|>\varepsilon\right]\leq 2\exp\left(-\frac{2\varepsilon^2n^2}{\sum_{i=1}^n a_i^2}\right).$$

The result now follows from the triangle inequality and the union bound. □

2.7 Outline

The remainder of this Part consists of three chapters: Chapter 3 presents a proof that polynomial maps of constant degree, and by extension also any $\mathsf{NC}^0[\oplus]$-circuit, cannot decode any corrupted error-correcting code with constant success probability. Chapter 4 shows that $\mathsf{QNC}^0[\oplus]$ circuits can decode a corrupted Hadamard code with constant success probability for fixed error parameter. Chapter 5 shows how to find a Reed-Muller codeword of degree 2 that correlates with a polynomial phase function given as input.

In the next sections, $C,c>0$ will denote absolute constants whose values may be different at each occurrence.

Chapter 3

Conventional hardness of list decoding

In this chapter we prove our main result for conventional approaches to decoding, which holds for any error-correcting code. We start slowly by showing the result for linear maps, which serves as a warm-up for the more general case of polynomial maps. We end with an improved result in the high-characteristic setting that hints towards possible improvements of our results.

3.1 Chapter overview

The main theorem proven in this chapter states that $\mathsf{NC}^0[\oplus]$-circuits cannot decode error-correcting codes with constant success probability.

3.1.1. Theorem (Impossibility of decoding by $\mathsf{NC}^0[\oplus]$). *For any $\rho \in [0,1)$, $d \in \mathbb{N}$ and $\varepsilon \in (0,1]$, there is a $k_0 = k_0(d,\rho,\varepsilon) \in \mathbb{N}$ such that the following holds. Let $k \geq k_0$ and n be positive integers, $E : \mathbb{F}_2^k \to \mathbb{F}_2^n$ be any map and $\phi : \mathbb{F}_2^n \to \mathbb{F}_2^k$ be a map computable by an $\mathsf{NC}^0[\oplus]$-circuit of depth at most d. Then, for a uniformly distributed $x \in \mathbb{F}_2^k$ and $Z \sim \mathcal{N}_\rho(0)$, we have that*

$$\Pr\big[\phi\big(E(x) + Z\big) = x\big] < \varepsilon. \tag{3.1}$$

In particular, this theorem shows that no $\mathsf{NC}^0[\oplus]$-circuit can correctly decode more than an ε-fraction of codewords with probability higher than ε over the noise distribution, provided the messages are long enough depending on ε, the error rate $(1-\rho)/2 > 0$, and the depth of the circuit. By Yao's minimax principle [Yao77] and the Chernoff bound, it follows that any probabilistic $\mathsf{NC}^0[\oplus]$-circuit also fails (with high probability) to correctly decode any binary error-correcting code in the worst-case Hamming model, for any constant error parameter $\delta \in (0, 1/2]$.

To prove this theorem, we use the basic observation that any function $f : \mathbb{F}_2^n \to \mathbb{F}_2^k$ that is computable by an $\mathsf{NC}^0[\oplus]$-circuit can be given by a collection of k constant-degree polynomials over $\mathbb{F}_2$ in n variables. Indeed, any gate with fan-in d implements a function $g : \mathbb{F}_2^d \to \mathbb{F}_2$ and any such function can be represented by a d-variable polynomial of total degree at most d. Degree is multiplicative under composition and composition occurs only between different layers of the circuit. Since the parities amount to addition in $\mathbb{F}_2$ and NC^0-circuits have constant depth, the total degree of the output is bounded.

We will therefore study the distribution of polynomial maps under biased input distributions. We will do so in a slightly more general setting over arbitrary finite fields of prime order[3]. For $\rho \in [0,1]$, an $\mathbb{F}_p$-valued random variable Z is *ρ-biased* if with probability ρ it equals 0 and with probability $1-\rho$ it is uniformly distributed over $\mathbb{F}_p$. Note that this corresponds to the noise $\mathcal{N}_\rho(0)$ added by the symmetric channel when the alphabet is $\mathbb{F}_p$.

A mapping $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ is called a *polynomial map* if there exist polynomials $f_1, \ldots, f_k \in \mathbb{F}_p[x_1, \ldots, x_n]$ such that $\phi = (f_1, \ldots, f_k)$. The degree of ϕ is the maximal degree among the f_i. To prove Theorem 3.1.1, it thus suffices to prove the following result.

3.1.2. Theorem (Impossibility of decoding by polynomial maps). *For every integer d and any $\rho, \varepsilon \in (0,1)$, there exists an integer $k_0 = k_0(p, d, \rho, \varepsilon)$ such that the following holds. Let $k \geq k_0$ and n be integers, $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ be a polynomial map of degree at most d and $E : \mathbb{F}_p^k \to \mathbb{F}_p^n$ be an arbitrary function. Then*

$$\Pr_{x \in \mathbb{F}_p^k, Z \sim \mathcal{N}_\rho(0)} \big[\phi\big(E(x) + Z\big) = x\big] \leq \varepsilon. \tag{3.2}$$

Studying the distribution of polynomial maps in many variables over a finite field falls within the purview of additive combinatorics. In the "unbiased" situation where Z is uniformly distributed, higher-order Fourier analysis provides powerful tools to study the distribution of $\phi(Z)$. In particular, Green and Tao [GT09] proved that if ϕ is "regular" (random-like), then $\phi(Z)$ is approximately uniformly distributed over $\mathbb{F}_p^k$. This implies that, for every $x \in \mathbb{F}_p^n$, the probability of the event $\{\phi(E(x) + Z) = x\}$ is small. A "regularity-type" lemma proved in [GT09] shows that one can "force" ϕ to be regular by restricting it to a partition defined by sufficiently many polynomial equations of degree less than the degree of ϕ. However, these techniques cause the size of the polynomial map ϕ considered to blow up considerably, and are only effective if k is an extremely slowly growing function of n.

To address this issue and to adapt these results to the case where Z is no longer uniform but biased, we employ a dichotomy often used in additive combinatorics.

[3]The restriction to prime order is done for notational reasons and for ease of exposition. Our arguments can be readily adapted to the case of non-prime finite fields.

This approach separately studies the "pseudorandom" case of regular maps and the "structured" case of maps that carry a certain algebraic structure. This is done in Section 3.3 by defining and studying a new notion of rank for (high-dimensional) polynomial maps, which we call the *analytic rank*[4], and which measures how equidistributed the values taken by the considered map are.

We consider the pseudorandom case in Section 3.4. The main tool we use in this case is a random restriction result for high-rank polynomial maps proved by Briët and Castro-Silva [BC24]. We use this result to show that the distribution of values taken by a high-rank polynomial map will be close to uniform even under a biased input distribution. This implies that the event considered in the theorem has very low probability for any fixed x, in which case we can conclude by averaging.

In the structured case we deal instead with polynomial maps of low rank, whose values are in a sense poorly distributed. Results from higher-order Fourier analysis then imply that they can be determined by "few" lower-degree polynomial maps (plus a few extra polynomials); by a simple Fourier-analytic argument we can reduce the analysis of a low-rank polynomial map to those lower-degree maps that specify it, making it amenable to an inductive argument. We use this idea in Section 3.5 where we prove Theorem 3.1.2.

The decay found on the probability in Equation (3.1) of correct message retrieval as a function of the message length is extremely slow, making Theorem 3.1.1 a qualitative result rather than quantitative. Nevertheless, we conjecture that the true decay of this probability is exponential in the message length k; this would clearly be optimal, as can be seen by taking a constant map ϕ which always returns some fixed message. In Sections 3.6 to 3.8 we will provide some evidence to support this conjecture.

We start off gradually by first proving in Section 3.2 that linear maps have exponentially small success probability in decoding corrupted error-correcting codes.

3.2 Impossibility of decoding linear maps

As a warm-up to our later arguments, we here present a proof of the first nontrivial case of Theorem 3.1.2, namely that of maps $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree 1. In this case, there is a matrix $U \in \mathbb{F}_p^{k\times n}$ and a vector $v \in \mathbb{F}_p^k$ such that

$$\phi(y) = Uy + v \quad \text{for all } y \in \mathbb{F}_p^n.$$

[4] A very similar notion of rank was defined for multilinear forms by Gowers and Wolf [GW11], who coined the term analytic rank when studying linear systems of equations over $\mathbb{F}_p^n$. We use the same name to highlight the similarity between our two notions, which are relevant for distinct types of mathematical objects.

Let x be a uniformly distributed random variable over $\mathbb{F}_p^k$ and let Z be an $\mathbb{F}_p$-valued *ρ-biased* random variable, meaning that with probability ρ it equals 0, and with probability $1-\rho$ it is uniformly distributed over $\mathbb{F}_p$. We denote this distribution by $\mathcal{N}_\rho(0)$. Our goal is then to bound the probability of the event

$$U(E(x)+Z)+v=x. \tag{3.3}$$

We follow a structure-versus-randomness approach, commonly used in additive combinatorics. For that we distinguish between U having low rank, the *structure* case, and U having high rank, the *randomness* case. We bound the probability of event $\mathcal{E}$ for both cases separately. Let $r=k/2$ be an integer.

3.2.1 Structured low-rank case

If U has rank at most r, then its image $\mathrm{Im}(U)$ is a subspace of size at most p^r. If event (3.3) holds, then x is contained in the coset $v+\mathrm{Im}(U)$ of this subspace, which (for x uniform over $\mathbb{F}_p^k$) happens with probability at most p^r/p^k. Hence, event (3.3) holds with probability at most $p^{-(k-r)}$ in this case.

3.2.2 Pseudorandom high-rank case

For the "pseudorandom case" of high-rank matrices, we make the following simple but important observation: one can sample $Z\sim\mathcal{N}_\rho(0)$ by first sampling the set $I\subseteq[n]$ of "corrupted coordinates", then sampling the "noise" y uniformly at random from $\mathbb{F}_p^I$ and setting[5] $Z_{|I}=y$, $Z_{|[n]\setminus I}=0$. Each index $i\in[n]$ has probability $1-\rho$ of belonging to the random set I, with these events being mutually independent; we denote this sampling scheme by $I\sim[n]_{1-\rho}$.

If we denote by $U_I\in\mathbb{F}_p^{k\times I}$ the restriction of U to the columns indexed by $I\subseteq[n]$, it follows that the random variable UZ has the same distribution as the random variable U_Iy, where $I\sim[n]_{1-\rho}$ and y is uniformly distributed over $\mathbb{F}_p^I$. Thus, for any given $x\in\mathbb{F}_p^k$, we have

$$\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[U(E(x)+Z)+v=x\big]=\mathbb{E}_{I\sim[n]_{1-\rho}}\Pr_{y\in\mathbb{F}_p^I}\big[U_Iy=x-UE(x)-v\big].$$

Now, if $I\subseteq[n]$ is fixed and y is uniformly distributed over $\mathbb{F}_p^I$, then the random variable U_Iy is uniformly distributed over $\mathrm{Im}(U_I)$; hence

$$\max_{w\in\mathbb{F}_p^k}\Pr_{y\in\mathbb{F}_p^I}\big[U_Iy=w\big]=\frac{1}{|\,\mathrm{Im}(U_I)|}=\frac{1}{p^{\mathrm{rk}(U_I)}}.$$

Taking the expectation over $I\sim[n]_{1-\rho}$ and $x\in\mathbb{F}_p^k$, we conclude that event (3.3) holds with probability at most $\mathbb{E}_{I\sim[n]_{1-\rho}}p^{-\mathrm{rk}(U_I)}$.

[5]Given $x\in\mathbb{F}_p^n$ and $I\subseteq[n]$, we denote by $x_{|I}\in\mathbb{F}_p^I$ the restriction of x to the coordinates indexed by I.

Suppose now that U has rank at least r, and let $J \subseteq [n]$ be a set of r linearly independent columns of U. By the Chernoff bound (see e.g. [HR90]), we have that

$$\Pr_{I \sim [n]_{1-\rho}}\left[|I \cap J| \leq \frac{(1-\rho)r}{2}\right] \leq e^{-(1-\rho)r/8}.$$

Thus, the matrix U_I will contain more than $(1-\rho)r/2$ linearly independent columns with probability at least $1 - e^{-(1-\rho)r/8}$; whenever this happens we have $\operatorname{rk}(U_I) \geq (1-\rho)r/2$. It follows that

$$\begin{aligned}\Pr_{x \in \mathbb{F}_p^k, Z \sim \mathcal{N}_\rho(0)}\big[U(E(x)+Z)+v = x\big] &\leq \mathbb{E}_{I \sim [n]_{1-\rho}} p^{-\operatorname{rk}(U_I)} \\ &\leq e^{-(1-\rho)r/8} + p^{-(1-\rho)r/2}.\end{aligned}$$

The choice of $r = k/2$ implies an exponential decay of the probability that event (3.3) holds for both cases, concluding the analysis.

3.3 The analytic rank of polynomial maps

Green and Tao introduced the notion of rank for polynomials $P \in \mathbb{F}_p[x_1, \ldots, x_n]$, defined as the smallest number of lower-degree polynomials needed to compute P. It is related to the *bias* of the polynomial P, or more specifically to the bias of the symmetric $\deg(P)$-multilinear form associated to P. The bias of a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ is an analytic measure of how well-equidistributed the values of f are when evaluated on a uniformly random input; formally,

$$\operatorname{bias}(f) = \left|\mathbb{E}_{x \in \mathbb{F}_p^n} \omega_p^{f(x)}\right|. \tag{3.4}$$

When dealing with a polynomial P of some bounded degree d, having non-negligible bias implies that it has a significant amount of internal structure. Such a result was first proven by Green and Tao [GT09] in the case of polynomials with degree d smaller than the characteristic p of the field considered, and motivated the introduction of both their notion of rank and Gowers and Wolf's notion of analytic rank [GW11]. We will need a similar result, proven by Kaufman and Lovett [KL08], that generalizes this theorem to higher characteristics $p \leq d$ and also gives more precise information on the structure of the polynomial.

The following result of Kaufman and Lovett shows that polynomials with large bias must be highly structured:

3.3.1. Theorem (Bias implies low rank; Theorem 4 [KL08]). *For every $d \in \mathbb{N}$ and $\varepsilon > 0$, there is an $r = r(p, d, \varepsilon) \in \mathbb{N}$ such that the following holds. If $P \in \mathbb{F}_p[x_1, \ldots, x_n]$ is a polynomial of degree at most d with* $\operatorname{bias}(P) \geq \varepsilon$*, then there exist $h_1, \ldots, h_r \in \mathbb{F}_p^n$ and a map $\Gamma : \mathbb{F}_p^r \to \mathbb{F}_p$ such that*

$$P(x) \equiv \Gamma\big(\Delta_{h_1} P(x), \ldots, \Delta_{h_r} P(x)\big).$$

For integers $d, n, k \geq 1$, we denote by $\mathrm{Pol}_{\leq d}(\mathbb{F}_p^n, \mathbb{F}_p^k)$ the space of all polynomial maps $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree at most d.

3.3.2. Definition (Analytic rank). Given a polynomial map $\phi \in \mathrm{Pol}_{\leq d}(\mathbb{F}_p^n, \mathbb{F}_p^k)$, we define its analytic rank $\mathrm{arank}_d(\phi)$ by

$$\mathrm{arank}_d(\phi) = -\log_p \left(\max_{\psi : \mathbb{F}_p^n \to \mathbb{F}_p^k,\, \deg(\psi) < d} \Pr_{x \in \mathbb{F}_p^n} \big[\phi(x) = \psi(x) \big] \right).$$

Note that, for affine-linear maps $\phi \in \mathrm{Pol}_{\leq 1}(\mathbb{F}_p^n, \mathbb{F}_p^k)$, this definition coincides with the usual notion of rank for the matrix $U \in \mathbb{F}_p^{k \times n}$ encoding its linear part. Indeed, suppose $\phi(x) = Ux + v$ for some $v \in \mathbb{F}_p^k$. Since Ux is uniformly distributed over $\mathrm{Im}(U) \simeq \mathbb{F}_p^{\mathrm{rk}(U)}$ when x is uniformly distributed over $\mathbb{F}_p^k$, we have that

$$\Pr_{x \in \mathbb{F}_p^n} \big[Ux + v = w \big] = \begin{cases} p^{-\mathrm{rk}(U)} & \text{if } w - v \in \mathrm{Im}(U), \\ 0 & \text{if } w - v \notin \mathrm{Im}(U). \end{cases}$$

This example might help explain the reason for the $-\log_p$ in the definition of analytic rank, as well as the need to maximize the probability of agreement over all lower-degree maps.

Another useful way of viewing the analytic rank of a polynomial map ϕ is as a measure of how well-equidistributed its values are in $\mathbb{F}_p^k$, up to lower-degree perturbations. Indeed, we can equivalently write

$$\mathrm{arank}_d(\phi) = \min_{\psi : \mathbb{F}_p^n \to \mathbb{F}_p^k,\, \deg(\psi) < d} -\log_p \left(\mathbb{E}_{v \in \mathbb{F}_p^k, x \in \mathbb{F}_p^n} \omega_p^{\langle v,\, \phi(x) - \psi(x) \rangle} \right).$$

The expectation inside the logarithm above is analogous to the notion of bias given in Equation (3.4), and can be seen as an analytic measure of how close to uniformly distributed over $\mathbb{F}_p^k$ the values taken by $\phi - \psi$ are.

It is clear from the definition that the function arank_d is nonnegative (since probabilities are bounded by 1), and that $\mathrm{arank}_d(\phi) = 0$ if and only if $\deg(\phi) \leq d-1$. It also shares several useful properties with the rank of matrices; in order to state them we will need some notation for considering coordinate restrictions:

3.3.3. Definition (Restriction). For a polynomial map $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ and subset $I \subseteq [n]$, we define the restriction $\phi_{|I} : \mathbb{F}_p^I \to \mathbb{F}_p^k$ to be the map given by $\phi_{|I}(y) = \phi(y_I)$, where $y_I \in \mathbb{F}_p^n$ agrees with y on the coordinates in I and is zero elsewhere.

The properties of analytic rank that will be important to us are summarized in the next lemma.

3.3.4. Lemma (Properties of analytic rank). *For all integers $d, n, k \geq 1$, the analytic rank function* arank_d *satisfies:*

1. *Symmetry:*
   $\operatorname{arank}_d(\phi) = \operatorname{arank}_d(-\phi)$ *for all* $\phi \in \operatorname{Pol}_{\leq d}(\mathbb{F}^n, \mathbb{F}^k)$.
2. *Sub-additivity:*
   $\operatorname{arank}_d(\phi + \gamma) \leq \operatorname{arank}_d(\phi) + \operatorname{arank}_d(\gamma)$ *for all* $\phi, \gamma \in \operatorname{Pol}_{\leq d}(\mathbb{F}^n, \mathbb{F}^k)$.
3. *Monotonicity under restrictions:*
   $\operatorname{arank}_d(\phi_{|I}) \leq \operatorname{arank}_d(\phi)$ *for all* $\phi \in \operatorname{Pol}_{\leq d}(\mathbb{F}^n, \mathbb{F}^k)$ *and all sets* $I \subseteq [n]$.
4. *Restriction Lipschitz property:*
   $\operatorname{arank}_d(\phi_{|I \cup J}) \leq \operatorname{arank}_d(\phi_{|I}) + |J|$ *for all polynomial maps* $\phi \in \operatorname{Pol}_{\leq d}(\mathbb{F}^n, \mathbb{F}^k)$ *and all sets* $I, J \subseteq [n]$.

Proof: The first property is trivial. To prove property 2, let $\psi, \chi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ be polynomial maps of degree at most $d-1$ such that

$$\begin{aligned}\operatorname{arank}_d(\phi) &= -\log_p \Pr_{x \in \mathbb{F}_p^n}\big[\phi(x) = \psi(x)\big],\\ \operatorname{arank}_d(\gamma) &= -\log_p \Pr_{x \in \mathbb{F}_p^n}\big[\gamma(x) = \chi(x)\big].\end{aligned}$$

Then $p^{-\operatorname{arank}_d(\phi) - \operatorname{arank}_d(\gamma)}$ can be expressed as

$$\begin{aligned}&\Pr_{x,y \in \mathbb{F}_p^n}\big[\phi(x) = \psi(x) \wedge \gamma(y) = \chi(y)\big]\\ &\qquad = \Pr_{x,y}\big[\phi(x) = \psi(x) \wedge \phi(x) + \gamma(x+y) = \psi(x) + \chi(x+y)\big],\end{aligned}$$

where we performed the change of variables $(x, y) \mapsto (x, x+y)$. Now, as

$$\gamma(x+y) = \gamma(x) + \Delta_y \gamma(x),$$

this equals

$$\begin{aligned}&\Pr_{x,y}\big[\phi(x) = \psi(x) \wedge \phi(x) + \gamma(x) = \psi(x) + \chi(x+y) - \Delta_y\gamma(x)\big]\\ &\qquad \leq \Pr_{x,y}\big[\phi(x) + \gamma(x) = \psi(x) + \chi(x+y) - \Delta_y\gamma(x)\big].\end{aligned}$$

Note that, for any fixed $y \in \mathbb{F}_p^n$, the function

$$x \mapsto \psi(x) + \chi(x+y) - \Delta_y\gamma(x)$$

is a polynomial map of degree at most $d-1$, as every term has degree at most $d-1$. The last probability above is then bounded by

$$\begin{aligned}&\max_y \Pr_x\big[\phi(x) + \gamma(x) = \psi(x) + \chi(x+y) - \Delta_y\gamma(x)\big]\\ &\qquad \leq \max_{\xi : \mathbb{F}_p^n \to \mathbb{F}_p^k,\, \deg(\xi) < d} \Pr_x\big[\phi(x) + \gamma(x) = \xi(x)\big]\\ &\qquad = p^{-\operatorname{arank}_d(\phi+\gamma)}.\end{aligned}$$

Sub-additivity now follows by taking logarithms.

To prove property 3 it suffices to show that $\operatorname{arank}_d(\phi_{|[n]\setminus\{i\}}) \le \operatorname{arank}_d(\phi)$ for any $i \in [n]$, which can then be applied iteratively. Assume for notational convenience that $i = n$, and let $\psi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ be a polynomial map of degree at most $d-1$ which satisfies

$$p^{-\operatorname{arank}_d(\phi)} = \Pr_{x\in\mathbb{F}_p^n}[\phi(x) = \psi(x)].$$

Factoring out the variable x_n allows us to write the probability on the right-hand side as

$$\mathbb{E}_{x_n\in\mathbb{F}_p}\Pr_{y\in\mathbb{F}_p^{n-1}}\big[\phi_{|[n-1]}(y) + \phi'(y,x_n)x_n = \psi_{|[n-1]}(y) + \psi'(y,x_n)x_n\big],$$

where ϕ' and ψ' are some polynomial maps of degree at most $d-1$. By the averaging principle, this is at most

$$\begin{aligned}\max_{x_n\in\mathbb{F}_p} &\Pr_{y\in\mathbb{F}_p^{n-1}}\big[\phi_{|[n-1]}(y) = \psi_{|[n-1]}(y) + \psi'(y,x_n)x_n - \phi'(y,x_n)x_n\big]\\ &\le \max_{\xi:\mathbb{F}_p^{n-1}\to\mathbb{F}_p^k,\,\deg(\xi)<d} \Pr_{y\in\mathbb{F}_p^{n-1}}\big[\phi_{|[n-1]}(y) = \psi_{|[n-1]}(y) + \xi(y)\big]\\ &= p^{-\operatorname{arank}_d(\phi_{|[n-1]})},\end{aligned}$$

showing that $\operatorname{arank}_d(\phi_{|[n-1]}) \le \operatorname{arank}_d(\phi)$ as wished.

Finally, for the Lipschitz property 4, let $\psi : \mathbb{F}_p^I \to \mathbb{F}_p^k$ be a map with $\deg(\psi) < d$ maximizing the agreement probability $\Pr_{x\in\mathbb{F}_p^I}\big[\phi_I(x) = \psi(x)\big]$, and suppose without loss of generality that $J \cap I = \emptyset$. Then

$$\begin{aligned}p^{-\operatorname{arank}_d(\phi_{|I\cup J})} &\ge \Pr_{x\in\mathbb{F}_p^I,\,y\in\mathbb{F}_p^J}\big[\phi_{|I\cup J}(x,y) = \psi(x)\big]\\ &\ge \Pr_{x\in\mathbb{F}_p^I,\,y\in\mathbb{F}_p^J}\big[\phi_{|I\cup J}(x,0) = \psi(x) \wedge y = 0\big]\\ &= p^{-|J|}\Pr_{x\in\mathbb{F}_p^I}\big[\phi_I(x) = \psi(x)\big]\\ &= p^{-\operatorname{arank}_d(\phi_I)-|J|},\end{aligned}$$

and the restriction Lipschitz property follows. □

3.4 Biased equidistribution of high-rank maps

As in the degree-1 case considered in Section 3.2, we will need to study the distribution of values $\phi(Z)$ taken by a polynomial map ϕ when the input is a ρ-biased random variable $Z \sim \mathcal{N}_\rho(y)$. This can be done by considering restrictions of ϕ to random subsets of variables, which model the coordinates "corrupted" by the random process.

As the analytic rank satisfies all properties of Lemma 3.3.4, Theorem 1.8 of Ref. [BC24] immediately gives that the random restrictions of a high-rank polynomial map will also have high rank with high probability, summarized in the next theorem.

3.4.1. Theorem (Random-restriction theorem). *For every integer d and every $\sigma, \varepsilon \in (0,1]$, there exist $\kappa = \kappa(d,\sigma) > 0$ and $R = R(d,\sigma,\varepsilon) \in \mathbb{N}$ such that the following holds. For every map $\phi \in \mathrm{Pol}_{\leq d}(\mathbb{F}^n, \mathbb{F}^k)$ with $\operatorname{arank}_d(\phi) \geq R$, we have that*

$$\Pr_{I\sim[n]_\sigma}\big[\operatorname{arank}_d(\phi_{|I}) \geq \kappa \cdot \operatorname{arank}_d(\phi)\big] \geq 1-\varepsilon.$$

With the help of this theorem, it is easy to show that high-rank polynomial maps are approximately equidistributed even under biased inputs:

3.4.2. Lemma (Biased equidistribution lemma). *For every integer d and every $\rho, \varepsilon \in (0,1)$ there exists a constant $R_0 = R_0(d,\rho,\varepsilon) > 0$ such that the following holds. If $\phi \in \mathrm{Pol}_{\leq d}(\mathbb{F}^n, \mathbb{F}^k)$ satisfies $\operatorname{arank}_d(\phi) \geq R_0$, then*

$$\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi(y+Z) = w\big] \leq \varepsilon \quad \textit{for all } y \in \mathbb{F}_p^n, w \in \mathbb{F}_p^k.$$

Recall that $I \sim [n]_\sigma$ denotes the random process of sampling a subset $I \subseteq [n]$ where each $i \in [n]$ belongs to I with probability σ, all events being mutually independent.

Proof: It suffices to prove the special case where both y and w are zero, that is

$$\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi(Z) = 0\big] \leq \varepsilon.$$

Indeed, for fixed $y \in \mathbb{F}_p^n$ and $w \in \mathbb{F}_p^k$, the map $\tilde{\phi} : x \mapsto \phi(y+x) - w$ has the same degree and analytic rank as ϕ, and satisfies $\tilde{\phi}(x) = 0$ if and only if $\phi(y+x) = w$.

We can sample $Z \sim \mathcal{N}_\rho(0)$ by first sampling $I \sim [n]_{1-\rho}$ (the "corrupted coordinates"), then sampling z uniformly from $\mathbb{F}_p^I$ (the "noise") and setting $Z_{|I} = z$, $Z_{|[n]\setminus I} = 0$; thus

$$\begin{aligned}\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi(Z) = 0\big] &= \mathbb{E}_{I\sim[n]_{1-\rho}}\Pr_{z\in\mathbb{F}_p^I}\big[\phi_{|I}(z) = 0\big] \\ &\leq \mathbb{E}_{I\sim[n]_{1-\rho}} p^{-\operatorname{arank}_d(\phi_{|I})}.\end{aligned}$$

Let $R = R(d, 1-\rho, \varepsilon/2)$ and $\kappa = \kappa(d, 1-\rho)$ be the constants guaranteed by Theorem 3.4.1. If $\operatorname{arank}_d(\phi) \geq R$, from that result we obtain

$$\mathbb{E}_{I\sim[n]_{1-\rho}} p^{-\operatorname{arank}_d(\phi_{|I})} \leq \varepsilon/2 + p^{-\kappa\cdot\operatorname{arank}_d(\phi)}.$$

Taking[6] $R_0 = \max\big\{R, \log_p(2/\varepsilon)/\kappa\big\}$ we conclude that

$$\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi(Z) = 0\big] \leq \mathbb{E}_{I\sim[n]_{1-\rho}} p^{-\operatorname{arank}_d(\phi_{|I})} \leq \varepsilon$$

whenever $\operatorname{arank}_d(\phi) \geq R_0$, as wished. □

[6]Note that this bound is non-increasing on the value of p, so we can obtain a field-independent bound by considering the smallest case $p = 2$.

3.5 The proof of Theorem 3.1.2

We are now ready to present the proof of Theorem 3.1.2, which proceeds by induction on the degree d. For degree-1 maps the result was already proven in the warm-up section,[7] so let $d \geq 2$ and assume the result holds for maps of degree at most $d-1$.

Similar to the base case, we will divide the argument into two parts, corresponding to whether the analytic rank of ϕ is "high" (the pseudorandom case) or "low" (the structured case). The pseudorandom case immediately follows from Lemma 3.4.2, the biased equidistribution lemma: let $R_0 = R_0(d, \rho, \varepsilon)$ be the constant guaranteed by that lemma, and suppose that $\operatorname{arank}_d(\phi) > R_0$. Then for every $x \in \mathbb{F}_p^k$ we have that

$$\Pr_{Z \sim \mathcal{N}_\rho(0)}\big[\phi\big(E(x) + Z\big) = x\big] \leq \varepsilon,$$

and we conclude by averaging over all such x.

For the structured case, suppose that $\operatorname{arank}_d(\phi) \leq R_0$, and let $\psi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ be a map of degree at most $d-1$ such that $\Pr_{x \in \mathbb{F}_p^n}\big[\phi(x) = \psi(x)\big] \geq p^{-R_0}$. For convenience, denote $\tilde{\phi} = \phi - \psi$, and let $P \in \mathbb{F}_p[y_1, \ldots, y_n, v_1, \ldots, v_k]$ be the polynomial given by

$$P(y, v) = \langle v, \tilde{\phi}(y) \rangle.$$

This polynomial has nonnegligible bias:

$$\operatorname{bias}(P) = \mathbb{E}_{y \in \mathbb{F}_p^n} \mathbb{E}_{v \in \mathbb{F}_p^k} \omega_p^{\langle v, \tilde{\phi}(y) \rangle} = \mathbb{E}_{y \in \mathbb{F}_p^n} \mathbf{1}\big[\tilde{\phi}(y) = 0\big] \geq p^{-R_0}.$$

By Theorem 3.3.1, there exist $s = s(p, d, R_0) \in \mathbb{N}$, a map $\Gamma : \mathbb{F}_p^s \to \mathbb{F}_p$ and pairs $(h_1, w_1), \ldots, (h_s, w_s) \in \mathbb{F}_p^n \times \mathbb{F}_p^k$ such that

$$P(y, v) = \Gamma\big(\Delta_{(h_1, w_1)} P(y, v), \ldots, \Delta_{(h_s, w_s)} P(y, v)\big).$$

Let $f : \mathbb{F}_p^s \to \mathbb{C}$ be the map given by $f(t) = \omega_p^{\Gamma(t)}$ and let $\widehat{f} : \mathbb{F}_p^s \to \mathbb{C}$ be its Fourier transform,

$$\widehat{f}(\alpha) = \mathbb{E}_{t \in \mathbb{F}_p^s} f(t) \omega_p^{-\langle \alpha, t \rangle}.$$

Since P is linear in v, the last k coordinates, it follows that

$$\begin{aligned} \Delta_{(h,w)} P(y, v) &= P(y + h, v + w) - P(y + h, v) + P(y + h, v) - P(y, v) \\ &= \langle w, \tilde{\phi}(y + h) \rangle + \langle v, \Delta_h \tilde{\phi}(y) \rangle. \end{aligned}$$

[7] It would also be possible to start the induction from the trivial base case $d = 0$ of constant maps, but we thought it is more instructive to first present the argument for degree-1 maps in order to gain some intuition.

By the Fourier inversion formula (Equation (1.11)), we conclude that

$$\begin{aligned}\omega_p^{P(y,v)} &= f\big(\Delta_{(h_1,w_1)}P(y,v),\dots,\Delta_{(h_s,w_s)}P(y,v)\big)\\ &= \sum_{\alpha\in\mathbb{F}_p^s}\widehat{f}(\alpha)\omega_p^{Q_\alpha(y)+\langle v,\gamma_\alpha(y)\rangle},\end{aligned}$$

where for $\alpha\in\mathbb{F}_p^s$ we denote

$$\begin{aligned}Q_\alpha(y) &= \sum_{i=1}^{s}\langle\alpha_i w_i,\tilde{\phi}(y+h_i)\rangle,\\ \gamma_\alpha(y) &= \sum_{i=1}^{s}\alpha_i\Delta_{h_i}\tilde{\phi}(y).\end{aligned}$$

Note crucially, that $\deg(\gamma_\alpha)\le d-1$ for all $\alpha\in\mathbb{F}_p^s$, which is what will eventually allow us to apply the induction hypothesis.

It follows from our expression for $\omega_p^{P(y,v)}$ that

$$\begin{aligned}\mathbf{1}[\phi(y)=x] &= \mathbb{E}_{v\in\mathbb{F}_p^k}\omega_p^{\langle v,\phi(y)-x\rangle}\\ &= \mathbb{E}_{v\in\mathbb{F}_p^k}\omega_p^{P(y,v)+\langle v,\psi(y)-x\rangle}\\ &= \sum_{\alpha\in\mathbb{F}_p^s}\widehat{f}(\alpha)\omega_p^{Q_\alpha(y)}\,\mathbb{E}_{v\in\mathbb{F}_p^k}\omega_p^{\langle v,\,(\gamma_\alpha+\psi)(y)-x\rangle}.\end{aligned}$$

Taking $y=E(x)+Z$, we then obtain

$$\begin{aligned}\Pr\big[\phi\big(E(x)+Z\big)=x\big] &= \mathbb{E}_{x,Z}\mathbf{1}\big[\phi(E(x)+Z\big)=x\big]\\ &\le \sum_{\alpha\in\mathbb{F}_p^s}|\widehat{f}(\alpha)|\,\mathbb{E}_{x,Z}\big|\mathbb{E}_{v\in\mathbb{F}_p^k}\omega_p^{\langle v,\,(\gamma_\alpha+\psi)(E(x)+Z)-x\rangle}\big|\\ &\le \Big(\sum_{\alpha\in\mathbb{F}_p^s}|\widehat{f}(\alpha)|\Big)\max_{\alpha\in\mathbb{F}_p^s}\mathbb{E}_{x,Z}\mathbf{1}\big[(\gamma_\alpha+\psi)\big(E(x)+Z\big)=x\big]\\ &\le p^{s/2}\max_{\alpha\in\mathbb{F}_p^s}\Pr\big[(\gamma_\alpha+\psi)\big(E(x)+Z\big)=x\big],\end{aligned}$$

where we have used the Cauchy-Schwarz inequality and Parseval's identity in the last line. Since $\deg(\gamma_\alpha+\psi)\le d-1$ and s ultimately depends only on p, d, ρ and ε, by taking

$$k\ge k_0(p,d,\rho,\varepsilon):=k_0(p,d-1,\rho,\varepsilon\,p^{-s/2})$$

we conclude from the induction hypothesis that

$$\Pr\big[\phi\big(E(x)+Z\big)=x\big]\le\varepsilon$$

in this structured case as well. The theorem follows.

3.6 The high-characteristic setting

The extremely slow decay found in Theorem 3.1.1 makes the result qualitative rather than quantitative. Yet, results from high characteristics seem to indicate room for improvements up to an exponential decay in the message length k. This is done by proving a "high-characteristic" analogue of Theorem 3.1.2 with much better bounds, presented below; as we see no reason to believe a result of this kind has a strong dependence on the characteristic of the finite field considered[8], we believe that a theorem also holds for low-characteristic fields such as $\mathbb{F}_2$.

3.6.1. Theorem (Exponential decay in high characteristic). *For every $d \in \mathbb{N}$ and $\rho \in [0,1)$ there exist constants $C = C(\rho, d)$ and $c = c(\rho, d) > 0$ such that the following holds. Let $p > d$ be a prime, and let n, k be integers with $k \geq p$. Then for every polynomial map $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree at most d and every function $E : \mathbb{F}_p^k \to \mathbb{F}_p^n$ we have*

$$\Pr_{x \in \mathbb{F}_p^k, Z \sim \mathcal{N}_\rho(0)} \big[\phi\big(E(x) + Z\big) = x\big] \leq Ce^{-ck/(\log k)^{d^2}}. \tag{3.5}$$

3.6.2. Remark. The presence of the polylogarithmic term in the exponential in Equation (3.5) is due to a polylogarithmic loss when passing between two distinct notions of tensor rank in our proof of Theorem 3.6.1. It is a widely-believed conjecture in additive combinatorics that these two notions of rank (see Section 3.7) are within a constant multiplicative factor of one another, in which case our proof would give an upper bound of the form Ce^{-ck} for the probability of correct message retrieval, which is optimal.

We use induction on the degree d to prove Theorem 3.6.1 and use the degree-1 case shown in Section 3.2 as the base case. The inductive argument again relies on a structure-versus-randomness dichotomy based on a notion of rank associated with the polynomial map ϕ. The better bounds we obtain stem from the fact that, in the high-characteristic case, one can work with tensors (i.e., multilinear forms) rather than with general polynomial maps. In the quasirandom case of our argument, we can then use a stronger version of the random restriction theorem for the analytic rank of tensors, while in the structured case we use a connection between analytic rank and partition rank of tensors [MZ22].

3.7 Tensors associated to polynomial maps

Given a polynomial map $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree at most d, we can define a $(d+1)$-tensor $T : (\mathbb{F}_p^n)^d \times \mathbb{F}_p^k \to \mathbb{F}_p$ associated to it by

$$T(y_1, \dots, y_d, v) = \langle v, \Delta_{y_1} \cdots \Delta_{y_d} \phi(0) \rangle.$$

[8] While the result is stated in the setting of prime fields $\mathbb{F}_p$, it easily generalizes to the case of non-prime finite fields $\mathbb{F}_q$, with only minor modifications in the proof.

While not immediately obvious, the formula above is indeed linear in each variable separately and thus defines a tensor. This follows from the fact that $\Delta_{y_1} \cdots \Delta_{y_d} \phi$ does not depend on the order of the derivatives, and that the polynomial map $\Delta_{y_1} \cdots \Delta_{y_{d-1}} \phi$ has degree at most 1 (since ϕ has degree at most d); note that, if ψ is a linear map, then $h \mapsto \Delta_h \psi$ is linear in h.

If the characteristic p of the field in strictly higher than the degree d, then we also have the *integration formula*

$$\phi(y) = \frac{1}{d!} T(y, \dots, y, \cdot) + \psi(y) \quad \text{for all } y \in \mathbb{F}_p^n,$$

where y is repeated d times inside T and ψ is a polynomial map of degree at most $d-1$. This follows from the (discrete) Taylor expansion theorem, and allows us to pass back and forth between tensors and polynomial maps.

We will use the following two notions of rank for tensors, originally introduced by Gowers and Wolf [GW11] and by Naslund [Nas20], respectively.

3.7.1. Definition (Tensor analytic rank). Let $X_1, \dots, X_r$ be positive integers and $T : \mathbb{F}_p^{X_1} \times \cdots \times \mathbb{F}_p^{X_r} \to \mathbb{F}_p$ be an r-tensor. The bias of T is defined as

$$\operatorname{bias}(T) = \mathbb{E}_{x_1 \in \mathbb{F}_p^{X_1}, \dots, x_r \in \mathbb{F}_p^{X_r}} \omega_p^{T(x_1, \dots, x_r)}.$$

The bias is always real and positive[9], and the analytic rank of T is defined by

$$\operatorname{arank}(T) = -\log_p \operatorname{bias}(T).$$

3.7.2. Definition (Partition rank). For positive integers $X_1, \dots, X_r$, a nonzero r-tensor $T : \mathbb{F}_p^{X_1} \times \cdots \times \mathbb{F}_p^{X_r} \to \mathbb{F}_p$ is said to have partition rank 1 if there exists a nonempty strict subset $I \subset [r]$ and tensors $S : \prod_{i \in I} \mathbb{F}^{X_i} \to \mathbb{F}$ and $R : \prod_{i \in [r] \setminus I} \mathbb{F}^{X_i} \to \mathbb{F}$ such that T can be factored as $T = SR$. The partition rank of T, denoted $\operatorname{prank}(T)$, is defined as the least $m \in \mathbb{N}$ such that there is a decomposition $T = T_1 + \cdots + T_m$ where each T_i has partition rank 1.

While these two notions of rank are defined in very different ways, it turns out that they are intimately related to each other. Lovett showed that for all tensors T, it holds that $\operatorname{arank}(T) \leq \operatorname{prank}(T)$ [Lov19]. It is a well-known open problem to determine whether a similar inequality holds in the converse direction, up to an absolute multiplicative factor. Moshkovitz and Zhu proved that the relation between these two rank functions is at worst quasilinear [MZ22].

3.7.3. Theorem (Moshkovitz–Zhu). *For every $r \geq 2$ there exists $L_r > 0$ such that for every r-tensor T over any finite field, we have*

$$\operatorname{arank}(T) \leq \operatorname{prank}(T) \leq L_r \operatorname{arank}(T) \log^{r-1}\big(1 + \operatorname{arank}(T)\big). \tag{3.6}$$

[9] It is not hard to show that $\operatorname{bias}(T) = \Pr_{x_1 \in \mathbb{F}_p^{X_1}, \dots, x_{r-1} \in \mathbb{F}_p^{X_{r-1}}} \mathbf{1}\big[T(x_1, \dots, x_{r-1}, \cdot) \equiv 0\big]$.

This result will be an important ingredient in our proof of Theorem 3.6.1; we note that the decay obtained could be improved to Ce^{-ck} if Theorem 3.7.3 were proven without the polylogarithmic factor on the right-hand side of Equation (3.6). Another important ingredient is the following random restriction theorem for tensors [BC24], stated here for the special case of the analytic rank.

3.7.4. Theorem (Tensor random restriction theorem). *For every integer d and every $\sigma \in (0,1]$, there exist constants $C, \kappa > 0$ such that for any order-d tensor T over any field, we have that*

$$\Pr_{I\sim[n]_\sigma}\big[\operatorname{arank}(T_{|I}) \geq \kappa \cdot \operatorname{arank}(T)\big] \geq 1 - Ce^{-\kappa \operatorname{arank}(T)}.$$

3.8 The proof of Theorem 3.6.1

The proof will proceed by induction on the degree of the polynomial map. Recall that Section 3.2 provides the proof for the base case of degree-1 maps.

Let $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ be a polynomial map of degree at most d, with $2 \leq d < p$, and suppose the theorem holds for polynomial maps of degree at most $d-1$. Define the $(d+1)$-tensor $T : (\mathbb{F}_p^n)^d \times \mathbb{F}_p^k \to \mathbb{F}_p$ by

$$T(y_1, \dots, y_d, v) = \langle v,\, \Delta_{y_1} \dots \Delta_{y_d} \phi(0) \rangle.$$

We split the analysis into two cases, depending on whether the analytic rank of T is above or below some cutoff value $r = \Theta\big(k/(\log k)^{d^2}\big)$.

3.8.1 Pseudorandom case

Assume that $\operatorname{arank}(T) \geq r$. We will show that the probability

$$\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi\big(E(x)+Z\big) = x\big] \tag{3.7}$$

decays exponentially in $\operatorname{arank}(T)$ for every $x \in \mathbb{F}_p^n$; we then conclude the pseudorandom case by averaging over all x. Now fix some $x \in \mathbb{F}_p^n$. As before, we write

$$\begin{aligned}\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi\big(E(x)+Z\big) = x\big] &= \mathbb{E}_{I\sim[n]_{1-\rho}} \Pr_{y\in\mathbb{F}_p^I}\big[\phi\big(E(x)+y\big) = x\big] \\ &= \mathbb{E}_{I\sim[n]_{1-\rho}} \mathbb{E}_{y\in\mathbb{F}_p^I, v\in\mathbb{F}_p^k} \omega_p^{\langle v,\, \phi(E(x)+y)-x\rangle}.\end{aligned}$$

Note that we can write $\phi(E(x)+y) - x = \phi(y) + \psi(y)$, where

$$\psi(y) := \Delta_{E(x)}\phi(y) - x$$

has degree at most $d-1$. Using this identity and the triangle inequality, it follows that

$$\begin{aligned}\Pr_{Z\sim\mathcal{N}_\rho(0)}\big[\phi\big(E(x)+Z\big) = x\big] &= \mathbb{E}_{I\sim[n]_{1-\rho}} \mathbb{E}_{y\in\mathbb{F}_p^I, v\in\mathbb{F}_p^k} \omega_p^{\langle v,\, \phi(y)+\psi(y)\rangle} \\ &\leq \mathbb{E}_{I\sim[n]_{1-\rho}} \mathbb{E}_{v\in\mathbb{F}_p^k} \big|\mathbb{E}_{y\in\mathbb{F}_p^I} \omega_p^{\langle v,\, (\phi+\psi)(y)\rangle}\big|.\end{aligned}$$

Repeated applications of the Cauchy-Schwarz inequality (or equivalently, the monotonicity property of the Gowers uniformity norms (Equation (1.17)) shows that, for any fixed $v \in \mathbb{F}_p^k$, $I \subseteq [n]$, we have

$$\left|\mathbb{E}_{y\in\mathbb{F}_p^I} \omega_p^{\langle v,\,(\phi+\psi)(y)\rangle}\right| \leq \left(\mathbb{E}_{y_0,y_1,\ldots,y_d\in\mathbb{F}_p^I} \omega_p^{\langle v,\,\Delta_{y_1}\ldots\Delta_{y_d}(\phi+\psi)(y_0)\rangle}\right)^{1/2^d}.$$

We then conclude that

$$\begin{aligned}
&\Pr_{Z\sim\mathcal{N}_\rho(0)}\left[\phi\big(E(x)+Z\big)=x\right] \\
&\qquad\leq \mathbb{E}_{I\sim[n]_{1-\rho}}\mathbb{E}_{v\in\mathbb{F}_p^k}\left(\mathbb{E}_{y_0,y_1,\ldots,y_d\in\mathbb{F}_p^I} \omega_p^{\langle v,\,\Delta_{y_1}\ldots\Delta_{y_d}(\phi+\psi)(y_0)\rangle}\right)^{1/2^d} \\
&\qquad\leq \mathbb{E}_{I\sim[n]_{1-\rho}}\left(\mathbb{E}_{v\in\mathbb{F}_p^k}\mathbb{E}_{y_0,y_1,\ldots,y_d\in\mathbb{F}_p^I} \omega_p^{\langle v,\,\Delta_{y_1}\ldots\Delta_{y_d}(\phi+\psi)(y_0)\rangle}\right)^{1/2^d},
\end{aligned}$$

where we applied Hölder's inequality once (or, alternatively, Cauchy-Schwarz d further times).

We now need to relate this last expression to the analytic rank of T. Derivating d times a polynomial map of degree at most $d-1$ gives the zero map, and so $\Delta_{y_1}\ldots\Delta_{y_d}\psi(y_0) \equiv 0$. Moreover, since $\deg(\phi) \leq d$, the d-th derivative $\Delta_{y_1}\ldots\Delta_{y_d}\phi$ is a constant map. We conclude that

$$\mathbb{E}_{v\in\mathbb{F}_p^k}\mathbb{E}_{y_0,y_1,\ldots,y_d\in\mathbb{F}_p^I}\omega^{\langle v,\,\Delta_{y_1}\ldots\Delta_{y_d}(\phi+\psi)(y_0)\rangle} = \mathbb{E}_{v\in\mathbb{F}_p^k}\mathbb{E}_{y_1,\ldots,y_d\in\mathbb{F}_p^I}\omega^{\langle v,\,\Delta_{y_1}\ldots\Delta_{y_d}\phi(0)\rangle}.$$

For each $v \in \mathbb{F}_p^k$, let $S(v)$ be the d-tensor given by $T(\cdot,\ldots,\cdot,v)$. Then the above is precisely the bias of the restricted tensor $S(v)_{|I^d}$, averaged over v, which (by definition) equals the average of $p^{-\operatorname{arank}(S(v)_{|I^d})}$. The probability in Equation (3.7) is then bounded from above by

$$\mathbb{E}_{v\in\mathbb{F}_p^k}\mathbb{E}_{I\sim[n]_{1-\rho}}p^{-\operatorname{arank}(S(v)_{|I^d})/2^d}.$$

Theorem 3.7.4 now implies that for some absolute constant $C = C(d,\rho) > 0$, the last quantity is bounded from above by $Cp^{-\operatorname{arank}(T)/C}$.

3.8.2 Structured case

Now we assume that $\operatorname{arank}(T) < r$.

Denote the partition rank of T by $s := \operatorname{prank}(T)$. Theorem 3.7.3 shows that $s \leq L_{d+1}r(\log r)^d$, where L_{d+1} is a universal constant. We can then write

$$T(y_{[d]}, v) = \sum_{i=1}^{s} R_i(y_{I_i})S_i(y_{I_i^c}, v)$$

for some nonempty sets $I_i \subseteq [d]$, $|I_i|$-tensors R_i and $(d-|I_i|+1)$-tensors S_i. Since $d < p$, by Taylor's expansion theorem we have that

$$\phi(y) = \frac{1}{d!}\Delta_y \dots \Delta_y \phi(0) + \psi_0(y), \quad \deg(\psi_0) < d.$$

Define $q_i : \mathbb{F}_p^n \to \mathbb{F}_p$, $\psi_i : \mathbb{F}_p^n \to \mathbb{F}_p^k$, for $i \in [s]$, by

$$q_i(y) = \frac{1}{d!} R_i(y_{I_i}), \quad \langle v, \psi_i(y)\rangle = S_i(y_{I_i^c}, v),$$

and note that $\deg(\psi_i) < d$ for all $i \in [s]$. By the definition of T we conclude that

$$\phi(y) = \psi_0(y) + \sum_{i=1}^{s} q_i(y)\psi_i(y), \quad \text{with } \deg(\psi_i) < d \text{ for } 0 \le i \le s.$$

Let $\mathcal{A} = \{A_1, \dots, A_m\}$ be the partition of $\mathbb{F}_p^n$ given by the level sets of the polynomial map $(q_1, \dots, q_s) : \mathbb{F}_p^n \to \mathbb{F}_p^s$; note that $m \le p^s \le p^{L_{d+1} r(\log r)^d}$. For each $j \in [m]$, ϕ will coincide on A_j with a polynomial map $\psi_{A_j} : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree at most $d-1$; simply substitute the $q_i(y)$ in the formula above by their value on $A_j \in \mathcal{A}$. Define the random events

$$\mathcal{E}_j = \big\{E(x) + Z \in A_j : x \sim \mathcal{U}(\mathbb{F}_p^k),\, Z \sim \mathcal{N}_\rho(0)\big\}, \quad j \in [m].$$

Since these events partition the probability space, it follows that

$$\begin{aligned}
&\Pr_{x\in\mathbb{F}_p^k, Z\sim\mathcal{N}_\rho(0)}\big[\phi\big(E(x)+Z\big) = x\big] \\
&\qquad = \sum_{i=1}^{m} \Pr_{x,Z}\big[\phi\big(E(x)+Z\big) = x \wedge \mathcal{E}_j\big] \\
&\qquad = \sum_{i=1}^{m} \Pr_{x,Z}\big[\psi_{A_j}\big(E(x)+Z\big) = x \wedge \mathcal{E}_j\big] \\
&\qquad \le m \cdot \max_{1\le j\le m} \Pr_{x,Z}\big[\psi_{A_j}\big(E(x)+Z\big) = x\big] \\
&\qquad \le p^{L_{d+1} r(\log r)^d} \cdot \max_{\deg(\psi)<d} \Pr_{x,Z}\big[\psi\big(E(x)+Z\big) = x\big],
\end{aligned}$$

where the last maximum is over all polynomial maps $\psi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree at most $d-1$. By the induction hypothesis we have that this maximum probability is at most $C' e^{-c'k/(\log k)^{(d-1)^2}}$, where $C' = C(d-1,\rho)$ and $c' = c(d-1,\rho)$; we conclude that

$$\begin{aligned}
&\Pr_{x\in\mathbb{F}_p^k, Z\sim\mathcal{N}_\rho(0)}\big[\phi\big(E(x)+Z\big) = x\big] \\
&\qquad \le C' \exp\left((\log p) L_{d+1} r(\log r)^d - \frac{c'k}{(\log k)^{(d-1)^2}}\right).
\end{aligned}$$

Taking

$$r = \frac{c'}{2L_{d+1}} \frac{k}{(\log k)^{d^2}},$$

and using our assumptions $k \geq p$ and $d \geq 2$, we have that

$$\begin{aligned}(\log p) L_{d+1} r (\log r)^d &\leq (\log k) L_{d+1} \frac{c'}{2L_{d+1}} \frac{k}{(\log k)^{d^2}} (\log k)^d \\ &= \frac{c'}{2} \frac{k}{(\log k)^{d^2-d-1}} \\ &\leq \frac{c'}{2} \frac{k}{(\log k)^{(d-1)^2}}.\end{aligned}$$

We conclude that

$$\Pr_{x \in \mathbb{F}_p^k, Z \sim \mathcal{N}_\rho(0)} \left[\phi\big(E(x) + Z\big) = x\right] \leq C' \exp\left(- \frac{c' k}{2(\log k)^{(d-1)^2}} \right)$$

in this case, and the theorem follows.

3.9 Reflections and outlook

This chapter proved that polynomial maps of constant-degree cannot decode any error-correcting code for any positive error rate with constant success probability (Theorem 3.1.2). As a result, any $\mathsf{NC}^0[\oplus]$ circuit could also not do so (Theorem 3.1.1). However, the decay in the theorems is slow, making the results qualitative rather than quantitative. We conjecture that this decay results from the used techniques, and that in general an exponential decay holds (Theorem 3.6.1). We provided evidence for this conjecture by studying the high-characteristic case.

The main strength of Theorem 3.1.2 is that it holds for any code and for any positive error rate. Complementary results are known for restricted classes of codes, and also for when the error rate tends to $1/2$.

A code $E : \mathbb{F}_2^k \to \mathbb{F}_2^n$ is *t-wise independent* if, for any size-t subset of coordinates $S \subseteq [n]$ and a uniformly random $X \in \mathbb{F}_2^k$, the restriction $E(X)_{|S}$ is uniformly distributed over $\mathbb{F}_2^{|S|}$. Many codes have this property; for instance, the dual code of a linear code of distance d is $(d-1)$-wise independent.

Under the same noise model considered in this chapter, Lee and Viola [LV17], using earlier work of Viola [Vio09], showed that $\mathsf{NC}^0[\oplus]$ circuits cannot distinguish a corrupted uniformly random codeword of an $\omega(1)$-wise independent linear code from a uniformly random element of $\mathbb{F}_2^n$. Note that this problem is formally easier than (list) decoding.

However, the result of Lee and Viola does not cover the Hadamard code, as it is not even 3-wise independent. Indeed, the Hadamard code is also easy to

distinguish, as it contains the sub-code $(x_1, x_2, x_1 + x_2)$. Since the parity of these three bits is always zero, the parity under noise is biased towards zero and is therefore easily distinguished from the parity of a random string.

Future work can explore the use of the techniques in this chapter to the situation sketched by Lee and Viola to see how their result extends to a broader class of codes. Additionally, it is interesting to see how our quantitative bounds can be improved, for instance, using the techniques by Lee and Viola.

The techniques used in this chapter use novel ideas from additive combinatorics. These techniques are of greater interest, and it would therefore be interesting to see where else they can be applied.

Finally, an interesting direction for future work is to see if we can prove similar results for other types of circuits. An example is $\mathsf{NC}^0[\oplus]$-circuits that additionally have access to a fixed number of unbounded-fan-in AND- or OR-gates. These circuits are strictly stronger than $\mathsf{NC}^0[\oplus]$, yet do not admit the power that $\mathsf{AC}^0[\oplus]$-circuits have. Another potential class of circuits would be those using bounded-fan-in gates and having non-constant depth $\mathcal{O}(\log \log n)$. In these circuits, before the parity functions, each output bit can only depend on $\mathcal{O}(\log n)$ input bits, instead of on a constant number of bits. The polynomials associated with these classes can have non-constant degree $\mathcal{O}(\log n)$.

Chapter 4

Decoding the Hadamard code with quantum circuits

In this chapter, we present a constant-depth quantum algorithm that can list-decode a heavily corrupted Hadamard codeword. To provide intuition for the algorithm, we start by formulating a non-local game that can list-decode a corrupted Hadamard code. We then turn this non-local game into a quantum circuit. With this quantum circuit, we arrive at a separation between quantum and conventional constant-depth circuits. We furthermore revisit our quantum circuit and show how it can be used to implement majority gates. We conclude by providing directions for quantum circuits for different error-correcting codes.

4.1 Chapter overview

The main result of this chapter is a family of quantum circuits that can retrieve a codeword such that, with high probability, its encoding is close (in Hamming distance) to the corrupted message.

4.1.1. Theorem (Decoding Hadamard code with $\mathsf{QNC}^0[\oplus]$). *There is a family of $\mathsf{QNC}^0[\oplus]$-circuits $(\mathcal{C}_n)_{n\in\mathbb{N}}$ such that the following holds. Let $k \in \mathbb{N}$, $n = 2^k$ and $\varepsilon \in (0, 1/2]$. Then, for any $y \in \mathbb{F}_2^n$ and $x \in \mathbb{F}_2^k$ satisfying $d\left(y, H(x)\right) \leq (\frac{1}{2} - \varepsilon)n$, on input y the circuit $\mathcal{C}_n$ returns x with probability $\Omega(\varepsilon^2)$.*

We note that the bound $\Omega(\varepsilon^2)$ obtained in the theorem is optimal, since in general there can be $\Theta(\varepsilon^{-2})$ messages $x \in \mathbb{F}_2^k$ satisfying $d\big(y, H(x)\big) \leq (\frac{1}{2} - \varepsilon)n$. Note that this bound is only nontrivial if $\varepsilon = \Omega(1/\sqrt{n})$, as there are n possible messages.

The family of quantum circuits of Theorem 4.1.1 originates directly from an n-player non-local game presented as warm-up in Section 4.2. The players employing a quantum strategy share k GHZ states and are given as input coordinates of the corrupted Hadamard codeword.

In Section 4.3, we turn the optimal quantum strategy for this non-local game into a quantum circuit. For the GHZ states, we use a technique by Bene Watts et al. [Ben+19]. We first prepare a *poor man's cat state*: $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$ for some binary vector z, and then correct it to a standard GHZ state.

We input the coordinates of the corrupted Hadamard codeword in the circuit using controlled phase flips. By carefully choosing the controls, we can link the i-th (possibly corrupted) bit with the $|i\rangle$-state. Standard techniques using stacked Toffoli-gates and auxillary qubits require depth $\mathcal{O}(\log \log n)$. Instead, we use an unbounded-fan-in AND-gate

$$|x_1\rangle \dots |x_k\rangle |b\rangle \quad \mapsto \quad |x_1\rangle \dots |x_k\rangle |\mathrm{AND}(x_1, \dots, x_k) \oplus b\rangle$$

to apply the phase flips. Conjugating the inputs of the AND-gate with X-gates ensures we target only the $|i\rangle$-state. We implement this gate using a constant-depth exact OR-routine by Takahashi and Tani [TT13].

We consider the size and depth of the quantum circuit in Section 4.4, and show that the circuits have size $\mathcal{O}(n^2 \log n)$ and depth 65.

Section 4.5 discusses a separation between conventional and quantum circuits based on decoding corrupted error-correcting codes. Note that running the circuit in parallel outputs the desired list of possible messages. We then introduce the *List-Hadamard problem* to prove a separation between $\mathsf{NC}^0[\oplus]$ and $\mathsf{QNC}^0[\oplus]$ for decoding the Hadamard code for any error rate $\delta > 0$, see also Theorem 4.5.3.

In the high-error regime where the parameter δ approaches the information-theoretic limit of $1/2$ (which is relevant for hardness amplification), a stronger separation follows by combining Theorem 4.1.1 with a result of Sudan showing the hardness of noisy decoding by $\mathsf{AC}^0[\oplus]$-circuits, implying a separation between $\mathsf{AC}^0[\oplus]$ and $\mathsf{QNC}^0[\oplus]$. We prove this separation in Section 4.6 and also give a quantum circuit for Majority, based on a quantum circuit for list decoding.

Next, Section 4.7 discusses the extension of the Hadamard circuit to higher field characteristics, similar to how Theorem 3.1.2 extends Theorem 3.1.1.

4.2 Quantum decoding the Hadamard code in a non-local game

Our quantum algorithm is inspired by the analysis of a particular non-local game. In a non-local game, a referee randomly sends questions to a set of players, accord-

ing to a probability distribution known to the players in advance. Then, without communicating with each other, the players individually answer the referee. Finally, the referee determines whether the players win or lose based solely on the questions and answers. The rule used by the referee is known to the players in advance as well. With a (deterministic) conventional strategy, the players decide before the game starts what to answer to each possible question. With a quantum strategy, the players base their answers on the outcomes of local measurements of their respective parts of a shared entangled state. We refer to [Cle+04] for further background on non-local games.

Let $H : \mathbb{F}_2^k \to \mathbb{F}_2^n$ be the Hadamard code and let $\varepsilon \in (0, 1/2)$ be some constant independent of n. We identify the codewords $H(x)$ with functions $\mathbb{F}_2^k \to \mathbb{F}_2$ given by $H(x)(y) = \langle x, y \rangle$. We consider the following n-player non-local game, which we shall refer to as the *Hadamard game*: Each player is labeled uniquely with an element in $\mathbb{F}_2^k$. The referee picks a uniformly chosen message $x \in \mathbb{F}_2^k$ and randomly corrupts the codeword $H(x)$ using the binary symmetric channel with error rate $1/2 - \varepsilon$, resulting in a function $c : \mathbb{F}_2^k \to \mathbb{F}_2$. He then sends player y the value $c(y)$. The players each return a string in $\mathbb{F}_2^k$ and they win the game if the sum of their answers equals x.

Let the n players share an n-partite GHZ state of local dimension 2^k:

$$\frac{1}{\sqrt{n}} \sum_{y \in \mathbb{F}_2^k} |y\rangle \otimes \ldots \otimes |y\rangle .$$

Consider the following strategy inspired by the famous Bernstein-Vazirani algorithm [BV97]:

1. Upon receiving input $c(y)$, player y applies a conditional phase flip on their part of the shared state:

$$|z\rangle \mapsto \begin{cases} (-1)^{c(y)} |z\rangle & \text{if } z = y \\ |z\rangle & \text{otherwise.} \end{cases} \tag{4.1}$$

2. Each player applies a k-qubit Hadamard gate to their local register;
3. Each player measures their local register in the computational basis;
4. Each player returns the measurement results.

4.2.1. Theorem. *The players win the Hadamard game with probability $\Omega(\varepsilon^2)$, if they follow the strategy outlined above.*

Proof: Once all players have applied their conditional phase flips, they share the state

$$\frac{1}{\sqrt{n}} \sum_{y} (-1)^{c(y)} |y\rangle \otimes \ldots \otimes |y\rangle . \tag{4.2}$$

The k-qubit Hadamard gates map this state to

$$n^{-(n+1)/2} \sum_{y \in \mathbb{F}_2^k} \sum_{b_1,\ldots,b_n \in \mathbb{F}_2^k} (-1)^{c(y)} (-1)^{\langle y, b_1+\cdots+b_n \rangle} |b_1\rangle \otimes \ldots \otimes |b_n\rangle .$$

The probability that the measurement results sum to some string $z \in \mathbb{F}_2^k$ is now given by

$$\begin{aligned} \Pr\left[\sum_{i=1}^{n} b_i = z\right] &= \frac{1}{n^{(n+1)}} \sum_{b_1+\cdots+b_n=z} \left| \sum_{y \in \mathbb{F}_2^k} (-1)^{c(y)+\langle y,z \rangle} \right|^2 \\ &= \left(1 - 2\frac{d\big(c, H(z)\big)}{n}\right)^2 . \end{aligned} \tag{4.3}$$

It follows from the Chernoff bound [HR90] that for fixed $x \in \mathbb{F}_2^k$ and the random c obtained by corrupting the codeword $H(x)$,

$$\Pr\left[\frac{d\big(c, H(x)\big)}{n} \geq \frac{1-\varepsilon}{2}\right] \leq \exp(-C\varepsilon^2 n).$$

Hence, by the union bound, for fixed $\varepsilon \in (0, 1/2)$, the players win with probability at least $C\varepsilon^2$, where the probability is taken over the message x, the noise corrupting the codeword $H(x)$ and the measurements done by the players. □

Note that this strategy succeeds with probability $C\varepsilon^2$ for *every* x and whenever at most any $(1/2-\varepsilon)$-fraction of the coordinates of $H(x)$ are flipped, independent of the error-model.

4.3 Details of quantum algorithm

Below we give more details on how to generate the GHZ states, as well as on how to implement the controlled phase gates.

4.3.1 Generating GHZ states

Preparing the GHZ state

$$|GHZ\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right)$$

requires depth $\Omega(\log n)$ given an all-to-all connectivity. This exceeds the constant-depth requirement of our circuits. Instead, we generate an intermediate state and then use the conventional parity gates to implement correction terms to correct the intermediate state in a GHZ state.

Starting with $2n-1$ qubits, we apply Hadamard gates to all $2i-1$ qubits for $i \in [n]$. Next, we apply two layers of parallel CNOT-gates: In the first layer, from qubit $2i-1$ to qubit $2i$, for $i \in [n-1]$; In the second layer, from qubit $2i+1$ to qubit $2i$, for $i \in [n-1]$. Next, all even-numbered qubits are measured, giving measurement results d_i, $i \in [n-1]$. The resulting poor man's cat state is then corrected to a GHZ state by applying an X-gate to qubit i based on a prefix sum computation, that is, an X-gate is applied to qubit $2i-1$ if and only if $\sum_{j=1}^{i-1} d_j \bmod 2 \equiv 1$ for $i \in [n]$. Figure 4.1 shows the quantum circuit to generate a 3-qubit GHZ state.

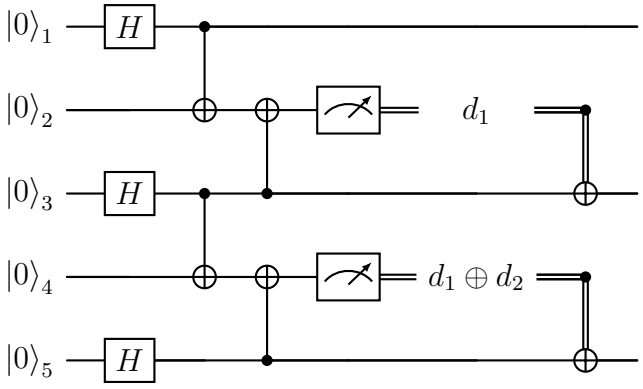


Figure 4.1: The quantum circuit to generate a 3-qubit GHZ state. First, we prepare a poor man's cat state $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$ with each $z \in \mathbb{F}_2^3$ equally likely to be found. The parity gates compute a prefix sum on the measurement results d_1 and d_2 and determine if a qubit has to be flipped to obtain the GHZ state.

With this construction, the depth of the circuit remains constant, independent of n. Also note that this circuit corresponds to a Clifford ladder, which is discussed later in Section 7.3.

4.3.2 Quantum fanout gate

We will use a quantum fanout gate to implement the controlled phase flips. For our construction, we will use ideas from distributed quantum computing. Eisert et al. and Yimsiriwattana and Lomonaco introduced a non-local CNOT-gate, using single-qubit gates, CNOT-gates and shared GHZ states [Eis+00; YL04]. We extend their construction to a quantum fanout gate, by using GHZ states shared by more parties. A circuit for the quantum fanout gate for $n = 3$ is given in Figure 4.2. The last Z-gate is applied only if the parity of all measurement results equals 1. The time steps denote which gates can be applied in parallel.

Later, Pham and Svore introduced another way to implement a quantum fanout

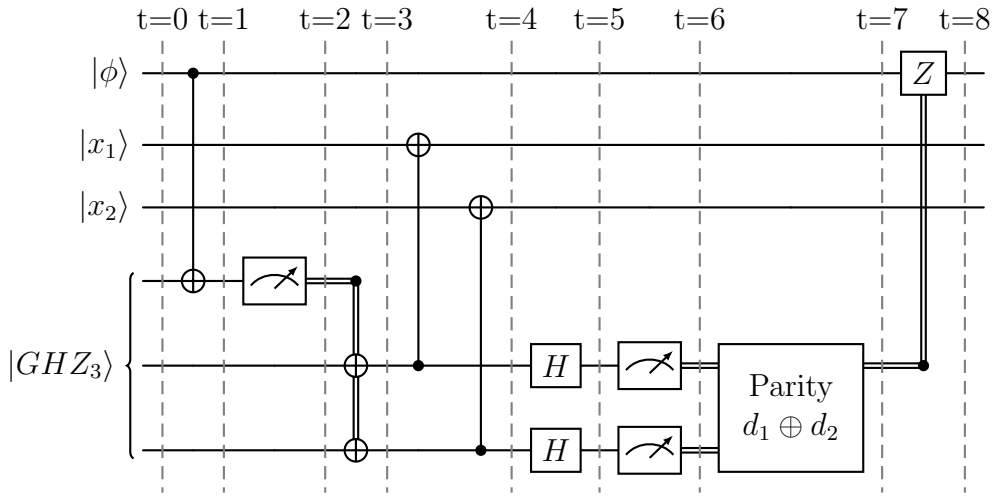


Figure 4.2: Implementation of a quantum fanout gate with one control qubit $|\phi\rangle$ and two target qubits $|x_1\rangle$ and $|x_2\rangle$. Only single- and two-qubit gates and conventional parity gates are used. The bottom three qubits are in the GHZ_3 state. The dotted lines denote time steps and which gates can be applied in parallel.

gate [PS13, Figure 4]. At first sight, their circuit seems to have a shallower depth. Yet, their circuit uses Bell-basis measurements that decompose into a CNOT-gate, a Hadamard gate and two standard basis measurements. Furthermore, the state produced corresponds to a GHZ state up to Pauli corrections, which depend on the measurement outcomes.

Even though both approaches implement a quantum fanout gate, we use the approach inspired by Eisert et al. and Yimsiriwattana and Lomonaco for its simplicity and its broader applicability in, for instance, distributed quantum computing.

4.3.1. Lemma. *The circuit of Figure 4.2, extended to arbitrary n, implements a quantum fanout gate.*

Proof: Let $|x\rangle$ be an n-qubit computational basis state and $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ be any single qubit quantum state. We will prove that the circuit implements the quantum fanout gate on the state $|\phi\rangle|x\rangle$. The lemma then follows by linearity of the operations.

The action of the quantum fanout gate on the quantum state is given by

$$|\phi\rangle|x\rangle \overset{\text{Fanout}}{\mapsto} \alpha|0\rangle|x\rangle + \beta|1\rangle X^{\otimes n}|x\rangle = \alpha|0\rangle|x\rangle + \beta|1\rangle|\bar{x}\rangle\,, \tag{4.4}$$

where $|\bar{x}\rangle$ is the computational basis state $|x\rangle$ with all qubits flipped.

The circuit indeed implements this map (when generalized to arbitrary n). As-

sume we have a GHZ_{n+1} state, up to a normalization factor of $1/\sqrt{2}$, we have:

$$\begin{aligned}
&\big[\alpha\,|0\rangle + \beta\,|1\rangle\big]\,|x\rangle \otimes \big[\,|00\cdots0\rangle + |11\cdots1\rangle\,\big] \\
&\quad\overset{(1)}{\longmapsto} \alpha\,|0\rangle\,|x\rangle \otimes \big[\,|00\cdots0\rangle + |11\cdots1\rangle\,\big] + \beta\,|1\rangle\,|x\rangle \otimes \big[\,|10\cdots0\rangle + |01\cdots1\rangle\,\big] \\
&\quad\overset{(2)}{\longmapsto} \alpha\,|0\rangle\,|x\rangle\,|d_0 0\cdots0\rangle + \beta\,|1\rangle\,|x\rangle\,|d_0 1\cdots1\rangle \\
&\quad\overset{(3)}{\longmapsto} \alpha\,|0\rangle\,|x\rangle\,|d_0 0\cdots0\rangle + \beta\,|1\rangle\,X^{\otimes n}\,|x\rangle\,|d_0 1\cdots1\rangle \\
&\quad\overset{(4)}{\longmapsto} \frac{1}{2^{n-1}} \sum_{d\in\mathbb{F}_2^n} \big[\alpha\,|0\rangle\,|x\rangle + \beta(-1)^{d_1+\ldots+d_n}\,|1\rangle\,X^{\otimes n}\,|x\rangle\,\big]\,|d_0 d_1\ldots d_n\rangle \\
&\quad\overset{(5)}{\longmapsto} \alpha\,|0\rangle\,|x\rangle\,|d_0 d_1\ldots d_n\rangle + (-1)^{d_1+\ldots+d_n}\beta\,|1\rangle\,X^{\otimes n}\,|x\rangle\,|d_0 d_1\ldots d_n\rangle \\
&\quad\overset{(6)}{\longmapsto} \big[\alpha\,|0\rangle\,|x\rangle + \beta\,|1\rangle\,|\bar{x}\rangle\,\big]\,|d_0 d_1\ldots d_n\rangle\,.
\end{aligned}$$

Step (1): Perform a CNOT-gate from the control qubit to the first qubit of the GHZ state; Step (2): Measure that qubit, with outcome d_0, and apply an X-gate to the remaining n qubits of the GHZ state if $d_0 = 1$; Step (3): Perform n parallel CNOT-gates between the $i+1$-th qubit of the GHZ state and the i-th target qubit; Steps (4) and (5): Apply Hadamard gates to each unmeasured qubit of the GHZ state and subsequently measure it; and, Step (6): Compute the parity $d_1 \oplus \ldots \oplus d_n$ and apply a Z-gate to the control qubit if this parity equals one. The circuit thus implements the quantum fanout gate as desired. □

4.3.3 Applying the phase flip

We apply the conditional phase flip by first conjugation the input qubits corresponding to the zeros in the binary representation of the index with X-gates and then computing the AND of these qubits.

As we have the identity $\mathrm{AND}(z_1,\ldots,z_k) = \neg\mathrm{OR}(\neg z_1,\ldots,\neg z_k)$, we can use the exact OR-gate by Takahashi and Tani [TT13, Lemma 1] to implement the phase flip. They used the Fourier inversion formula of Equation (1.11), to rewrite the OR as the sum of the parities of all nonempty subsets of inputs. Their method uses single- and two-qubit gates, as well as quantum fanout gates.

We can use quantum fanout gates to compute the parity of all subsets in parallel. Via a phase kickback trick, implemented via a fanout gate combined with a Hadamard gate, we have computed the OR in an auxiliary qubit. We use that auxiliary qubit to apply the phase flip in the GHZ state.

4.4 Circuit complexity

We now count all single-qubit gates, CNOT-gates and unbounded-fan-in parity gates used in the circuit to determine its size and width. In the circuit, we will

also use controlled-R_Z-gates, which we can implement in depth four, using two CNOT-gates and three single-qubits gates [NC10, Corollary 4.2].

4.4.1. Lemma. *The circuit for decoding the Hadamard code has size* $\mathcal{O}(n^2 \log n)$ *and depth* 65*.*

Proof: We break down the steps of the previous section and count the size and depth. Generating n-qubit GHZ states requires $2n-1$ qubits and depth 6. Each of the GHZ states can be prepared in parallel.

The quantum fanout gates used require depth 8. The GHZ states used to implement the quantum fanout gate can be initialized at the same time with the GHZ states used for the initial superposition.

The conditional phase gates can be applied in parallel. Hence, for the depth, we consider only a single instance for index i, which requires the operations:

1. Apply X-gates corresponding to the zeros in the binary representation of index i and at the same time apply an X-gate to an auxillary qubit;
2. Compute the OR of the input in this auxilliary qubit;
3. Apply a Z-gate to the auxillary qubit, conditioned on the input bit $c(i)$;
4. Uncompute the OR of the input;
5. Apply X-gates corresponding to the zeros in the binary representation of index i and at the same time apply an X-gate to an auxillary qubit.

The odd steps correspond to single qubit gates, and hence add 1 to the circuit depth. The second and fourth step apply the circuit for the OR-gate, which has depth 28 [TT13, Lemma 2]. In the circuit, auxiliary GHZ states are used, which can be prepared in parallel with the initial uniform superposition.

1. For every of the k local qubits part of the GHZ state, apply a fanout gate from that qubit to $n-1$ auxiliary qubits;
2. Compute the parity of all possible nonempty subsets of the k indices in parallel using the quantum fanout gate, conjugated by Hadamard gates. The conjugation by Hadamard gates can be integrated in the circuit for the quantum fanout gates and therefore does not increase the circuit depth;
3. Apply controlled-R_Z-gates from the parity computation circuit to a GHZ state;
4. Apply a fanout gate on the auxiliary GHZ state to reduce it to a single bit;
5. Apply a Hadamard gate for a phase kick-back into the target qubit.

As the depth of the fanout gate is 8, the depth of the OR-gate is $3 \cdot 8 + 4 = 28$. This gives a total circuit depth of 65.

For input size k, the size of circuit for applying a conditional phase gate is $\mathcal{O}(k2^k) = \mathcal{O}(n \log n)$. As we have to apply n conditional phase gates in parallel, the overall circuit size is $\mathcal{O}(n^2 \log n)$. □

We can use an OR-reduction by Høyer and Špalek to further reduce the circuit size [HŠ05, Lemma 5.1]. This OR-reduction uses a $\mathcal{O}(k \log k)$-sized constant-depth quantum circuit to prepare a quantum state on $\lceil \log(k+1) \rceil$ qubits, such that the OR on these $\lceil \log(k+1) \rceil$ qubits evaluates to the same value as the OR on the original k qubits.

4.4.2. Corollary. *There exists a quantum circuit for decoding the Hadamard code of size* $\mathcal{O}(n \log n \log\log n)$ *and depth* 103.

Proof: The result follows by noting that the OR-reduction has depth 19 and counting the gates of the exact OR-routine by Takahashi and Tani. □

Note that we can use a fanout gate to simplify the proof of Theorem 4.2.1 slightly: First, apply a fanout gate to map the state of Equation (4.2) to obtain

$$\frac{1}{\sqrt{n}} \sum_y (-1)^{c(y)} |y\rangle |0\rangle^{\otimes n-1}, \tag{4.5}$$

Next, apply Hadamard gates only to the first register. A final measurement of the first register then directly gives a bitstring according to the probability distribution. For the Hadamard code, this approach only simplifies the proof; for other codes, this step is necessary for correct decoding.

4.5 Separating $\mathsf{NC}^0[\oplus]$ from $\mathsf{QNC}^0[\oplus]$

In this section we combine our conventional and quantum results, Theorem 3.1.2 and Theorem 4.1.1 respectively, to show a separation between $\mathsf{NC}^0[\oplus]$, $\mathsf{AC}^0[\oplus]$, and $\mathsf{QNC}^0[\oplus]$. Note that our quantum circuit outputs a single possible message, whereas the conventional circuit returns a list. This requires minor changes and gives the following theorem.

4.5.1. Corollary. *There is a family of* $\mathsf{QNC}^0[\oplus]$*-circuits* $(\mathcal{C}_n)_{n\in\mathbb{N}}$ *such that the following holds. Let* $k \in \mathbb{N}$, $n = 2^k$ *and* $\varepsilon \in [1/\sqrt{n}, 1/2]$. *Then, on any input* $y \in \mathbb{F}_2^n$, *with probability* $1 - \mathcal{O}(\varepsilon)$ *the circuit* $\mathcal{C}_n$ *returns a list* $L(y)$ *of size* $\mathcal{O}(\varepsilon^{-2} \log(1/\varepsilon))$ *which contains every* $x \in \mathbb{F}_2^k$ *with* $d\big(y, H(x)\big) \leq (\frac{1}{2} - \varepsilon)n$.

Proof: For a large enough constant $C > 0$, consider $C\varepsilon^{-2} \log(1/\varepsilon)$ parallel instances of the circuit from Theorem 4.1.1. This gives a list $L(y)$ of the claimed size such that any message $x \in \mathbb{F}_2^k$ satisfying $d\big(y, H(x)\big) \leq (\frac{1}{2} - \varepsilon)n$ appears in $L(y)$ with probability at least $1 - \varepsilon^3$. Since there are at most $O(1/\varepsilon^2)$ such messages, it follows from the union bound that with probability at least $1 - \mathcal{O}(\varepsilon)$ every such message appears in $L(y)$. □

4.5.2. Remark. Note that the circuits obtained in this corollary also output several messages whose codewords differ from the input y in more than $(\frac{1}{2} - \varepsilon)n$ coordinates; this differs from the usual notion of the list decoding problem, which aims to output a list of all messages $x \in \mathbb{F}_2^k$ with $d\big(y, H(x)\big) \leq (\frac{1}{2} - \varepsilon)n$ and none other. One can also solve the usual list-decoding problem for the Hadamard code using $\mathsf{QNC}^0[\oplus]$-circuits, by making use of Majority gates (and more general threshold gates) to prune the obtained list (see also Section 4.6).

As a consequence we arrive at the main result of this chapter: List-decoding the Hadamard code separates the classes $\mathsf{NC}^0[\oplus]$ and $\mathsf{QNC}^0[\oplus]$. This result thus gives a quantum advantage for a problem appearing naturally.

In the high-error setting, where the error parameter δ approaches the information-theoretic limit of $1/2$ (which is relevant for hardness amplification), a stronger separation follows by combining Theorem 4.1.1 with a result of Sudan showing hardness of noisy decoding by $\mathsf{AC}^0[\oplus]$-circuits. Stating this problem requires us to consider a slightly different problem:

List-Hadamard problem: Let $\varepsilon : \mathbb{N} \to (0, 1]$ be a function. For each dyadic number $n = 2^k$ we define the problem $\mathrm{LH}_n(\varepsilon)$ as follows: given $y \in \mathbb{F}_2^n$, output a list of at most $n/4$ elements in $\mathbb{F}_2^k$ that contains every $x \in \mathbb{F}_2^k$ satisfying $d\big(y, H(x)\big) \leq \big(\frac{1}{2} - \varepsilon(n)\big)n$.

In the next section, we discuss the hardness of the list-Hadamard problem and its implications further. For now, it gives us the tools to state the most general form of our quantum advantage result:

4.5.3. Theorem (Quantum-vs-conventional separation). *For every constant $\delta \in (0, \frac{1}{2})$, list decoding the Hadamard code with error parameter δ separates $\mathsf{QNC}^0[\oplus]$ from $\mathsf{NC}^0[\oplus]$. Moreover, for any $(\log n)/\sqrt{n} \leq \varepsilon(n) \leq 1/(\log n)^{\omega(1)}$, the list-Hadamard problem $\mathrm{LH}_n(\varepsilon)$ separates $\mathsf{QNC}^0[\oplus]$ from $\mathsf{AC}^0[\oplus]$.*

4.6 A quantum circuit for Majority

This section shows how to obtain a $\mathsf{QNC}^0[\oplus]$-circuit that computes Majority, which uses our $\mathsf{QNC}^0[\oplus]$-circuit for list decoding the Hadamard code. As a corollary, we prove the second statement of Theorem 4.5.3.

We first show the reduction from list decoding to Majority by introducing a new problem called IsBal and then transform that into a $\mathsf{QNC}^0[\oplus]$-circuit for Majority.

4.6.1 From list decoding to Majority

Sudan (see [Vio06, Section 6.2]) showed that list decoding with error parameter $1/2 - \varepsilon$ requires probabilistic $\mathsf{AC}^0[\oplus]$-circuits to have size $\exp(\mathrm{poly}(1/\varepsilon))$. The

hardness of the list-Hadamard problem will follow as a corollary. For concreteness, we state his result restricted to the Hadamard code; as can be easily seen from its proof, one could instead consider any other error-correcting code.

4.6.1. Theorem (List-Hadamard implies Majority). *Let $\mathcal{C}$ be a probabilistic circuit that solves the list-Hadamard problem* $\mathrm{LH}_n(\varepsilon)$ *with probability at least* $3/4$*. Then there exists a (deterministic) oracular* AC^0*-circuit* $\mathcal{D}$ *of size* $\mathrm{poly}(n, 1/\varepsilon)$ *that, when given oracle access to* $\mathcal{C}$ *and the ability to fix its random bits, computes Majority on* $\Omega(1/\varepsilon)$ *bits.*

As a corollary, the circuit lower bound for Majority due to Razborov [Raz87] and Smolensky [Smo87] gives the following (known) hardness result for list decoding the Hadamard code.

4.6.2. Corollary (Hardness of list-Hadamard). *If* $\varepsilon(n) \leq 1/(\log n)^{\omega(1)}$*, then the list-Hadamard problem* $\mathrm{LH}_n(\varepsilon)$ *cannot be solved by a probabilistic* $\mathsf{AC}^0[\oplus]$*-circuit with probability* $\Omega(1)$*.*

Combining this corollary with our $\mathsf{QNC}^0[\oplus]$-circuits for list-Hadamard given in Corollary 4.5.1, we obtain the second separation of complexity classes stated in Theorem 4.5.3.

To prove Theorem 4.6.1, let Maj_t denote the Majority function on t bits. We first introduce a promise problem called IsBal_t, which asks to determine whether a given binary string is balanced. We then show that a (probabilistic) circuit solving IsBal_t can be turned into a deterministic circuit computing Maj_t. Finally, we show how a circuit for $\mathrm{LH}_n(\varepsilon)$ can be used to solve IsBal_t for $t = \Omega(1/\varepsilon)$.

4.6.3. Definition (The IsBal_t problem). For an even positive integer t, define $\mathrm{IsBal}_t : \{x \in \mathbb{F}_2^t : |x| \leq t/2\} \to \mathbb{F}_2$ by

$$\mathrm{IsBal}_t(x) = \begin{cases} 1 & \text{if } |x| = t/2, \\ 0 & \text{otherwise.} \end{cases}$$

Given an arbitrary $x \in \mathbb{F}_2^t$, define the IsBal_t problem as returning $\mathrm{IsBal}_t(x)$ if $|x| \leq t/2$, and an arbitrary bit otherwise.

A probabilistic circuit for the IsBal_t problem takes in a string x and random coin tosses u. We say that a probabilistic circuit solves the IsBal_t problem if the probability it correctly outputs 1 on balanced inputs is at least $2/3$, and the probability it correctly outputs 0 on unbalanced inputs is at least $1/2$, both taken over the random coin tosses. Next, we have the following derandomization lemma to obtain a deterministic circuit from a probabilistic one.

4.6.4. Lemma (Derandomization lemma). *Let $\mathcal{C}$ be a probabilistic circuit that solves* IsBal_t *with probability at least* $2/3$ *for every input. There exists a deterministic oracle* AC^0*-circuit $\mathcal{C}'$ that, when given oracle access to $\mathcal{C}$ and the ability to fix its random bits, solves* IsBal_t*.*

Proof: For some large enough constant $c \in \mathbb{N}$, consider ct parallel instances of $\mathcal{C}$. It follows from the Chernoff bound that, for any fixed $x \in \mathbb{F}_2^t$ given to all of these instances, with probability $1 - \exp(-10\,t)$ at least 55% of the instances solves the IsBal_t problem on input x.

By the union bound, one can fix the randomness in the instances of $\mathcal{C}$ in order to get a deterministic conventional circuit that, for every input $x \in \mathbb{F}_2^t$ with $|x| \leq t/2$, returns a ct-bitstring whose Hamming weight is at least $0.55t$ if $\mathrm{IsBal}_t(x) = 1$ and at most $0.45t$ if $\mathrm{IsBal}_t(x) = 0$. Distinguishing these two types of strings is known as the approximate majority problem, for which there is an AC^0-circuit [Ajt83]. Combining these circuits gives the result. □

We now show how a deterministic circuit that solves IsBal_t can be used to compute Maj_t.

4.6.5. Lemma. *Let $\mathcal{C}$ be a deterministic circuit for* IsBal_t*. There exists an oracle* AC^0*-circuit $\mathcal{D}$ that, given oracle access to $\mathcal{C}$, computes* Maj_t*.*

Proof: For $x \in \mathbb{F}_2^t$ and $i \in \{0, \ldots, t\}$, define x_i as the string x with the first i bits set to zero and the rest of the bits equal to those of x. So, for instance, $x_0 = x$ and x_t is the all-zeros string. Let $\mathcal{D}$ be the circuit that runs $t+1$ parallel instances of $\mathcal{C}$ with inputs $x_0, x_1, \ldots, x_t$, respectively, and returns the OR of the $t+1$ outputs.

We claim that $\mathcal{D}$ computes Maj_t. Indeed, if x has fewer than $t/2$ ones then $\mathcal{C}$ returns 0 for each input x_i, as the number of ones only decreases with i. If x has at least $t/2$ ones, then $\mathcal{C}$ returns 1 for at least one i, since x_0 has at least $t/2$ ones, whereas x_t is the all-zeros string. This completes the proof. □

Towards turning a circuit $\mathcal{C}$ for $\mathrm{LH}_n(\varepsilon)$ into a circuit for IsBal_t, we associate with each input $x \in \mathbb{F}_2^t$ to IsBal_t a random error vector N_x over $\mathbb{F}_2^n$ as follows: independently, each coordinate of N_x is a uniformly random entry of x. In particular, for balanced x, the error vector N_x will correspond to an error rate of $1/2$ and we refer to it as $N_{1/2}$. The next lemma shows that there exists some message $m \in \mathbb{F}_2^k$ that has small probability of recovery by $\mathcal{C}$ under the error vector $N_{1/2}$.

4.6.6. Lemma. *Let $\mathcal{C}$ be a probabilistic circuit that, on input $y \in \mathbb{F}_2^n$, returns a (random) list $L(y) \subseteq \mathbb{F}_2^k$ of at most $2^k/4$ elements. Then there exists $m \in \mathbb{F}_2^k$ such that*

$$\Pr[m \in L(H(m) + N_{1/2})] \leq 1/4, \tag{4.6}$$

where the probability is taken over L and $N_{1/2}$.

Proof: Note that, for any $y \in \mathbb{F}_2^n$, the vector $y + N_{1/2}$ is uniformly distributed over $\mathbb{F}_2^n$; in particular, it has the same distribution as $N_{1/2}$. Let $M \in \mathbb{F}_2^k$ be a uniformly distributed random element. Then, by independence of M, $L(y)$

and $N_{1/2}$, get that

$$\begin{aligned}\Pr_{M,L,N_{1/2}}[M \in L(H(M) + N_{1/2})] &= \Pr_{M,L,N_{1/2}}[M \in L(N_{1/2})] \\ &\leq \frac{1}{2^k}\,\mathbb{E}_{L,N_{1/2}}\,|L(N_{1/2})| \\ &\leq 1/4.\end{aligned}$$

Hence, there exists a value m of M such that Equation (4.6) holds. □

Finally, we prove that the circuit $\mathcal{C}$ in Theorem 4.6.1 can solve IsBal_t.

4.6.7. Lemma. *Let $\mathcal{C}$ be a probabilistic circuit as in Theorem 4.6.1. There exists a probabilistic oracle AC^0-circuit $\mathcal{D}$ of size $\mathrm{poly}(n, 1/\varepsilon)$ that, when given oracle access to $\mathcal{C}$, solves IsBal_t with probability at least 3/4 for $t = \Omega(1/\varepsilon)$.*

Proof: We may assume without loss of generality that $\varepsilon \leq 1/4$. Let $\delta \in [\varepsilon, 1/4]$ be minimized such that $t = 1/(2\delta)$ is an even integer; note that, since $\delta \geq \varepsilon$, the circuit $\mathcal{C}$ also solves $\mathrm{LH}_n(\delta)$ with probability at least 3/4. Fix a message m as in Lemma 4.6.6, and let $x \in \mathbb{F}_2^t$ be some string to serve as input to $\mathcal{D}$.

The circuit $\mathcal{D}$ has three layers. The first layer has the string $H(m)$ hardwired into it and uses n independent uniform samples to the coordinates of x to compute the random string $H(m)+N_x$. This layer is a probabilistic circuit using n parallel two-bit XOR-gates. The second layer consists of the circuit $\mathcal{C}$, which produces a random list $L(H(m) + N_x)$ of size at most $n/4$. The third layer consists of an AC^0-circuit of size $\mathrm{poly}(n)$ that returns 0 if and only if $m \in L(H(m) + N_x)$. This can be done by checking equality between m and the $\mathcal{O}(n)$ elements of the list. We claim that this solves IsBal_t.

If x is balanced then it follows from Lemma 4.6.6 that $\mathcal{D}$ correctly returns 1 with probability at least 3/4. If x has Hamming weight strictly less than $t/2$, then each coordinate of N_x is 1 with probability at most $1/2 - 1/t = 1/2 - 2\delta$. By the Chernoff bound, N_x has Hamming weight at most $(1/2 - \delta)n$ with probability $1 - \exp(-\Omega(\delta^2 n))$. Hence, the properties of the circuit $\mathcal{C}$ imply that in this case $\mathcal{D}$ correctly outputs 0 with probability at least 3/4. □

Theorem 4.6.1 now follows directly by combining Lemmas 4.6.4, 4.6.5 and 4.6.7.

4.6.2 Obtaining a quantum circuit

In the previous section we discussed conventional circuits for Majority using list-decoding circuits. Now we sketch how the above proof can be used to turn our $\mathsf{QNC}^0[\oplus]$-circuit for decoding the Hadamard code into one that computes Majority with polynomially small error.

Let $\varepsilon = n^{-1/4}$ and let $\mathcal{C}$ be the circuit from Corollary 4.5.1. Since $\mathcal{C}$ returns lists of size at most $n^{3/4}$, a stronger version of Lemma 4.6.6 holds, where the probability

of Equation (4.6) – taken additionally over the measurement outcomes of $\mathcal{C}$ – is bounded from above by $n^{3/4}/2^k = n^{-1/4}$.

The proof of Lemma 4.6.7 then gives an oracle $\mathsf{QNC}^0[\oplus]$-circuit $\mathcal{D}$ of size $\mathrm{poly}(n)$ that, given oracle access to $\mathcal{C}$, solves IsBal_t with probability $1 - \mathcal{O}(n^{-1/4})$ for $t = \Omega(n^{1/4})$. Here, the AC^0-circuit used to check membership of m can be replaced with our $\mathsf{QNC}^0[\oplus]$-circuit for the OR-function (see above) applied to the entrywise sum of m with each element in the list.

Now let $t' = \lfloor n^{1/8} \rfloor$, and note that the same circuit $\mathcal{D}$ above can be used to solve $\mathrm{IsBal}_{t'}$ with probability $1 - \mathcal{O}(n^{-1/4})$: it suffices to pad the input with zeros and ones in the same number until we have a string of the correct size. Finally, with the proof of Lemma 4.6.5 and the union bound we obtain a $\mathsf{QNC}^0[\oplus]$-circuit $\mathcal{D}'$ that, given oracle access to $\mathcal{D}$, solves $\mathrm{Maj}_{t'}$ with probability $1 - \mathcal{O}(n^{-1/8})$. We again use our $\mathsf{QNC}^0[\oplus]$-circuit for the OR-function as discussed in Section 4.3.

4.7 Hadamard code for higher field characteristics

In this section, we extend the $\mathsf{QNC}^0[\oplus]$-circuit for decoding the Hadamard code to higher field characteristics. Before doing so, we first briefly discuss quantum gates for higher characteristics.

4.7.1 Quantum gates for higher field characteristics

Most definitions for qubits generalize directly to multilevel systems. The p-level generalization of a qubit, a qudit, has computational basis states $|0\rangle, \ldots, |p\rangle$. These form the basis for the Hilbert space $\mathcal{H} = \mathbb{C}^p$. The quantum gates for qubits also generalize to qudits (see also, for example, [Wan+20]). Let $+_p$ denote addition modulo p, then we have the following quantum gates:

$$X_p : |i\rangle \mapsto |i +_p 1\rangle \qquad Z_p : |i\rangle \mapsto \omega_p^i |i\rangle$$

$$R_{Z,p}(\theta) : |i\rangle \mapsto \omega_p^{i\theta} |i\rangle \qquad H_p : |i\rangle \mapsto \frac{1}{\sqrt{p}} \sum_{j \in \mathbb{F}_p} \omega_p^{ij} |j\rangle$$

$$CNOT_p : |x\rangle |y\rangle \mapsto |x\rangle |x +_p y\rangle \qquad QFT_p : |x\rangle \mapsto \frac{1}{\sqrt{p^n}} \sum_{y \in \mathbb{F}_p^n} \omega_p^{\langle x,y \rangle} |y\rangle$$

4.7.2 Hadamard code for higher field characteristics

The Hadamard code over larger prime fields is similar to the definition used in previous sections over a binary field. Let $n = p^k$ for some prime p. The Hadamard

code over $\mathbb{F}_p^k$ is then given by

$$H : \mathbb{F}_p^k \to \mathbb{F}_p^n, \qquad x \mapsto \langle i, x \rangle_{i \in \mathbb{F}_p^k},$$

where the inner product is now taken modulo p.

A quantum circuit for list decoding the Hadamard code over $\mathbb{F}_p^n$ looks similar to the circuit exposed in Section 4.3, with the key difference that we use quantum gates that operate on qudits instead of qubits. In the noiseless case we have:

$$\begin{aligned}
|0\rangle &\xrightarrow{(1)} \frac{1}{\sqrt{p^k}} \sum_{i \in \mathbb{F}_p^k} |i\rangle \\
&\xrightarrow{(2)} \frac{1}{\sqrt{p^k}} \sum_{i \in \mathbb{F}_p^k} \omega_p^{\langle i,x \rangle} |i\rangle \\
&\xrightarrow{(3)} \frac{1}{p^k} \sum_{j \in \mathbb{F}_p^k} \sum_{j \in \mathbb{F}_p^k} \omega_p^{\langle i,x+j \rangle} |j\rangle \\
&= |x\rangle
\end{aligned}$$

Step (1): Create a uniform superposition of n states using k H_p-gates; Step (2): Apply for every index in parallel a conditional phase flip Z_p-gate controlled by the corrupted input and the specific index; and, Step (3): Apply an inverse generalized Hadamard gate H_p^{-1}. The last equality follows directly from the orthogonality of the ω_p, see also Lemma 1.7.2.

In case of errors, some of the inputs $c(i)$ are corrupted. The probability to find output z is then given by:

$$\Pr[z] = \frac{1}{p^{2k}} \left| \sum_{y \in \mathbb{F}_p^k} \omega_p^{c(y)+\langle y,z \rangle} \right|^2. \tag{4.7}$$

In the binary case, every corrupted term cancels precisely one uncorrupted term. For higher characteristics, this cancellation depends on the exact value of the terms, and hence on the corruption. Considering the worst-case corruption, we obtain a lower bound on the probability of correct decoding:

$$\begin{aligned}
\Pr[z] &= \frac{1}{p^{2k}} \left| \sum_{y \in \mathbb{F}_p^k} \omega_p^{c(y)+\langle y,z \rangle} \right|^2 \\
&\geq \left(1 - 2\frac{d(c, H(z))}{p^k} \right)^2,
\end{aligned} \tag{4.8}$$

where d denotes the Hamming distance. By comparing this expression with Equation (4.3) we see that the success probability increases with higher field characteristics. By the same reasoning as before, the circuit for higher characteristics remains of constant depth.

4.8 Reflections and outlook

This chapter presented a $\mathsf{QNC}^0[\oplus]$-circuit that can list decode the Hadamard code and that successfully returns the message with probability $\Omega(\varepsilon)$ for error rate $\frac{1}{2} - \varepsilon$.

A similar circuit can also be constructed using the Goldreich-Levin algorithm and the implementation of the Majority gate using a $\mathsf{QNC}^0[\oplus]$-circuit [HŠ05; TT13]. However, the circuit based on the Goldreich-Levin algorithm depends on the error parameter ε, and as a result, the size of the Majority gates will depend on ε. In contrast, the quantum circuits presented in Theorem 4.1.1 and Theorem 4.6.2 have size independent of ε.

Adcock and Cleve considered a quantum version of the Goldreich-Levin algorithm [AC02]. However, their focus was on the cryptographic applications instead of decoding error-correcting codes, leading to more complex proofs. The circuit resulting from their approach also depends on the error parameter ε.

Kawachi and Yamakami considered quantum list-decoding algorithms for conventional error-correcting codes [KY10]. They introduce shuffled codeword states, similar to Equation (4.5), and a way to retrieve a list of possible messages from them. However, preparing shuffled codeword states is nontrivial for most codes, especially with restricted computational resources. This chapter provides an explicit construction of the shuffled codeword state for the Hadamard code. In a follow-up work, Yamakami extend this line of research by considering faulty quantum circuits that implement the encoding [Yam16].

None of these approaches focused on the circuit depth. Thus, our result of separation (Theorem 4.5.3) is the first separation with respect to decoding error-correcting codes. Future work can extend our result in multiple directions:

1. Provide a $\mathsf{QNC}^0[\oplus]$-circuit for practical error-correcting codes other than the Hadamard code considered in this chapter. In Chapter 5 we take a first step in this direction by considering Reed-Muller codewords of degree at most 2;

2. Prove a general statement on the existence of $\mathsf{QNC}^0[\oplus]$-circuits for any error-correcting code. Such a result would be an important next step, as the result for $\mathsf{NC}^0[\oplus]$-circuits holds for any error-correcting code, whereas the quantum result focused on the Hadamard code;

3. Extend the result to noisy quantum circuits. Current quantum devices are noisy, limiting implementations of quantum algorithms. We expect that specific noise models can correspond to corruptions in the codeword. If error rates are sufficiently low, we can use the $\mathsf{QNC}^0[\oplus]$-circuits presented in this chapter to retrieve the correct message with high probability.

Chapter 5

Decoding quadratic codes

In this chapter we extend the results from the previous two chapters to retrieve a Reed-Muller codeword of degree at most 2, given as input a polynomial phase function with large Gowers U^3-norm. We revisit earlier work of Tulsiani and Wolf and show an improved query complexity by employing a quantum Fourier sampling routine, omitting a Fourier estimation subroutine, and by using a different algorithmic version of the Balog-Szemerédi-Gowers theorem with a simpler proof and better quantitative bounds [BS94; Gow98; Sch14]. This algorithmic version might be of independent interest.

5.1 Chapter overview

The previous two chapters considered list decoding corrupted error-correcting codes and a quantum approach for list decoding the corrupted Hadamard code. In this chapter, we extend this line of research by considering Reed-Muller codewords of degree at most 2. For corrupted linear Reed-Muller codewords, i.e., the Hadamard code, Fourier analysis provides the tools to retrieve the codeword. Fourier analysis is closely related to the Gowers U^2-norm, see also Equation (1.15). Higher-order Fourier analysis provides the tools to analyse the results for corrupted Reed-Muller codewords of degree at most 2. These corrupted codewords have large Gowers U^3-norm.

The algorithm described in this chapter can find a Reed-Muller codeword of degree at most 2, such that it correlates with the corrupted input with large Gowers U^3-norm. We let $\delta(f, g)$ denote the normalized Hamming distance between two functions f and g, where both functions are evaluated on all possible inputs. We

let $\delta_{\mathrm{RM}_2}(f)$ denote the minimum Hamming distance between the function f and any Reed-Muller codeword of degree at most 2. The next theorem will be the main theorem of this chapter.

5.1.1. Theorem. *For any $\eta > 0$, there is an $\varepsilon > 0$ such that the following holds: There exists a quantum algorithm that, given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ satisfying $\delta_{\mathrm{RM}_2}(f) \leq \frac{1}{2} - \varepsilon$, makes at most $\mathcal{O}_\varepsilon(n \log n)$ queries to f and returns a polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree at most 2 such that $\delta(f, g) \leq \frac{1}{2} - \eta$.*

The dependency of η on ε in the theorem is exponential: $\eta = \exp(-1/\varepsilon^C)$ for some constant $C > 0$. The bulk of the work in proving Theorem 5.1.1 goes into finding a quadratic phase function that correlates with f sufficiently well, as summarized in the next lemma. Note that this lemma holds for any fixed prime p.

5.1.2. Lemma. *There is a quantum algorithm that, given a polynomial phase function $f : \mathbb{F}_p^n \to \mathbb{D}$ satisfying $\|f\|_{U^3} \geq \gamma$, makes $\mathrm{poly}(1/\gamma) n \log n$ queries to f and with probability $\mathrm{poly}(\gamma)$, returns a matrix $M \in \mathbb{F}_p^{n \times n}$ such that*

$$\mathbb{E}_{h \in \mathbb{F}_p^n} \left| \widehat{\Delta_h f}(Mh) \right|^2 \geq \exp(-\mathrm{poly}(1/\gamma)).$$

Once we have such a matrix M, we can use the next two lemmas, together with the Bernstein-Vazirani algorithm, to obtain (with good probability) a quadratic phase that correlates with f. We distinguish between odd primes and $p = 2$.

5.1.3. Lemma (Green–Tao [GT08]). *Let p be an odd prime, $M \in \mathbb{F}_p^{n \times n}$ and $f : \mathbb{F}_p^n \to \mathbb{D}$ be such that*

$$\mathbb{E}_{h \in \mathbb{F}_p^n} \left| \widehat{\Delta_h f}(Mh) \right|^2 \geq \delta.$$

Then, there is a $b \in \mathbb{F}_p^n$ such that the symmetric matrix $M' = (M + M^{\mathsf{T}})/2$ satisfies

$$\left| \mathbb{E}_{x \in \mathbb{F}_p^n} f(x) \omega_p^{\langle x, M'x \rangle / 2 + \langle x, b \rangle} \right| \geq \Omega_{p,\delta}(1).$$

For the case $p = 2$, we have a similar result, due to Samorodnitsky [Sam07] (see also [HHL19, Section 4.3]).

5.1.4. Lemma (Samorodnitsky). *Let $M \in \mathbb{F}_2^{n \times n}$ and $f : \mathbb{F}_2^n \to \{-1, 1\}$ be such that*

$$\mathbb{E}_{h \in \mathbb{F}_2^n} \left| \widehat{\Delta_h f}(Mh) \right|^2 \geq \delta.$$

Let $W = \ker(M + M^{\mathsf{T}})$ and $W^\perp \subseteq \mathbb{F}_2^n$ be its orthogonal complement. Let $M' \in \mathbb{F}_2^n$ be the matrix satisfying $M'x = Mx$ for all $x \in W$ and $\ker(M') = W^\perp$. Let v be the diagonal of the matrix M' and let $M'' = M' + vv^{\mathsf{T}}$. Then, there is a $b \in \mathbb{F}_2^n$ such that

$$\left| \mathbb{E}_{x \in \mathbb{F}_2^n} f(x) \omega_p^{\langle x, M''x \rangle + \langle x, b \rangle} \right| \geq \Omega_\delta(1).$$

Lemma 5.1.3 and Lemma 5.1.4 show how, given a matrix M as in Lemma 5.1.2, to obtain a matrix M' such that the function $g(x) = f(x)\omega_p^{-\langle x,M'x\rangle}$ satisfies $|\widehat{g}(b)| \geq \Omega_{p,\gamma}(1)$ for some $b \in \mathbb{F}_p^n$. Since we can query g by sequentially querying f and then the quadratic phase $\omega_p^{-\langle x,M'x\rangle}$, we can use the Bernstein-Vazirani algorithm to find such a b with positive probability. Finally, we can find a c such that

$$\Re\left(\mathbb{E}_x f(x)\omega_p^{-\langle x,Mx+b\rangle-c}\right) \geq \Omega_\gamma(1).$$

Specifically, as p is a fixed constant, we can evaluate this expression for all p possible values of c and choose the one among them that maximizes the expression.

The next two subsections give an outline of the proof of Lemma 5.1.2. Again, we start of gradually with a warm-up. Section 5.2 shows how the Fourier sampling algorithm introduced in Lemma 1.7.5 can be used to decoded corrupted quadratic Reed-Muller codewords within the unique decoding radius. With this warm-up, we can get accustomed to the ideas underlying the later proofs. The remainder of this chapter then provides the algorithmic proofs of the results in the next two subsections. Specifically, Section 5.1.1 provides an outline for Section 5.3; Section 5.1.2 provides an outline for both Section 5.4 and Section 5.5; and, Section 5.6 will discuss how to obtain a matrix M that satisfies the constraints outlined in Lemma 5.1.3 and Lemma 5.1.4, thereby completing the proof of Lemma 5.1.2 and hence of Theorem 5.1.1. Finally, Section 5.7 proves a lower bound on the query complexity, that matches the query complexity of our algorithm up to a logarithmic factor.

5.1.1 From non-uniformity to weak linearity

The starting point for proving Lemma 5.1.2 is the following basic result.

5.1.5. Proposition. *Suppose that $f : \mathbb{F}_p^n \to \mathbb{D}$ satisfies $\|f\|_{U^3} \geq \gamma$. Then, there is a set $S \subseteq \mathbb{F}_p^n$ of size at least $\gamma^8 p^n/2$ and a map $\phi : S \to \mathbb{F}_p^n$ such that for all $h \in S$, we have*

$$\left|\widehat{\Delta_h f}\big(\phi(h)\big)\right|^2 \geq \frac{\gamma^8}{2}.$$

Proof: It follows by the nesting property of the Gowers norms (Equation (1.17)) and Parseval's identity (Theorem 1.7.3) that

$$\begin{aligned}\gamma^8 &\leq \|f\|_{U^3}^8 \\ &= \mathbb{E}_{h\in\mathbb{F}_p^n}\|\Delta_h f\|_{U^2}^4 \\ &\leq \mathbb{E}_{h\in\mathbb{F}_p^n}\max_{a\in\mathbb{F}_p^n}|\widehat{\Delta_h f}(a)|^2.\end{aligned}$$

Letting $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^n$ be any map such that $|\widehat{\Delta_h f}(\phi(h))|$ is maximal for each $h \in \mathbb{F}_p^n$ and applying Markov's inequality (Lemma 2.6.1) then gives the result. □

The next step establishes that a map ϕ as in Proposition 5.1.5 satisfies a weak form a linearity. For a finite Abelian group G, an *additive quadruple* is a four-tuple $(a, b, c, d) \in G^4$ satisfying $a + b = c + d$. The *energy* of a set $A \subseteq G$ is then defined as the number of additive quadruples contained in A. A set $A \subseteq G$ can be shown to have energy $|A|^3$ if and only if A is the coset of a subgroup. For two sets $A, B \subseteq G$, we define $A + B$ as the set $\{a + b \mid a \in A, b \in B\}$. Then, if $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^n$ is some map and its graph

$$A = \big\{(h, \phi(h)\big) \mid h \in \mathbb{F}_p^n\big\} \subseteq \mathbb{F}_p^n \times \mathbb{F}_p^n$$

has energy p^{3n}, it turns out that ϕ must be an affine linear map (and vice versa).

5.1.6. Lemma (Weak linearity). *For a map $f : \mathbb{F}_p^n \to \mathbb{D}$, a set $S \subseteq \mathbb{F}_p^n$ and a map $\phi : S \to \mathbb{F}_p^n$, suppose that*

$$\sum_{h \in S} \big|\widehat{\Delta_h f}\big(\phi(h)\big)\big|^2 \geq \varepsilon p^n.$$

Then, the set $\{(h, \phi(h)) \mid h \in S\}$ has energy at least $\varepsilon^4 p^{3n}$.

5.1.2 From weak linearity to true linearity

Given the set A from Lemma 5.1.6, we now use results from additive combinatorics to lift the weak form of linearity of ϕ to true linearity on a large affine subspace. Let $A \subseteq G$, we call $A + A$ the doubling set of A, and the relative size $|A + A|/|A|$ the doubling constant. In practice, determining the doubling constant exactly is hard. Yet, lower bounds can often be given. The next two results show that the set A from Lemma 5.1.6 has small doubling and that this implies that it is contained in a slightly larger affine subspace. On this affine subspace, ϕ behaves linearly.

5.1.7. Theorem (Balog–Szemerédi–Gowers). *There is an absolute positive constant C such that the following holds. Let G be a finite Abelian group and suppose that $A \subseteq G$ has energy at least $\delta|A|^3$. Then, there is a set $A' \subseteq A$ of size $|A'| \geq \delta^C |A|$ such that $|A' + A'| \leq \delta^{-C}|A'|$.*

5.1.8. Theorem (Freĭman–Ruzsa). *For any $K \geq 1$ and prime number p, there is a $K' > 0$ such that the following holds. Suppose $A \subseteq \mathbb{F}_p^n$ satisfies $|A + A| \leq K|A|$. Then, the linear span of A satisfies $|\langle A \rangle| \leq K'|A|$.*

5.1.9. Remark. Recently, Gowers et al. proved the longstanding Marton conjecture, also known as the polynomial Freiman-Ruzsa conjecture [Gow+25], improving the dependency of K' on K in the previous theorem. The setting considered by Gowers et al. differs however from the situation considered in our work. Additionally, their result is non-algorithmic. A future algorithmic version of their result might improve the bounds found in this chapter.

It then follows that there exists a set

$$A' \subseteq \left\{ \big(h, \phi(h)\big) \mid h \in \mathbb{F}_p^n \right\}$$

of large size $|A'| \geq cp^n$ such that $|\langle A' \rangle| \leq C|A'|$, for some $c, C > 0$ depending on ε and p only. Moreover, each pair $(h, \phi(h)) \in A'$ satisfies

$$\big|\widehat{\Delta_h f}\big(\phi(h)\big)\big| \geq c.$$

Let $(h_1, \phi(h_1)), \ldots, (h_m, \phi(h_m))$ be a maximal set of linearly independent elements in A'. Let $M \in \mathbb{F}_p^{n \times n}$ be a matrix satisfying the set of linearly independent equations

$$Mh_i = \phi(h_i)$$

for all $i \in [m]$. The map ϕ then behaves linearly on all of A', in the sense that $\phi(h) = Mh$ for all $(h, \phi(h)) \in A'$. Hence, due to the relative size of A' in its linear span, we get that

$$\begin{aligned} \mathbb{E}_{h \in \mathbb{F}_p^n} \big| \mathbb{E}_{x \in \mathbb{F}_p^n} \Delta_h f(x) \omega_p^{\langle x, Mh \rangle} \big|^2 &= \mathbb{E}_{h \in \mathbb{F}_p^n} \big| \widehat{\Delta_h f}(Mh) \big|^2 \\ &\geq \mathbb{E}_{h \in \mathbb{F}_p^n} \mathbf{1}\big[\big(h, \phi(h)\big) \in A' \big] \big| \widehat{\Delta_h f}(Mh) \big|^2 \\ &\geq c. \end{aligned} \tag{5.1}$$

5.2 Quantum decoding of quadratic codes in a noiseless case

Below we discuss how to learn quadratic functions in a noiseless setting and how we can modify this algorithm to learn the function with high probability in a low-error setting, in both cases using the Fourier sampling subroutine (Lemma 1.7.5).

5.2.1 Noiseless case

Montanaro considered a problem of learning an unknown multilinear polynomial of degree at most d and presented a $\mathcal{O}(n^{d-1})$-query quantum algorithm that solves it [Mon12].

5.2.1. Theorem. *Given a quadratic phase function $f : \mathbb{F}_p^n \to \mathbb{D}$, we can learn f using $pn + 2$ quantum queries to f.*

The algorithm uses Fourier sampling to learn the quadratic phase function. In the noiseless case, we have the following lemma on the Fourier coefficients of multiplicative derivatives found using Fourier sampling.

5.2.2. LEMMA. *Let $f : \mathbb{F}_p^n \to \mathbb{D}$ be a quadratic phase given by $f(x) = \omega_p^{\langle x,Mx\rangle+\langle x,b\rangle+c}$, for some matrix $M \in \mathbb{F}_p^{n\times n}$, vector $b \in \mathbb{F}_p^n$ and scalar $c \in \mathbb{F}_p$. Then,*

$$\widehat{\Delta_h f}(y) = \begin{cases} 1 & \text{if } y = (M + M^T)h \\ 0 & \text{else.} \end{cases}$$

Proof: We expand the derivative and see which terms cancel:

$$\begin{aligned} \widehat{\Delta_h f}(y) &= \frac{1}{p^n} \sum_{x\in\mathbb{F}_p^n} \overline{f(x)} f(x+h) \omega_p^{\langle x,y\rangle} \\ &= \frac{1}{p^n} \sum_{x\in\mathbb{F}_p^n} \omega_p^{\langle x,(M+M^T)h\rangle+\langle b,h\rangle} \omega_p^{\langle x,y\rangle} \\ &= \frac{1}{p^n} \omega_p^{\langle b,h\rangle} \sum_{x\in\mathbb{F}_p^n} \omega_p^{\langle x,(M+M^T)h+y\rangle} \\ &= \begin{cases} 1 & \text{if } y = (M + M^T)h \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

□

Proof of Theorem 5.2.1: Lemmas 1.7.5 and 5.2.2 show that for fixed $h \in \mathbb{F}_2^n$ and using p queries to f, we obtain the output $(M + M^T)h$ with certainty.

Choosing n linearly independent h's (for instance the standard basis vectors) gives n linearly independent pairs of the form $(h, (M + M^T)h)$. We now learn M using Gaussian elimination combined with the fact that we can set M to be upper diagonal.

Next, we learn b by running the Bernstein-Vazirani algorithm, where f is queried and multiplied by the phase $\omega_p^{\langle x,Mx\rangle}$. We learn c by querying $f(0)$. □

5.2.2 Decoding within the unique-decoding radius

In the noisy case, the Fourier coefficients no longer behave as delta functions, but instead as "approximate delta functions". By this, we mean that in the Fourier spectrum, almost all terms have small magnitude and only a bounded number have larger amplitude. The number of terms with larger magnitude depends on the actual error rate.

Below we illustrate what happens in case of few errors and how our algorithm changes. We restrict ourselves to $p = 2$ for simplicity, yet the results extend to odd primes p. We know that within the unique-decoding radius, a unique codeword exists that is closest to the corrupted codeword. In this case, essentially the same algorithm works as in the noiseless case, with the main difference that the Fourier

transforms of the multiplicative derivatives are no longer delta functions. Instead, they satisfy that there exists a unique Fourier coefficient whose absolute valued squared is strictly greater than 1/2. The existence of this Fourier coefficient implies that $\mathcal{O}(\log n)$ repetitions of the Fourier sampling algorithm can locate the large Fourier coefficient with high probability via Hoeffding's inequality.

5.2.3. THEOREM. *Let $\varepsilon > 0$. Let f' have relative distance at most $\frac{1}{4} - \frac{1}{4}\sqrt{\frac{1}{2} + \varepsilon}$ from a degree-2 Reed-Muller codeword f. Then, with success probability at least $1 - \frac{1}{n}$, we can learn f using $\mathcal{O}_\varepsilon(n \log n)$ quantum queries.*

Note that the allowed relative distance in the theorem is slightly smaller than the unique decoding radius of $\frac{1}{8}$ for Reed-Muller codewords of degree at most 2.

Proof: With probability $\frac{3}{4} + \frac{1}{4}\sqrt{\frac{1}{2} + \varepsilon}$, $f'(x) = f(x)$ for a random x. By the union bound, it holds for uniformly random x and h that $f'(x+h)\overline{f'(x)} = f(x+h)\overline{f(x)}$ with probability at least $\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \varepsilon}$.

Let $c(x)$ be the indicator variable that evaluates to 1 if $f'(x) \neq f(x)$. Then, revisiting the proof of Lemma 5.2.2, we find that the probability for the correct measurement outcome y is given by:

$$\begin{aligned}\Pr[y] &= |\widehat{\Delta_h f}(y)|^2 \\ &= \frac{1}{2^{2n}} \Big| \sum_{x \in \mathbb{F}_2^n} (-1)^{c(x)+c(x+h)+\langle x,(M+M^T)h+y\rangle} \Big|^2\end{aligned}$$

For $y = (M + M^T)h$, this thus evaluates to

$$\begin{aligned}\Pr[y = (M+M^T)h] &= \frac{1}{2^{2n}} \Big| \sum_{x \in \mathbb{F}_2^n} (-1)^{c(x)+c(x+h)} \Big|^2 \\ &\geq \frac{1}{2} + \varepsilon. \end{aligned} \tag{5.2}$$

We can now amplify the probability of correctly finding y by taking the majority of multiple independent runs of the algorithm. By Hoeffding's inequality, the majority over the measurement outcomes of $m = \frac{1}{\varepsilon^2} \log n$ independent runs for the same h gives a valid pair $(h, (M + M^T)h)$ with probability $1 - \frac{1}{n^2}$.

Now run the above procedure for n linearly independent h-values (for instance the n standard basis vectors) to obtain n pairs $(h_i, (M + M^T)h_i)$. By the union bound, all pairs are correct with probability $1 - \frac{1}{n}$.

Similar to Theorem 5.2.1, we can learn M from these n linearly independent pairs. With two additional queries (one in superposition and one to a single index) we learn L and c with good probability, and thereby learn f. □

Conventional queries only return a single $f(h)$ value. Polynomial interpolation is therefore needed to learn a pair $(h, (M + M^T)h)$. Additionally, every query can be corrupted, requiring more queries to obtain a valid pair with high probability.

As the error rate increases beyond the unique-decoding radius, the Fourier coefficients of the multiplicative derivatives are no longer peaked at a single point. In fact, multiple Fourier coefficients can have the same absolute value squared. As a result, the outcome for various directions h and h' may result in pairs originating from different quadratic phases, that is, we might obtain two pairs $(h, (M+M^T)h)$ and $(h', (M' + M'^T)h')$ for distinct matrices $M \neq M'$.

As a result, combining the obtained pairs might result in a combined matrix with a distance to the correct quadratic Reed-Muller codeword larger than expected.

5.3 Quantum-algorithmic weak linearity

In this section, we show how to sample from a set with high energy provided we have a polynomial phase function $f : \mathbb{F}_p^n \to \mathbb{D}$ satisfying $\|f\|_{U^3} \geq \gamma$.

Given some $\delta \in (0, 1)$, define the *spectral set* S_δ to be the set of all (h, a) pairs with high Fourier coefficient:

$$S_\delta = \left\{(h, a) \in \mathbb{F}_p^n \times \mathbb{F}_p^n \mid |\widehat{\Delta_h f}(a)| \geq \delta\right\}. \tag{5.3}$$

Let $\delta_1 = \text{poly}(\gamma)$ be some fixed parameter.

Let $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^n$ be the random map where each coordinate is independently sampled according to the Fourier sampling algorithm from Lemma 1.7.5, namely, for each $h \in \mathbb{F}_p^n$,

$$\Pr\big[\phi(h) = a\big] = |\widehat{\Delta_h f}(a)|^2. \tag{5.4}$$

Based on the map ϕ, define the random set

$$A_0 = \big\{\big(h, \phi(h)\big) \mid h \in \mathbb{F}_p^n\big\}. \tag{5.5}$$

Note that we can sample a uniformly random element from A_0 by first sampling a uniformly random $h \in \mathbb{F}_p^n$ and then sampling $\phi(h)$ using Fourier sampling. This way, we sample sequentially from the joint probability distribution imposed on A_0. Later, we will also have to check if a given pair (h, a) belongs to A_0. We do this by applying the Fourier sampling routine on h and comparing the outcome with a.

We will keep track of the spectral elements in this set

$$A_1 = A_0 \cap S_{\delta_1}. \tag{5.6}$$

5.3.1. Proposition. *With probability* $\mathrm{poly}(\gamma)$*, we have that* $|A_1| \geq \mathrm{poly}(\gamma)p^n$.

Proof: Proposition 5.1.5 and Markov's inequality (Lemma 2.6.1) show that there are at least $\frac{\gamma^8}{2}p^n$ many $h \in \mathbb{F}_p^n$ such that $\max_a |\widehat{\Delta_h f}(a)|^2 \geq \frac{\gamma^8}{2}$. Define the indicator random variables

$$X_h = \mathbf{1}\left[\phi(h) = \arg\max_a |\widehat{\Delta_h f}(a)|\right]$$

and let $\overline{X} = \sum_{h\in\mathbb{F}_p^n} X_h$. Note that if X_h holds, then $\big(h, \phi(h)\big) \in S_{\delta_1}$. Then,

$$\mathbb{E}\big[\overline{X}\big] \geq \frac{\gamma^{16}}{4}p^n.$$

By Markov's inequality, we have that $\overline{X} \geq \frac{\gamma^{16}}{8}p^n$ with probability at least $\frac{\gamma^{16}}{8}$. □

It follows from Lemma 5.1.6 that if the event from Theorem 5.3.1 holds, then the (random) set A_1 has energy at least $\mathrm{poly}(\gamma)|A_1|^3$. As the size of A_1 is large, a sample from A_0 is in A_1 with constant probability. As a result, contrary to the approach of Tulsiani and Wolf, we do not have to sample from A_1 directly, thereby reducing the query complexity. The next step is to construct an algorithmic version of the Balog-Szemerédi-Gowers theorem.

5.4 A variant of the Balog-Szemerédi-Gowers theorem

In this section, we prove a variant of the Balog–Szemerédi–Gowers theorem that is inspired by a proof of this theorem due to Schoen [Sch14]. The variant outlined here is tailored for the purpose of turning it into an algorithm that we give in the following section.

Let G be a finite Abelian group and let $A \subseteq G$, let $E(A)$ denote the energy of A. For an element $x \in G$, define the *popularity* of x relative to A by

$$r_A(x) = |\{(a, b) \in A^2 : a - b = x\}|.$$

The next lemma defines two sets $H_1 \subseteq H_0$ such that H_0 has small doubling and H_1 is large. We can interpret the lemma as an algorithmic way of preparing a bipartite graph. Two nodes in the bipartite graph are connected if their difference is popular. It turns out that a large fraction of the vertices in the graph has high degree, which we can use to construct the set of small doubling.

5.4.1. Lemma. *Let* $\delta, \eta > 0$ *and suppose that* $A_1 \subseteq A_0 \subseteq G$ *are sets such that* $|A_1| = \eta|A_0|$ *and* A_1 *has energy at least* $\delta|A_1|^3$*. Let* X, Y *be independent uniformly distributed random variables over* A_1 *and define the random sets*

$$B_i = A_i \cap (A_i - X + Y). \tag{5.7}$$

Define subsets $H_1 \subseteq H_0$ by

$$H_0 = \Big\{ a \in B_0 \mid \sum_{b \in B_0} \mathbf{1}\big[r_{A_0}(a-b) \geq \tfrac{\delta^2}{32}|A_0|\big] \geq \tfrac{5}{8}|B_0| \Big\} \tag{5.8}$$

$$H_1 = \Big\{ a \in B_1 \mid \sum_{b \in B_0} \mathbf{1}\big[r_{A_0}(a-b) \geq \tfrac{\delta^2}{16}|A_0|\big] \geq \tfrac{3}{4}|B_0| \Big\}. \tag{5.9}$$

Then, with probability at least $\frac{\delta^2\eta^2}{4}$ over the choice of X and Y, we have that

$$|H_1| \geq \tfrac{\delta}{C}|A_1| \quad \textit{and} \quad |H_0 + H_0| \leq \frac{C}{\delta^{11}\eta^{11}}|A_0|,$$

where $C > 0$ is an absolute constant.

By this lemma, the set H_0 has small doubling and the set H_1 is large. As also $A_1 \subseteq A_0$, we see that there exists a large set of small doubling interspersed between A_1 and A_0. The next lemma states that most vertices in the bipartite graph induced by the sets H_0 and H_1 have high degree.

5.4.2. Lemma. *Let $A_1 \subseteq A_0$ be finite additive sets such that $|A_1| = \eta|A_0|$ and $E(A_1) = \delta|A_1|^3$. Let X, Y be independent uniformly distributed A_1-valued random variables. For $i \in \{0,1\}$, let*

$$B_i = A_i \cap (A_i - X + Y).$$

Then, for any $\varepsilon > 0$, with probability at least $\frac{1}{4}\delta^2\eta^2$, we have that $|B_1| \geq \frac{1}{4}\delta|A_1|$ and $B_1 \times B_0$ has at least $\big(1 - \frac{2\varepsilon}{\delta^2\eta^2}\big)|B_1||B_0|$ pairs (a,b) such that $r_{A_0}(a-b) \geq \varepsilon|A_0|$.

Proof: Let $\mathbf{1}_A[a]$ denote the indicator value of the event $a \in A$. Since $E(A_1)$ equals the number of triples a, x, y such that $a + x - y \in A_1$, we have

$$\delta|A_1| = \frac{E(A_1)}{|A_1|^2} = \mathbb{E}_{x,y \in A_1} \sum_{a \in A_1} \mathbf{1}_{A_1}[a + x - y] = \mathbb{E}|B_1|.$$

Define the set of unpopular pairs

$$S = \{(a,b) \in A_1 \times A_0 \mid r_{A_0}(a-b) < \varepsilon|A_0|\}.$$

Let $T = B_1 \times B_0 \cap S$. Note that $\Pr[X - Y = z] = r_{A_1}(z)/|A_1|$. Then,

$$\begin{aligned}
\mathbb{E}|T| &= \mathbb{E}_{x,y \in A_1}\Big[\sum_{(a,b) \in S} \mathbf{1}_{A_1 - x + y}[a] \mathbf{1}_{A_0 - x + y}[b] \Big] \\
&= \frac{1}{|A_1|} \sum_z r_{A_1}(z) \sum_{(a,b) \in S} \mathbf{1}_{A_1 - z}[a] \mathbf{1}_{A_0 - z}[b] \\
&\leq \frac{1}{|A_1|} \sum_{(a,b) \in S} \sum_z \mathbf{1}_{A_1}[a + z] \mathbf{1}_{A_0}[b + z] \\
&\leq \frac{1}{|A_1|} \sum_{(a,b) \in S} \sum_z \mathbf{1}_{A_0}[z] \mathbf{1}_{A_0}[z - a + b].
\end{aligned}$$

We see that this last expression is equal to

$$\frac{1}{|A_1|}\sum_{(a,b)\in S} r_{A_0}(a-b),$$

which we can upper bound by $\varepsilon|A_0|^2$. Using the inequalities for $\mathbb{E}|T|$ and $\mathbb{E}|B_1|$, it follows from the Cauchy-Schwarz inequality that for $\mu = \frac{\delta^2\eta^2}{2\varepsilon}$, we have

$$\mathbb{E}\big(|B_1|^2 - \mu|T|\big) \geq \big(\mathbb{E}|B_1|\big)^2 - \mu\mathbb{E}|T| \geq \frac{\delta^2\eta^2}{2}|A_0|^2.$$

Hence, by Markov's inequality, with probability at least $\frac{\delta^2\eta^2}{4}$, we have

$$|B_1|^2 - \mu|T| \geq \frac{\delta^2\eta^2}{4}|A_0|^2.$$

This gives the desired lower bound on $|B_1|$. Since $B_1 \subseteq B_0$, it also implies that

$$|B_1||B_0| \geq \mu|T|.$$

Rearranging, this last expression shows that there are at most $\frac{2\varepsilon}{\delta^2\eta^2}|B_1||B_0|$ pairs $(a,b) \in B_1 \times B_0$ such that $r_{A_0}(a-b) < \varepsilon|A_0|$. □

In proving Lemma 5.4.1, we will use a triangle-inequality-like result [Zha23, Corollary 7.3.6].

5.4.3. Lemma. *For any three sets $A, B, C \subseteq G$, we have that*

$$|A||B+C| \leq |A+B||A+C|.$$

Proof of Lemma 5.4.1: Let B_1, B_0 be as in Lemma 5.4.1 and set $\varepsilon = \frac{\delta^2\eta^2}{16}$. Define the bipartite graphs

$$\Gamma_1 = \big\{(a,b) \in B_0 \times B_0 \mid r_{A_0}(a-b) \geq \tfrac{\delta^2\eta^2}{32}|A_0|\big\}$$
$$\Gamma_2 = \big\{(a,b) \in B_1 \times B_0 \mid r_{A_0}(a-b) \geq \tfrac{\delta^2\eta^2}{16}|A_0|\big\}.$$

Then, $|\Gamma_2| \geq \frac{7}{8}|B_1||B_0|$. Define the sets

$$H_0 = \big\{a \in B_0 \mid \deg_{\Gamma_1}(a) \geq \tfrac{5}{8}|B_0|\big\}$$
$$H_1 = \big\{a \in B_1 \mid \deg_{\Gamma_2}(a) \geq \tfrac{3}{4}|B_0|\big\}.$$

Due to the size of Γ_2, it follows that $|H_1| \geq \frac{1}{8}|B_1|$.

By the inclusion-exclusion principle, since any two distinct left vertices $u, v \in H_0$ have such a high degree in Γ_1, they must have at least $\frac{1}{4}|B_0|$ common (right)

neighbors. If $x \in B_0$ is a right neighbor of u, then $r_{A_0}(u-x) \geq \frac{\delta^2\eta^2}{32}|A_0|$ and similarly for v. It follows that

$$\sum_{x \in B_0} r_{A_0}(u-x) r_{A_0}(v-x) \geq \left(\tfrac{1}{4}|B_0|\right)\left(\tfrac{\delta^2\eta^2}{32}|A_0|\right)^2 \geq 2^{-14}\delta^5\eta^5|A_0|^3.$$

Observe that the left-hand side of the equation is a lower bound on the number of four-tuples $(a,b,c,d) \in A_0^4$ such that $(a-b)-(c-d) = u-v$. For each $w \in H_0 - H_0$, let $u_w, v_w \in H_0$ be an arbitrary pair such that $u_w - v_w = w$. Then,

$$|A_0|^4 \geq \sum_{w \in H_0 - H_0} \sum_{x \in B_0} r_A(u_w - x) r_{A_0}(v_w - x) \geq 2^{-14}\delta^5\eta^5 |H_0 - H_0||A_0|^3.$$

Rearranging then gives

$$|H_0 - H_0| \leq \frac{2^{14}}{\delta^5\eta^5}|A_0|. \tag{5.10}$$

Since we have $H_0 \subseteq H_1$, combining Equation (5.10) with Lemma 5.4.3 applied with $A = -B = -C = H_0$ then gives the desired bound on the doubling of H_0:

$$|H_0 + H_0| \leq \frac{|H_0 - H_0|^2}{|H_0|} \leq \frac{2^{31}}{\delta^{11}\eta^{11}}|A_0|.$$

□

5.5 Sampling high-degree elements

This section gives an algorithmic version of Lemma 5.4.1. Let X, Y be uniformly random elements from the (random) set A_0 as given in Equation (5.5). Based on A_0, and the spectral set A_1 as in Equation (5.6), define the sets B_0 and B_1 as in Equation (5.7) of Lemma 5.4.1. Define the event

$$\mathcal{E}_1 = \big\{|A_1| \geq \text{poly}(\gamma)p^n \quad \wedge \quad X, Y \in A_1\big\}. \tag{5.11}$$

By Theorem 5.3.1, this event holds with probability $\text{poly}(\gamma)$. Conditioned on this event, both X and Y are uniformly distributed over A_1.

Let $\delta_2 = E(A_1)/|A_1|^3$ and $\eta = |A_1|/p^n$. Define subsets $H_0 \supseteq H_1$ as in Equations (5.8) and (5.9) of Lemma 5.4.1 using $\delta = \delta_2$. Let C be the constant from Lemma 5.4.1 and define the event

$$\mathcal{E}_2 = \big\{|H_1| \geq \tfrac{\delta_2}{C}|A_1| \text{ and } |H_0 + H_0| \leq \tfrac{C}{\delta_2^{11}\eta^{11}}|A_0|\big\}. \tag{5.12}$$

Conditioned on $\mathcal{E}_1$, event $\mathcal{E}_2$ holds with probability $\text{poly}(\gamma)$.

5.5.1 Approximating sets of small doubling

By event $\mathcal{E}_1$, we learn that A_1 is large. By event $\mathcal{E}_2$ know that H_1 has size at least a constant fraction of the size of A_1, hence H_1 is also large. The same event also tells us that H_0, which contains H_1, has doubling bounded by a constant times the size of A_0. As $A_1 \subseteq A_0$, we see that there should exist a set, called B_2, approximately between H_1 and H_0 that is large and has small doubling. Below, we give an algorithm that allows us to sample from B_2.

5.5.1. Definition (m-Popularity Estimator). Given input $u \in \mathbb{F}_p^{2n}$, independently and uniformly sample $v_1, \ldots, v_m$ from A_0. Return

$$\mathrm{PopEst}_m(u) := \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}[u + v_i \in A_0].$$

Note that the indicator function requires membership queries to A_0.

5.5.2. Claim. *Let $\delta \in (0,1)$ and $L = \lceil 12/\delta^2 \rceil$. Conditioned on A_0, the random events*

$$\Big\{ \big| \mathrm{PopEst}_L(u) - \tfrac{r_{A_0}(u)}{|A_0|} \big| > \delta/2 \Big\}, \quad u \in \mathbb{F}_p^{2n}$$

are jointly independent, and each occurs with probability at most $1/100$.

Proof: The independence follows as the samples in the popularity estimator are taken independent of the input u. The second claim follows from Hoeffding's inequality (Lemma 2.6.2) and the fact that $\mathbb{E}_u \mathrm{PopEst}_L(u) = |A_0|^{-1} r_{A_0}(u)$. □

5.5.3. Definition ((m,δ)-Degree Estimator). Given input $w \in B_0$, independently and uniformly sample $u_1, \ldots, u_m$ from B_0. Return

$$\mathrm{DegEst}_{m,\delta}(w) := \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\big[\mathrm{PopEst}_L(w - u_i) \geq \delta \big],$$

where $L = \lceil 12/\delta^2 \rceil$.

Let $m_2 = \Omega(\log n)$ and define the random set

$$B_2 = \big\{ w \in B_0 \mid \mathrm{DegEst}_{m_2,\delta_2}(w) \geq 3/4 \big\}. \tag{5.13}$$

We show that with high probability – conditioned on the events $\mathcal{E}_1$ and $\mathcal{E}_2$ – a negligible fraction of B_2 lies outside of H_0, while it contains nearly all of H_1. To this end, define the random event

$$\mathcal{E}_3 = \big\{ |B_2 \cap H_0| \geq \big(1 - \tfrac{1}{n^2}\big)|B_2| \text{ and } |B_2 \cap H_1| \geq \big(1 - \tfrac{1}{n^2}\big)|H_1| \big\}, \tag{5.14}$$

which indicates that B_2 is very nearly sandwiched between H_0 and H_1. We show that $\mathcal{E}_3$ holds with high probability:

5.5.4. Lemma. *Conditioned on $\mathcal{E}_1$ and $\mathcal{E}_2$, the probability of event $\mathcal{E}_3$, taken over the random choices of the Popularity and Degree Estimators, is at least $1 - 1/n^2$.*

Proof: We prove that for all $w \notin H_0$ we have $\Pr[w \in B_2] \leq 1/n^2$. The case where if $w \in H_1$, then $\Pr[w \notin B_2] \leq 1/n^2$ follows via similar arguments.

Assume $w \in B_0 \setminus H_0$, then

$$\left| \left\{ v \in B_0 \mid \tfrac{r_{A_0}(w-v)}{|A_0|} < \frac{\delta_2^2}{32} \right\} \right| \geq \frac{3|B_0|}{8}.$$

Let $I \subseteq [m_2]$ denote the random set

$$\left\{ i \in [m_2] \mid \frac{r_{A_0}(w-u_i)}{|A_0|} < \frac{\delta_2^2}{32} \right\},$$

where $u_1, \ldots, u_{m_2} \in B_0$ are sampled independent and uniformly at random and are accepted by the (m_2, δ_2)-degree estimator. By Hoeffding's inequality, we have

$$\Pr\Big(|I| < \tfrac{7m_2}{24} \Big) \leq 2e^{-2m_2/9}. \tag{5.15}$$

Note that for any ξ and L, we have that

$$\sum_{i=1}^{m_2} \mathbf{1}[\mathrm{PopEst}_L(w-u_i) \geq \xi] \leq \sum_{i \in I} \mathbf{1}[\mathrm{PopEst}_L(w-u_i) \geq \xi] + m_2 - |I|.$$

It follows that

$$\begin{aligned}
&\Pr\left(\sum_{i=1}^{m_2} \mathbf{1}[\mathrm{PopEst}_L(w-u_i) \geq \xi] \geq \frac{3m_2}{4} \right) \\
&\leq \Pr\left(\sum_{i \in I} \mathbf{1}[\mathrm{PopEst}_L(w-u_i) \geq \xi] \geq |I| - \frac{m_2}{4} \right) \\
&\leq \Pr\Big(|I| < \frac{7m_2}{24} \Big) + \Pr\left(\sum_{i \in I} \mathbf{1}[\mathrm{PopEst}_L(w-u_i) \geq \xi] \geq \frac{m_2}{24} \,\Big|\, |I| \geq \frac{7m_2}{24} \right).
\end{aligned}$$

The first term is small as Equation (5.15) shows. For the second term, we use that for every $i \in I$, Theorem 5.5.2 combined with $\xi = \delta_2^2/16$ and $L = \lceil 12/\xi^2 \rceil$ gives

$$\begin{aligned}
\Pr\big(\mathrm{PopEst}_L(w-u_i) \geq \delta_2^2/16 \big) &\leq \Pr\left(\left| \mathrm{PopEst}_L(w-u_i) - \tfrac{r_{A_0}(w-u_i)}{|A_0|} \right| > \tfrac{\delta_2^2}{32} \right) \\
&\leq 1/100.
\end{aligned}$$

Hoeffding's inequality then gives

$$\Pr\left(\sum_{i \in I} \mathbf{1}\Big[\Big| \mathrm{PopEst}_L(w-u_i) - \tfrac{r_{A_0}(w-u_i)}{|A_0|} \Big| > \tfrac{\delta_2^2}{32} \Big] \geq \frac{|I|}{24} \right) \leq 2e^{-|I|/288}.$$

From this and Equation (5.15), we can conclude that

$$\begin{aligned}
&\Pr\left(\sum_{i\in I}\mathbf{1}\left[|\mathrm{PopEst}_L(w-u_i)|\geq\tfrac{\delta_2^2}{16}\right]\geq\frac{m_2}{24}\,\middle|\,|I|\geq\frac{7m_2}{24}\right)\\
&\leq\Pr\left(\sum_{i\in I}\mathbf{1}\left[\left|\mathrm{PopEst}_L(w-u_i)-\tfrac{r_{A_0}(w-u_i)}{|A_0|}\right|\geq\tfrac{\delta_2^2}{32}\right]\geq\frac{m_2}{24}\,\middle|\,|I|\geq\frac{7m_2}{24}\right)\\
&=\frac{\Pr\left(\sum_{i\in I}\mathbf{1}\left[\left|\mathrm{PopEst}_L(w-u_i)-\tfrac{r_{A_0}(w-u_i)}{|A_0|}\right|\geq\tfrac{\delta_2^2}{32}\right]\geq\tfrac{m_2}{24}\wedge|I|\geq\tfrac{7m_2}{24}\right)}{\Pr\left(|I|\geq\tfrac{7m_2}{24}\right)}\\
&\leq\frac{\Pr\left(\sum_{i\in I}\mathbf{1}\left[\left|\mathrm{PopEst}_L(w-u_i)-\tfrac{r_{A_0}(w-u_i)}{|A_0|}\right|\geq\tfrac{\delta_2^2}{32}\right]\geq\tfrac{|I|}{24}\wedge|I|\geq\tfrac{7m_2}{24}\right)}{\Pr\left(|I|\geq\tfrac{7m_2}{24}\right)}\\
&\leq\frac{\Pr\left(\sum_{i\in I}\mathbf{1}\left[\left|\mathrm{PopEst}_L(w-u_i)-\tfrac{r_{A_0}(w-u_i)}{|A_0|}\right|\geq\tfrac{\delta_2^2}{32}\right]\geq\tfrac{|I|}{24}\right)}{\Pr\left(|I|\geq\tfrac{7m_2}{24}\right)}\\
&\leq 4e^{-|I|/288},
\end{aligned}$$

where in the last line we upper bounded Equation (5.15) by $1/2$.

Combined, this gives $\Pr(w\in B_2)\leq 2e^{-2m_2/9}+4e^{-m_2/288}\leq 1/n^2$ for $w\notin H_0$. □

5.5.2 Finding a small spanning set

Next, we work towards obtaining a basis for the set B_2 of small doubling. Let $\mu=|B_2\cap H_1|/|B_2|$ and let

$$B_3=\{v_1,\ldots,v_{20n/\mu}\}\subseteq B_2$$

be a set of independent uniformly random elements from B_2. Define the event

$$\mathcal{E}_4=\{B_3\subseteq H_0 \text{ and } |\langle B_3\rangle\cap H_1|\geq|H_1|/2\}.$$

5.5.5. Lemma. *We have that* $\Pr[\mathcal{E}_4\mid\mathcal{E}_1\wedge\mathcal{E}_2\wedge\mathcal{E}_3]\geq 1/3$.

Proof: Assume events $\mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3$ all hold true. Since $|B_2\cap H_0|\geq(1-\frac{1}{n^2})|B_2|$, a uniform sample from B_2 lies in H_0 with probability at least $1-\frac{1}{n^2}$ and hence by the union bound, $B_3\subseteq H_0$ holds with probability $1-\mathcal{O}(\frac{1}{n})$.

Let $H_1'=B_2\cap H_1$ and let $I=\{i\mid v_i\in H_1'\}$. For each $i\in I$, the random variable v_i is uniformly distributed over H_1'. Since $|H_1'|=\mu|B_2|$, by Hoeffding's inequality, $|I|\geq 10n$ with probability $1-\mathcal{O}(\frac{1}{n})$. Assume this is the case and that $B_3\subseteq H_0$; note that both events hold simultaneously with probability $1-\mathcal{O}(\frac{1}{n})$.

Write $V_\emptyset=\{0\}\subseteq\mathbb{F}_p^{2n}$ and $V_J=\mathrm{span}\{v_i\mid i\in J\}\subseteq\mathbb{F}_2^{2n}$ for $J\subseteq I$.

We show by contradiction that

$$\Pr_{\{v_i | i \in I\}}\big(|V_I \cap H_1'| \geq |H_1'|/2\big) \geq 1/2.$$

Suppose that this is false. Then for every $J \subseteq I$,

$$\Pr_{\{v_i | i \in J\}}\big(|V_J \cap H_1'| \leq |H_1'|/2\big) \geq 1/2.$$

For any strict subset $J \subset I$ and $j \in I \setminus J$, we can then conclude that

$$\begin{aligned}
&\Pr\big(v_i \notin V_J\big) \\
&\quad \geq \Pr\big(|V_J \cap H_1'| \leq |H_1'|/2\big) \Pr\big(v_j \notin V_J \mid |V_J \cap H_1'| \leq |H_1'|/2\big) \\
&\quad \geq \frac{1}{2}\Pr\big(v_j \in H_1' \setminus V_J \mid |H_1' \setminus V_J| \geq |H_1'|/2\big) \\
&\quad \geq \frac{1}{4}.
\end{aligned}$$

For simplicity, assume that $I = \{1, \ldots, |I|\}$. As a result of the above

$$\mathbb{E}\dim(V_I) = 1 + \sum_{i=1}^{|I|-1} \Pr\big(v_{i+1} \notin V_{\{v_1,\ldots,v_i\}}\big) \geq 10n/4.$$

This is a contradiction since $\dim(V_I) \leq \dim(\langle H_1' \rangle) \leq \dim(H_0) \leq 2n$. Taking the union bound over the three events then completes the proof. □

5.6 Constructing an approximating matrix for ϕ

Recall that from event $\mathcal{E}_4$, we have that $B_3 \subseteq H_0$ and $|\langle B_3 \rangle \cap H_1| \geq |H_1|/2$. Moreover, by event $\mathcal{E}_2$, we have $|H_0 + H_0| \leq \mathrm{poly}(1/\gamma)|H_0|$ and from event $\mathcal{E}_1$ we have that $|H_1| \geq \mathrm{poly}(\gamma)|H_1|$. It thus follows from Theorem 5.1.8 that

$$|\langle B_3 \rangle| \leq |\langle H_0 \rangle| \leq \exp(-\mathrm{poly}(1/\gamma))|H_0|. \tag{5.16}$$

Note that in this upper bound an algorithmic polynomial version of Theorem 5.1.8 gives improved dependencies on γ; see also Remark 5.1.9 for a discussion.

Denote $V = \langle B_3 \rangle$ and let $\pi : V \to \mathbb{F}_p^n$ be the projection to the first n coordinates. Let $T \subseteq V$ be the complement of the subspace $\ker(\pi)$ such that $V = T \oplus \ker(\pi)$. Combing two results of Green and Tao [GT08] (see also [Gre07, Proposition 2.6]) and Samorodnitsky [Sam07, Lemma 6.10][10] shows that

$$|H_1 \cap T| \geq \frac{|H_1|}{2|\ker(\pi)|} \geq \exp(-\mathrm{poly}(1/\gamma))|H_0|, \tag{5.17}$$

[10] The lemma is proved only for $\mathbb{F}_2^n$ but the same proof works for larger prime fields.

where we used that $|\pi(H_1)| = |H_1|$. Since T is a linear subspace of V on which π acts injectively, it follows that there exists a matrix $M \in \mathbb{F}_p^{n\times n}$ such that

$$T = \big\{(h, Mh) \mid h \in \pi(T)\big\}.$$

By standard linear algebra techniques, we can find a basis for $\pi(T)$ which can be used to get an explicit description of such a matrix M.

By event $\mathcal{E}_2$, we have that $|H_0| \geq |H_1| \geq \text{poly}(\gamma)|A_1|$. Since $H_1 \subseteq A_1 \subseteq A_0$, it follows from Equation (5.17) and event $\mathcal{E}_1$ that

$$|A_1 \cap T| \geq \exp(-\text{poly}(1/\gamma))|A_1|.$$

It follows that

$$\begin{aligned}\mathbb{E}_{h\in\mathbb{F}_p^n}\big|\widehat{\Delta_h f}(Mh)\big|^2 &\geq \mathbb{E}_{h\in\mathbb{F}_p^n}\mathbf{1}_{S_{\delta_1}\cap A_0\cap T}[(h,Mh)]\,\big|\widehat{\Delta_h f}(Mh)\big|^2\\ &\geq \exp(-\text{poly}(1/\gamma)).\end{aligned}$$

5.7 Lower bound on query complexity

Montanaro [Mon12] obtained tight bounds on the query complexity of learning multilinear polynomials over a finite field $\mathbb{F}_p$, both in the conventional setting and in the quantum setting. His arguments for proving the lower bounds were based on elementary information theory, and easily generalize to the following result:

5.7.1. Lemma. *Let $\mathcal{F} \subseteq \{f : \mathbb{F}_p^n \to \mathbb{F}_p\}$ be a family of functions, and let f be a uniformly chosen element of $\mathcal{F}$. Then:*

1. *Any conventional query algorithm that learns f with bounded error must make $\Omega\big(\log|\mathcal{F}|/(\log p)\big)$ queries to f.*
2. *Any quantum query algorithm that learns f with bounded error must make $\Omega\big(\log|\mathcal{F}|/(n\log p)\big)$ queries to f.*

Proof: We view f as a random variable uniformly distributed over $\mathcal{F}$, which thus has entropy $H(f) = \log|\mathcal{F}|$. We will use the intuitive fact that one cannot learn f from much fewer than $H(f)$ bits of information.

Fano's inequality formalizes this fact: for any random variable X we have[11]

$$H(f \mid X) \leq H\big(\mathbf{1}[f \neq X]\big) + \Pr[f \neq X](\log|\mathcal{F}| - 1).$$

See for example [NC10, Chapter 12] for a proof of this inequality. One can think of X as being our best guess for the function f based on some partial information.

[11]Here $H(f \mid X)$ is the entropy of f conditional on knowing X, defined as $H(f \mid X) = H(f) - I(f : X)$ where $I(f : X)$ is the mutual information.

Using that $H(f \mid X) = \log|\mathcal{F}| - I(f:X)$ and that $H\big(\mathbf{1}[f \neq X]\big) \leq 1$, we obtain

$$\Pr[f \neq X] \geq 1 - \frac{I(f:X)+1}{\log|\mathcal{F}|}. \tag{5.18}$$

To prove Item 1, suppose that a conventional algorithm $\mathcal{A}$ makes r queries to f and let X be the output of $\mathcal{A}$ after making those queries. Every query returns an element from $\mathbb{F}_p$ and thus provides at most $\log p$ bits of information about f. After r queries we then must have $I(f:X) \leq r\log p$, and so by Equation (5.18)

$$\Pr[f \neq X] \geq 1 - \frac{r\log p + 1}{\log|\mathcal{F}|}.$$

In order for this error probability to be bounded away from 1, $\mathcal{A}$ must make $\Omega\big(\log|\mathcal{F}|/(\log p)\big)$ queries as wished.

To prove Item 2, we interpret a quantum query algorithm $\mathcal{A}$ as a communication protocol between two players: Alice, who runs the algorithm, and Bob, who knows the function f. Each query can be regarded as an exchange of messages where Alice sends Bob $n+1$ *qudits* (say $|x\rangle|b\rangle$ for some $x \in \mathbb{F}_p^n$, $b \in \mathbb{F}_p$) and Bob sends $n+1$ qudits back to Alice (the query oracle output $|x\rangle|b+f(x)\rangle$). It follows as a consequence of Holevo's theorem [NC10, Chapter 12] that, after r such rounds of communication, Alice obtains at most $2r(n+1)\log p$ bits of information (see [Cle+13, Theorem 2] for a derivation of this result). We conclude that

$$\Pr[f \neq X] \geq 1 - \frac{2r(n+1)\log p + 1}{\log|\mathcal{F}|}.$$

This is bounded away from 1 when $r = \Omega\big(\log|\mathcal{F}|/(n\log p)\big)$, as wished. □

Using this lemma with $\mathcal{F} = \mathrm{Pol}_d(\mathbb{F}_p^n)$, the set of all polynomials $f : \mathbb{F}_p^n \to \mathbb{F}_p$ of degree at most d, we immediately obtain Montanaro's lower bounds on the conventional and quantum query complexities of learning degree-d multilinear polynomials, as we have $|\mathcal{F}| = n^d$.

We will next prove that the *same* lower bound holds even if we only require the algorithm to learn *any* polynomial Q which is at a nontrivial distance from the original polynomial P.

5.7.2. Theorem. *Let $P \in \mathrm{Pol}_d(\mathbb{F}_p^n)$ be a polynomial of degree $d \geq 2$ and let $\varepsilon > 0$. Suppose $\mathcal{A}$ is a query algorithm which, with nonnegligible probability, outputs a polynomial $Q \in \mathrm{Pol}_d(\mathbb{F}_p^n)$ such that $\delta(P,Q) \leq 1 - 1/p - \varepsilon$. Then:*

1. *If $\mathcal{A}$ is a conventional algorithm, it must make $\Omega(n^d)$ queries to P.*
2. *If $\mathcal{A}$ is a quantum algorithm, it must make $\Omega(n^{d-1})$ queries to P.*

This result is a simple consequence of the next theorem, which concerns (a special case of) the Gowers inverse theorem.

5.7.3. THEOREM. *Let $f : \mathbb{F}_p^n \to \mathbb{C}$ be a polynomial phase function of degree $d \geq 2$ and let $\varepsilon > 0$. Suppose $\mathcal{A}$ is a (conventional or quantum) query algorithm that, with nonnegligible probability, outputs a polynomial $Q \in \mathrm{Pol}_d(\mathbb{F}_p^n)$ such that*

$$\left|\mathbb{E}_{x \in \mathbb{F}_p^n} f(x) \omega_p^{Q(x)}\right| \geq \varepsilon.$$

Then:

1. *If $\mathcal{A}$ is a conventional algorithm, it must make $\Omega(n^d)$ queries to f.*
2. *If $\mathcal{A}$ is a quantum algorithm, it must make $\Omega(n^{d-1})$ queries to f.*

Proving this theorem requires the following lemma.

5.7.4. LEMMA. *Let $d \geq 2$ be an integer. For every prime p and every sufficiently large integer n, there exists a family $\mathcal{F} \subseteq (\mathbb{F}_p^n)^{\otimes d}$ of symmetric zero-diagonal tensors with size $|\mathcal{F}| \geq p^{n^d/(10d!)}$ for which*

$$\mathrm{prank}(S - T) \geq \frac{n}{10d!} \quad \textit{for all distinct } S, T \in \mathcal{F}.$$

Proof: We will show that a randomly chosen family of $p^{n^d/(10d!)}$ symmetric zero-diagonal tensors will satisfy the desired property with high probability.

We first determine the number of symmetric zero-diagonal tensors in $(\mathbb{F}_p^n)^{\otimes d}$. Note that the upper-diagonal indices $\Delta_+ = \left\{(i_1, \ldots, i_d) \in [n]^d : \, i_1 < i_2 < \cdots < i_d\right\}$ of any such tensor T, fully determine the tensor: all other entries of T follow either by symmetry from the upper diagonal entries or are zero if two indices coincide. For any index $(i_1, \ldots, i_d) \in \Delta_+$ it holds that $T(i_1, \ldots, i_d) \in \mathbb{F}_p$. As $|\Delta_+| = \binom{n}{d}$, it follows that the number of such tensors T is precisely $p^{\binom{n}{d}}$.

Next, we bound the number of tensors in $(\mathbb{F}_p^n)^{\otimes d}$ having partition rank at most r (for any given $r \in \mathbb{N}$). Per the definition of the partition rank, each such tensor admits a decomposition

$$\sum_{i=1}^{r} S_{I_i} \otimes T_{[d] \setminus I_i},$$

where $I_i \neq \emptyset$ is a proper subset of $[d]$ for each $1 \leq i \leq r$, $S_{I_i} \in (\mathbb{F}_p^n)^{\otimes I_i}$ and $T_{[d]\setminus I_i} \in (\mathbb{F}_p^n)^{\otimes [d]\setminus I_i}$. Suppose the size of the i-th set I_i is j for some $1 \leq j \leq d-1$, then there are $\binom{d}{j}$ ways of choosing the set $I_i \subset [d]$, p^{n^j} ways of choosing S_{I_i} and $p^{n^{d-j}}$ ways of choosing $T_{[d]\setminus I_i}$. It follows that the total number of such r-term decompositions is at most

$$\left(\sum_{j=1}^{d-1} \binom{d}{j} p^{n^j} p^{n^{d-j}}\right)^r \leq \left(2^d p^{2n^{d-1}}\right)^r.$$

We conclude that the number of tensors in $(\mathbb{F}_p^n)^{\otimes d}$ having partition rank at most r is at most $2^{dr}p^{2rn^{d-1}}$.

Now take $m := p^{n^d/(10d!)}$ symmetric zero-diagonal tensors $T_1, \ldots, T_m \in (\mathbb{F}_p^n)^{\otimes d}$ uniformly at random. As for any given $i \neq j$, the tensor $T_i - T_j$ is also uniformly distributed over the set of symmetric zero-diagonal tensors, it follows that

$$\Pr\big[\operatorname{prank}(T_i - T_j) \leq r\big] \leq \frac{2^{dr}p^{2rn^{d-1}}}{p^{\binom{n}{d}}} \quad \text{for all } 1 \leq i < j \leq m.$$

By the union bound, the probability that there exist two indices $i, j \in [m]$ such that $\operatorname{prank}(T_i - T_j) \leq r$ is at most

$$\sum_{1 \leq i < j \leq m} \Pr\big[\operatorname{prank}(T_i - T_j) \leq r\big] \leq \binom{m}{2} \frac{2^{dr}p^{2rn^{d-1}}}{p^{\binom{n}{d}}}.$$

If we take $r = n/(10d!)$ then this last quantity will tend to zero as n grows, and thus the random collection of tensors will satisfy the property in the statement with high probability. □

Using this lemma we can now prove Theorem 5.7.3, which gives the desired lower bound on the query complexity.

Proof of Theorem 5.7.3: Let $\mathcal{F}$ be the family of tensors promised by Lemma 5.7.4. For each tensor $T \in \mathcal{F}$, define the multilinear polynomial $P_T : \mathbb{F}_p^n \to \mathbb{F}_p$ by

$$P_T(x_1, \ldots, x_n) = \sum_{i_1 < \cdots < i_d} T(i_1, \ldots, i_d) \prod_{j=1}^{d} x_{i_j};$$

in other words, P_T denotes the homogeneous degree-d form on $\mathbb{F}_p^n$ associated with the upper-diagonal part of the tensor T. Now define $\mathcal{F}' = \{P_T : T \in \mathcal{F}\}$.

We now apply Lemma 5.7.1 to the family $\mathcal{F}'$, which by construction has size at least $p^{n^d/(10d!)}$. Hence, any query algorithm that learns a uniformly random element from $\mathcal{F}'$ must make either $\Omega(n^d)$ conventional queries or $\Omega(n^{d-1})$ quantum queries. In order to complete the proof, it suffices to show that one can infer any element $P_T \in \mathcal{F}'$ from any polynomial $Q \in \operatorname{Pol}_d(\mathbb{F}_p^n)$ such that

$$\left|\mathbb{E}_{x \in \mathbb{F}_p^n} \omega_p^{P_T(x)} \omega_p^{-Q(x)}\right| \geq \varepsilon.$$

Fix a function $P_T \in \mathcal{F}'$ and a degree-d polynomial Q such that the above inequality holds. Let S denote the partial derivative of Q, that is, let $S := \partial Q$. We can check that $\partial P_T = T$. Using the nesting property of the Gowers norms (Equation (1.17)), we now have

$$\left|\mathbb{E}_{y_1, \ldots, y_d \in \mathbb{F}_p^n} \omega_p^{T(y_1, \ldots, y_d) - S(y_1, \ldots, y_d)}\right| = \left\|\omega_p^{P_T - Q}\right\| U^{d^{2^d}} \geq \left|\mathbb{E}_{x \in \mathbb{F}_p^n} \omega_p^{P_T(x) - Q(x)}\right|^{2^d} \geq \varepsilon^{2^d}.$$

From this lower bound, we conclude that $\operatorname{arank}(T - S) \leq 2^d \log_p(1/\varepsilon)$. By the qualitative equivalence between analytic rank and partition rank (Theorem 3.7.3), it follows that (say) $\operatorname{prank}(T - S) \leq n/(30d!)$ if n is large enough.

Now we are essentially done: by assumption we have $\operatorname{prank}(T' - T) \geq n/(10d!)$ for all $T' \in \mathcal{F} \setminus \{T\}$, and by subadditivity we have

$$\operatorname{prank}(T' - T) \leq \operatorname{prank}(T' - S) + \operatorname{prank}(T - S).$$

We then conclude that $\operatorname{prank}(T' - S) \geq 2n/(30d!)$ for all $T' \neq T$ in $\mathcal{F}$ while $\operatorname{prank}(T - S) \leq n/(30d!)$, so learning $S = \partial Q$ suffices to infer $T \in \mathcal{F}$. Thus, learning $Q \in \operatorname{Pol}_d(\mathbb{F}_p^n)$ suffices to infer $P_T \in \mathcal{F}'$, concluding the proof. □

5.8 Reflections and outlook

This chapter revisited the work of Tulsiani and Wolf [TW14] on list decoding corrupted Reed-Muller codewords of degree at most two. We provided a novel proof of their work using new techniques from higher-order Fourier analysis discovered since. We specifically used an improved algorithmic version of the result by Balog–Szemerédi–Gowers (see also Theorem 5.1.7), with a simpler proof and better quantitative bounds. This algorithmic Balog–Szemerédi–Gowers theorem can also be of independent interest in non-quantum applications. Additionally, our new proof makes no use of Fourier estimation as a subroutine, giving a simpler prove and reducing the query complexity. Finally, by employing the Fourier sampling subroutine we managed to reduce the query complexity by a factor n by using quantum queries.

Multiple paths remain open to consider in follow-up work: First, the Freĭman-Ruzsa theorem gives an exponential upper bound in the doubling constant for the size of the set of small doubling, as shown in Equation (5.16). A recent result by Gowers et al. considers a related, but slightly different, situation, where they improve the upper bound to a polynomial dependence on the doubling constant [Gow+25]. However, their work does not translate to our situation and is non-algorithmic, stopping us from applying their result directly. An interesting research direction is to revisit this recent result and try to parse it to our current situation, as well as making it algorithmic.

Second, in this chapter, we restricted our attention to polynomial phase functions. It is interesting to see how our results extend to general functions $f : \mathbb{F}_p^n \to \mathbb{D}$ such that $\|f\|_{U^3} \geq \varepsilon$, for some $\varepsilon > 0$. The main reason for the restriction in our work is that the Fourier sampling subroutine requires a query to $\overline{f}$, the complex conjugate of f. For polynomial phase functions, $n-1$ queries to f correspond to a single query to $\overline{f}$. For general functions f however, we have no such equivalence. Instead, only probabilistic methods exist that query $\overline{f}$ [Ebl+23]. Using such a

probabilistic method affects both the success probability of the algorithm and the query complexity.

Third, a recent result showed an algorithmic approach to the inverse theorem for the U^4-norm, with which corrupted Reed-Muller codes of degree at most 3 can be decoded [KLT23]. Follow-up work might look at combining the techniques used by Kim, Li, and Tidor and the techniques used in this chapter. This way, we might improve the techniques in this chapter further, and we might improve the qubic Goldreich-Levin algorithm by Kim, Li, and Tidor. Additionally, we believe that the quantum Fourier sampling routine directly gives a factor n improvement in the query complexity for the qubic Goldreich-Levin algorithm.

Fourth, it is interesting to see at what other places quantum algorithms can provide improvements within higher-order Fourier analysis. In this chapter, we only use a quantum subroutine to efficiently sample from the Fourier spectrum. This direction for future research relates to the more general question of the relation of higher-order Fourier analysis and quantum algorithms.

Finally, inverse theorems of the U^k-norm are only known algorithmically for $k \leq 4$. For $k \geq 5$, quantitative bounds are only known for large field characteristics. Providing quantitative bounds for low field characteristics for any $k \geq 5$ or finding an algorithmic approach for $k \geq 4$ will be a significant breakthrough. Ideas from quantum computing might help finding such a breakthrough, this research line recently received new attention [AD24].

Part Two

LAQCC

Chapter 6

Introduction to shallow-depth computing

This chapter provides an introduction to shallow-depth quantum computing. We provide a rationale for considering such circuits and the added benefit of intermediate conventional computing. We also discuss multiple quantum states for which we later provide shallow-depth quantum circuits that prepare them.

6.1 Near-term quantum computers

Current quantum hardware is unable to carry out universal quantum computations due to the buildup of errors that occur during the computation. The magnitude of the individual error is currently above the value that the Threshold Theorem requires in order to kick-start quantum error correction and fault-tolerant quantum computation [AB97; KLZ98; Kit97; NC10, Section 10.6]. Although the experimentally achieved fidelity rates are promising and the error bounds are inching closer to the required threshold, we will have to work for the foreseeable future with quantum hardware with errors that build-up during the computation. This implies that we can only do a limited number of steps before the output of the computation has become completely uncorrelated with the intended one.

For fault-tolerant quantum computing, we repeat four steps: 1) Apply a number of single- and two-qubit quantum gates, in parallel whenever possible; 2) Perform a syndrome measurement on a subset of the qubits; 3) Perform fast conventional computations to determine which errors occurred and how to correct them; and, 4) Apply correction terms. We then repeat these four steps with the next sequence of gates. These four steps are essential for fault-tolerant quantum computing.

The starting point of this work is to use the four steps outlined above, not to carry out error correction and fault-tolerant computation, but to enhance short, constant-depth, *uncorrected* quantum circuits that perform single-qubit gates and *nearest-neighbor* two-qubit gates. Since in the long run we will have to implement error-correction and fault-tolerant computation anyhow, and this is done by such a four-step process, why not make other use of this architecture? Moreover, on some of the quantum hardware platforms, these operations are already in place. Embracing this idea we naturally arrive at the question: what is the computational power of *low-depth* quantum-conventional circuits organized as in the four steps outlined above? We thus investigate circuits that execute a small, ideally constant, number of stages, where at each stage we may apply, in parallel, single-qubit gates and *nearest-neighbor* two-qubit gates, followed by measurements, followed by low-depth conventional computations of which the outcome can control quantum gates in later stages. It is not clear, at first, whether such circuits, especially with constant depth, can do anything remotely useful. But we will see that this is indeed the case: many quantum computations can be done by such circuits in constant depth. By parallelizing quantum computations in this way, we improve the overall computational capabilities of these circuits, as we do not incur errors on qubits that are idle, simply because qubits are not idle for a very long time. Furthermore, reducing the depth of quantum circuits, at the cost of increasing width, allows the circuit to be run faster even if errors occur.

6.2 Quantum-conventional computations

The first usage of such a four-step process, not to do error correction, but to perform computations, can be found in measurement-based quantum computing.

Pham and Svore were the first to formalize the four-step process for performing computations [PS13]. They included specific hardware topologies by considering two-dimensional graphs for imposing constraints on qubit interactions. In their model, they develop circuits for particularly useful multi-qubit gates and presented the cost of these circuits in terms of the width, number of qubits, depth, number of consecutive time steps, size, and total number of non-identity operations. They use these gates to construct an algorithm that factors integers in polylogarithmic depth. Browne, Kashefi, and Perdrix showed that the main tool in the work by Pham and Svore, the fanout gate, can also be replaced by additional log-depth conventional computations in the measurement-based quantum computing setting [BKP11].

More recently, Piroli, Styliaris, and Cirac introduced a scheme to implement unitary operations involving quantum circuits combined with Local Operations and Classical Communication (LOCC) channels: LOCC-assisted quantum cir-

cuits [PSC21]. Similarly to the four-step process we just described, they allow for a short-depth quantum circuit, followed by one round of LOCC, in which auxiliary qubits are measured and local unitaries are applied based on the measurement outcomes. They show that in this model any 1D transitionally-invariant matrix-product state (MPS) with fixed bond dimension is in the same phase of matter as the trivial state. Similar ideas can be found in [TVV23; Tan+24].

We propose a new model called *Local Alternating Quantum-Classical Computations* (LAQCC), where we bound the power of both the quantum circuits (in computational power and locality) and the conventional computations following intermediate quantum circuit measurements. The outcome of the conventional computations is used to control operations in future quantum circuits. We allow for flexibility in this model by imposing different constraints on the power of both the quantum circuits and the conventional circuits as well as the number of alternations between them. Most attention will be given to LAQCC containing quantum circuits of constant depth, conventional circuits of logarithmic depth and at most a constant number of alternations between them. Any circuit constructed in this model is considered to be of constant depth. We restrict the conventional computations to be of logarithmic depth, as this is the first natural and nontrivial extension beyond constant-depth conventional computations.

The definition of LAQCC sharpens the original definition of Pham and Svore by adding constraints to the intermediate conventional computations. The bound on the conventional computations allows for a bound on the power of these circuits as a whole. The LOCC-assisted circuits of Piroli, Styliaris, and Cirac are not low-depth, as they allow for long sequential measurement-based corrections of the auxiliary qubits needed for their calculations. These measurement-based operations are considered as sequential alternations between the quantum and conventional circuits in LAQCC, resulting in increasing the total depth.

6.3 From computations to complexity

The considered LAQCC-circuits are bounded in power. Specifically, we show that $\mathsf{LAQCC} \subseteq \mathsf{QNC}^1$ and that LAQCC-circuits are unlikely to be efficiently simulatable as they contain instantaneous quantum polynomial-time (IQP)-circuits [BJS10; SB09]. An efficient convention simulation of IQP-circuits implies the collapse of the polynomial hierarchy to the third level, which is believed to be false [BMS17].

It is interesting to see how the computational power of LAQCC changes with more quantum resources, more conventional resources, or more alternations between the two. This new view asks for a complexity theoretical analysis beyond standard decision problems. Aaronson considered a relative complexity, where the complexity is measured between two given states and corresponds to the number of gates required from a given gate set to transform one state into the other

state [Aar04]. Rosenthal and Yuen, and Metger and Yuen instead consider classes based on sequences of quantum states preparable by polynomial-sized quantum circuits, where the circuits are uniformly generated by a computational class. For instance, the class PSPACE implies the class StatePSPACE [RY22; MY23].

We combine the ideas from these notions. We omit the uniformity constraint from [RY22; MY23] and define a class StateX as a states preparable by X-circuits. This notion is similar to the relative complexity from [Aar04], where one state is the $|0\rangle$-state and instead of counting the number of gates, we consider the set of states preparable by a bounded number of gates. We use this notion of state complexity to show that any state preparable by a LAQCC*-circuit –an instance of LAQCC with enhanced conventional and quantum computational power– is also preparable by a PostQPoly-circuit, the class of circuits of polynomial depth with an additional post-selection gate, see also Section 7.6.

6.4 State preparation

Even though the power of LAQCC is bounded for solving computational problems, LAQCC-circuits can still prove useful as a subroutine in other quantum algorithms. In light of the new notion of complexity, it is important to know how this new LAQCC-model performs in preparing often-used quantum states. We show that LAQCC-circuits with constant quantum depth and logarithmic conventional depth can prepare states with long-range entanglement.

Chapter 8 introduces new efficient state-preparation routines for three types of non-stabilizer states. Efficient circuits for preparing stabilizer states are already known from measurement-based quantum computing, as discussed in Section 7.3.

Section 8.2 presents a LAQCC-circuit to prepare the first non-stabilizer state: a uniform superposition over an arbitrary number of states. This circuit uses an exact version of Grover search, which returns a marked item with certainty [Lon01].

Section 8.3 gives a LAQCC-circuit for the second state: the W-state. This state corresponds to the uniform superposition over all computational basis states of Hamming weight 1 and is a natural long-range entangled state that displays entanglement fundamentally different from the Greenberger–Horne–Zeilinger (GHZ) state [DVC00]. LAQCC-type constant-depth circuits for the GHZ state were already known [PS13; PSC21]. The W-state is often used as benchmark for new quantum hardware [Häf+05; Nee+10; Gar+23]. A circuit for preparing the W-state was given in [PSC21], but this implementation requires sequentially alternating measurements followed by local unitaries, which is not considered constant depth in the LAQCC-model. We improve their protocol by giving a LAQCC-implementation of the W-state, based on an uncompress-compress method that links the one-hot representation and binary representation of integers.

The third state considered is the Dicke state, which generalizes the W-state to a superposition over all computational basis states with Hamming weight k [Dic54]. Dicke states prove useful for various applications, including quantum game theory [ÖSI07], quantum storage [BCH06; PB10], quantum error correction [Ouy14], quantum metrology [Tót12], and quantum networking [Pre+09]. Dicke states have been used as a starting state for variational optimization algorithms, most notably in Quantum Alternating Operator Ansatz [Had+19], to find solutions to problems such as Maximum k-vertex Cover [Bra+22; CEB20].

Dicke states also arise naturally in physics. The ground states of physical Hamiltonians describing one-dimensional chains, such as those states resulting from the Bethe ansatz, tend to show resemblance to Dicke states making them an ideal starting state when investigating the ground state behavior of these Hamiltonians [TDL10; Bra13; Bez+21]. For instance, the algorithm by Van Dyke et al., which prepares the Bethe ansatz eigenstates of the spin-1/2 XXZ spin chain, starts by first preparing a Dicke state [Van+21].

Efficient deterministic circuits for preparing Dicke states have been proposed by Bärtschi and Eidenbenz [BE19; BE22]. They provide a quantum circuit of depth $\mathcal{O}(k \log(\frac{n}{k}))$, allowing arbitrary connectivity, to prepare a Dicke state, which they conjecture to be optimal for constant k. We present a LAQCC-circuit that prepares the Dicke state with better depth than their conjecture already for constant k. However, this does not directly disprove their conjecture, as we allow for intermediate measurements and conventional computations. More significantly, in Section 8.4 we construct constant-depth LAQCC-circuits for $k = \mathcal{O}(\sqrt{n})$ greatly improving their conjectured lower bound. This construction extends the compress-uncompress method for the W-state combined with additional subroutines. In Section 8.5, we provide a log-depth circuit that prepares the Dicke state for arbitrary k by mapping efficiently between the factoradic representation and the combinatorial number representation of positive integers.

6.5 Introducing noise

The main idea behind LAQCC is to parallelize operations, such that qubits are idle only briefly. Near-term hardware will remain noisy and it is unclear when error-rates will decrease sufficiently far for long elaborate algorithms to be run faithfully. Some even doubt whether quantum computers can ever overcome the error threshold imposed by the Threshold Theorem [Kal16; Kal20], claiming that the error rate will scale linearly with the number of qubits, whereas they hypothesize that the effort to suppress these errors scales exponentially. As long as error rates remain above the error threshold, error-correcting techniques fail to scale quantum computers to the fault-tolerant setting.

In addition to improved quantum hardware and error-correcting codes, conven-

tional mitigation techniques can also help further reduce error rates [Kan+19; Cai+23]. Examples include zero-noise extrapolation [TBG17], dynamic decoupling [VL98; VKL99; Pok+18] and improved compilation and transpilation routines [WE16]. Our LAQCC-model also opens the way to improved quantum circuits by off-loading part of the computations to a conventional controller.

However, even with shallow-depth quantum circuits, errors cannot be ruled out. A careful analysis of the success probability of different quantum circuits is needed. This careful analysis also helps compare different quantum circuits and determine which is best for specific situations, given a problem instance and quantum hardware.

Such an analysis naturally requires a noise model that describes the decohering behavior of the qubits. Common noise models include single- and two-qubit gate errors, read-out errors, depolarizing noise, and dephasing noise. These noise models typically assume some underlying physical behavior. Naturally, a trade-off exists between the complexity of the noise model and the correspondence with the actual behavior.

6.6 Outline

The remainder of this part consists of three chapters: Chapter 7 formally introduces the class of Local Alternating Quantum-Classical Computations (LAQCC). In that chapter, we also provide some complexity theoretic results on LAQCC and show that all Clifford circuits have an equivalent LAQCC-circuit and we discuss some complex gates from literature that admit a LAQCC-implementation. Chapter 8 presents LAQCC-circuits for preparing three types of non-stabilizer quantum states: the uniform superposition over an arbitrary number of states, the W-state and the Dicke state for $k = \mathcal{O}(\sqrt{n})$. We then provide a protocol that prepares the Dicke state for any k, at the cost of a logarithmic number of alternations of quantum and conventional circuits. Chapter 9 presents an error analysis for preparing a GHZ state and preparing a W-state. We compare LAQCC-circuits for both states with standard methods and determine which protocol should work best. We also implement the protocols and compare the results found on quantum hardware with our theoretical estimates.

Chapter 7

The LAQCC-model

This chapter formally defines the LAQCC-model. We show that Clifford circuits are in a specific instance of LAQCC, as well as multiple more complex gates. We conclude this chapter with some complexity theoretical results for LAQCC and a version of LAQCC with enhanced conventional and quantum computational capabilities, a version we call LAQCC*.

7.1 Chapter overview

This chapter introduces a new computational model to bring structure to hybrid quantum-conventional circuits where quantum and conventional computations are alternated. We define a new class of *Local Alternating Quantum-Classical Computations* (LAQCC). This class alternates d times between quantum computations $\mathcal{Q}$ and conventional computations $\mathcal{C}$ and extends other commonly used classes, as those in general do not impose bounds on the conventional computations. Theorem 7.2.1 gives the formal definition.

The remainder of the chapter mainly focuses on a specific instance of this new model, where constant-depth quantum circuits are alternated with logarithmic-depth conventional computations. Unless stated otherwise, LAQCC will refer to this specific instance of LAQCC$(\mathcal{Q}, \mathcal{C}, d)$. Section 7.3 will show that every Clifford circuit has an equivalent LAQCC-circuit.

7.1.1. Lemma. *Every n-qubit Clifford unitary has an equivalent* LAQCC*-circuit.*

We prove this lemma by first noting that every Clifford unitary has an equivalent linear-depth implementation on a linear nearest-neighbor architecture and

then for these so-called Clifford-grid circuits show the existence of an equivalent LAQCC-circuit. This LAQCC-circuit requires a precomputation to account for the correction terms to be applied. However, this precomputation only depends on the intermediate measurement outcomes and is independent of the circuit itself.

We continue in Section 7.4 by discussing different quantum gates implementable with LAQCC-circuits. Most of these gates critically depend on the fanout gate, which generalizes the CNOT-gate to multiple targets. In this same section, Lemma 7.4.2 will give a version of exact Grover search not to search a database, but instead to prepare quantum states. This circuit corresponding to this lemma can prepare a uniform superposition based on a unitary that identifies target states. The next chapter will use this lemma to prepare different quantum states.

This chapter finishes with a complexity theoretical analysis of the LAQCC-model. First, Section 7.5 will bound the computational power of LAQCC: Lemma 7.5.1 shows how every LAQCC-circuit has an equivalent QNC^1-circuit. However, despite this upper bound, we still expect LAQCC-circuits to be hard to simulate in general. We support this intuition in Lemma 7.5.3 by linking LAQCC-circuits to instantaneous quantum polynomial-time (IQP)-circuits.

Section 7.6 considers an instance of $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$ with unbounded conventional computations. We will show that, even for this powerful instance, its computational power with respect to state preparation is upper bounded by circuits equipped with post-selection gates, where the quantum state is considered conditional on an auxiliary qubit being in the one state.

7.2 Model definition

We define the computational model *Local Alternating Quantum-Classical Computations* (LAQCC) as follows:

7.2.1. Definition. Let $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$ be the class of circuits such that

- every quantum layer implements a quantum circuit $Q \in \mathcal{Q}$ constrained to a grid topology;
- every conventional layer implements a conventional circuit $C \in \mathcal{C}$;
- there are d alternating layers of quantum and conventional circuits;
- after every quantum circuit Q, a subset of the qubits is measured;
- the conventional circuit receives the measurement outcomes as input;
- the conventional circuit can control quantum operations in future layers.

A circuit in $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$ is required to deterministically prepare a pure state starting from the $|0\rangle^{\otimes n}$-state.

The grid topology imposed on the quantum operations implies that qubits can only interact with their direct neighbors. A circuit in $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$ can use the output of conventional intermediate layers to control quantum operations in future layers. Thus, in some sense, information is fed forward in the circuit. Note, as conventional computations are in general significantly faster than the quantum operations, we only count the quantum operations towards the depth of the circuit, unless specified otherwise.

Even with this definition, a trade-off between the power of the quantum and conventional layers exists, as the following example shows:

7.2.2. Remark. We have the equivalence

$$\mathsf{LAQCC}(\mathsf{QPoly}(n), \mathsf{P}, \mathcal{O}(1)) = \mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{P}, \mathcal{O}(\mathrm{poly}(n))).$$

We can prove this equivalence by noting that any $\mathsf{QPoly}(n)$-circuit can be written as $\mathrm{poly}(n)$ concatenated QNC^0-circuits, with no intermediate measurements and trivial conventional computations.

Our main focus will be on a specific instance of $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$.

7.2.3. Notation. Unless specified otherwise, in the remainder, LAQCC refers to the instance $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(1))$. The quantum layers can use all single-qubit gates and the two-qubit CNOT-gate.

The class NC^1 is a natural nontrivial class beyond constant-depth complexity classes. LAQCC contains many useful gates and subroutines already, as outlined in the next section, specifically the fanout gate. Note that LAQCC equals $\mathsf{QNC}^0[\oplus]$ if the conventional computations are restricted to parity computations only.

The current definition of $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$ concerns circuits. Many quantum applications consider the potential of circuits in approximating quantum states. Approximating quantum states requires a different notion of complexity classes:

7.2.4. Definition. Let $\mathcal{H}_n$ be a Hilbert space on n qubits, then define

$$\mathsf{StateX}_{n,\varepsilon} = \{|\psi\rangle \in \mathcal{H}_n \mid \exists\, \mathsf{X}\text{-circuit } A \text{ such that } |\langle\psi| A |0\rangle^{\otimes n}| \geq \varepsilon\},$$

as the set of n-qubit quantum states that are ε-close to a state preparable by an X-circuit. Define $\mathsf{StateX}_\varepsilon = \bigcup_{n\in\mathbb{N}} \mathsf{StateX}_{n,\varepsilon}$.

This definition extends already existing ideas and definitions of state complexity [Sus18; AAS20; RY22]. Our definition shows similarities to state complexity defined in [MY23], but with the uniformity requirement dropped.

7.3 Clifford circuits in LAQCC

The idea of intermediate measurements with subsequent computations is closely related to measurement-based quantum computing. A famous result from this

field shows that all Clifford circuits can be parallelized using measurements. We now use this result to show that any Clifford circuit has a LAQCC-implementation.

This result is best understood in the teleportation-based quantum computing model [Joz06], a specific instance of measurement-based quantum computing that applies quantum operations using Bell measurements. In teleportation, qubits are measured in the Bell basis, which projects the measured qubits onto an entangled two-qubit Bell state, up to local Pauli corrections. This projection combined with an entangled Bell state teleports a quantum state between qubits. After teleportation, one needs to correct the local Pauli gate introduced by the Bell measurement. A similar process also allows to implement quantum gates.

Most gates do not commute with Pauli gates, which hinders parallelization. As Clifford circuits stabilize the Pauli group, full parallelization of the circuit is possible using parallel measurements [Joz06].

A Bell basis measurement projects two qubits on $\sum_{i\in\{0,1\}} \langle ii| \, P^{a,b} \otimes I$, where $P^{a,b} = Z^a X^b$ and $a, b \in \{0, 1\}$ correspond to the four possible measurement outcomes. Using one Bell-basis measurement, we can in parallel apply two sequential Clifford gates U_1 and U_2 on a quantum state $|\psi\rangle$ and a GHZ state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, which up to normalization gives:

$$\begin{aligned}
&\sum_{i,j\in\{0,1\}} \left[(\langle ii| \, P^{a,b} \otimes I) \otimes I\right] \left[U_1 \otimes I \otimes U_2\right] |\psi\rangle \, |jj\rangle \\
&\quad = \sum_{i,j\in\{0,1\}} \langle i| \, P^{a,b} U_1 \, |\psi\rangle \, \langle i|j\rangle \, U_2 \, |j\rangle \\
&\quad = \sum_{i\in\{0,1\}} U_2 \, |i\rangle \, \langle i| \, P^{a,b} U_1 \, |\psi\rangle = U_2 P^{a,b} U_1 \, |\psi\rangle \, .
\end{aligned}$$

As U_2 is a Clifford gate, there exists $(\hat{a}, \hat{b})$ such that $U_2 P^{a,b} U_1 = P^{\hat{a},\hat{b}} U_2 U_1$. The same argument shows that all correction terms can be postponed to the end of the computation.

7.3.1 Clifford-ladder circuit

Clifford-ladder circuits form a special type of Clifford circuits:

7.3.1. Definition (Clifford-ladder circuit). Given a set $\{U^i\}_{i=0}^{n-1}$ of two-qubit Clifford unitaries. A Clifford-ladder circuit C_{ladder} is a circuit of depth $\mathcal{O}(n)$ and width $\mathcal{O}(n)$ of the following form:

$$C_{ladder} = \prod_{i=0}^{n-1} U^i_{(i,i+1)}$$

where $U^i_{(i,i+1)}$ denotes that unitary U^i is applied on qubits i and $i+1$.

By definition, every two-qubit Clifford unitary has a constant-depth implementation. The next lemma shows that any Clifford-ladder circuit has an equivalent LAQCC-circuit.

7.3.2. Lemma. *Any Clifford-ladder circuit has an equivalent* LAQCC*-circuit of depth* $\mathcal{O}(1)$ *and width* $\mathcal{O}(n)$.

Proof: Every Clifford-ladder circuit can be made parallel using a Bell measurement and a fresh GHZ state via similar arguments as Clifford gate teleportation.

What remains to show is that an NC^1-circuit computes the Pauli-correction terms.

The i-th Bell measurement result gives Pauli error $P_i = Z^{a_i} X^{b_i}$. A Clifford-ladder circuit of size n hence has an error vector $\big(a\, b\big)$ of length $2n$. The correction terms that have to be applied have the same form: we can label every Pauli correction term by an index j, such that $\hat{P}_j = Z^{\hat{a}_j} X^{\hat{b}_j}$. This gives a correction vector $\big(\hat{a}\, \hat{b}\big)$. Note that Pauli matrices anti-commute, hence reordering them will only incur a global phase. This implies a binary linear map $M : \big(a\, b\big) \mapsto \big(\hat{a}\, \hat{b}\big)$. As matrix vector multiplication is in NC^1, this error calculation is in NC^1. Hence, every Clifford-ladder circuit has an equivalent LAQCC-circuit. □

Figure 7.1 shows a LAQCC-circuit that implements a Clifford-ladder circuit. Every two-qubit unitary is parallelized using gate teleportation. Using the Clifford commutation relations, the Pauli correction terms are pushed to the end of the computation. The caps and cups denote Bell-state measurements and Bell-state creation, respectively.

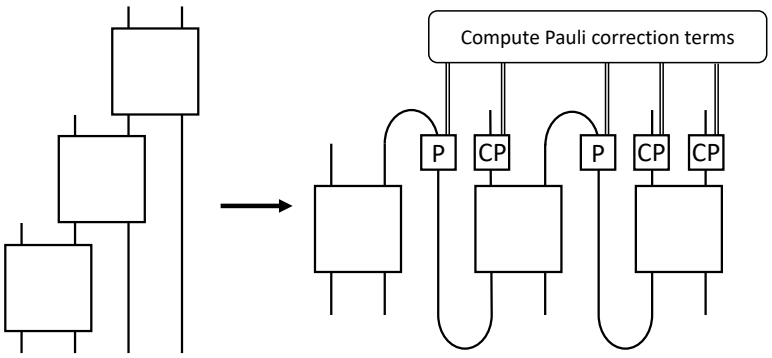


Figure 7.1: Graphical representation of Clifford-ladder circuit parallelization. Time flows upward and lines represent qubits and boxes quantum gates. Half circles represent Bell-state creation (ends pointing upwards) and Bell-state measurement (ends pointing downwards). Bell-state measurements can produce Pauli errors $P = Z^a X^b$, which are corrected by the boxes CP (corrective Pauli). Conventional computations determine the correction terms.

7.3.3. Remark. Constructing the binary linear map M is L instead of NC^1. However, as M follows from the quantum circuit and is independent from the measurement outcomes, we can compute it beforehand.

As a SWAP gate corresponds to a Clifford-ladder circuit, we see that it has an equivalent LAQCC-circuit and that we can also apply any two-qubit Clifford gate on any pair of qubits in constant depth.

A special Clifford-ladder circuit that we will use extensively is the fanout gate.

$$\text{Fanout}_n : |x\rangle |y_1\rangle \dots |y_n\rangle \mapsto |x\rangle |y_1 \oplus x\rangle \dots |y_n \oplus x\rangle . \tag{7.1}$$

This gate is a concatenation of n CNOT-gates, all with the same control qubit.

7.3.2 Clifford-grid circuit

Any Clifford unitary can be mapped to a linear-depth circuit on a linear nearest-neighbor architecture [MR18]. The most general representation of these circuits are so-called Clifford-grid circuits.

7.3.4. Definition (Clifford-grid circuit). A Clifford-grid circuit of depth d on n qubits is a circuit of the form

$$C_{grid} = \prod_{i=0}^{d} \bigotimes_{j=0}^{\frac{n}{2}} U_{i,j},$$

for Clifford unitaries $U_{i,j}$ and such that in the i-th layer, gate $U_{i,j}$ acts on qubits $2j$ and $2j+1$ if i is even, and $2j+1$ and $2j+2$ is i is odd.

The next lemma shows that Clifford-grid circuits also have an equivalent LAQCC-circuit, and with that, that any Clifford unitary has an equivalent LAQCC-circuit.

7.3.5. Lemma. *Any Clifford-grid circuit of depth $\mathcal{O}(n)$ has an equivalent* LAQCC*-circuit of depth $\mathcal{O}(1)$ and width $\mathcal{O}(n^2)$.*

Proof: Again, using gate teleportation we can parallelize the $\mathcal{O}(n^2)$ Clifford gates and obtain a LAQCC-circuit. Every Clifford gate requires a Bell measurement and a GHZ state, giving a total of $\mathcal{O}(n^2)$ qubits required.

Any Bell measurement in the circuit can incur a Pauli error, which has to be dealt with at the end of the circuit. Similar to the Clifford-ladder circuits, a vector $(a\, b)$ exists containing information on the correction terms, this time of length $\mathcal{O}(n^2)$. This vector induces a vector of correction terms $(\hat{a}\, \hat{b})$ of length $\mathcal{O}(n)$.

As Pauli errors anti-commute, there exists a binary linear map $M : (a\, b) \mapsto (\hat{a}\, \hat{b})$. The corresponding matrix is rectangular and the error-correction calculations are in NC^1, completing the proof. □

Determining M is more difficult for Clifford-grid circuits than for Clifford ladder circuits. This difficulty stems from errors having multiple possible paths that can contribute to an error in a single output qubit. The final error depends on the parity of all these paths. The computation still only depends on the measurement outcomes and not on the actual input of the circuit. Hence, M can follow from a precomputation. This precomputation is in $\oplus\mathsf{L} \subseteq \mathsf{NC}^2$ [12].

7.3.6. Corollary. *Any Clifford circuit has an equivalent* LAQCC*-circuit, provided an input-independent precomputation.*

7.4 Useful gates and routines in LAQCC

As we saw in the previous section, the fanout gate is a Clifford ladder and hence has an equivalent LAQCC-circuit using $\mathcal{O}(n)$ qubits.

Green et al. and Høyer and Špalek used the fanout gate to show how to implement a set of pairwise-commuting gates in parallel [Gre+02; HŠ05].

7.4.1. Theorem. *[Theorem 3.2 [HŠ05]] Let $\{U_i\}_{i=1}^n$ be pairwise-commuting gates on k qubits. Let K be a gate changing the basis in which all U_i are diagonal. Then, there exists a quantum circuit with fanout computing $U = \Pi_{i=1}^n U_i$ having depth $\max_{i=1}^n \mathrm{depth(U_i)} + 2 \cdot \mathrm{depth(K)} + 2$, size $\sum_{i=1}^n \mathrm{size(U_i)} + 2 \cdot \mathrm{size(K)} + 2\mathrm{nk}$, and using $(n-1)k$ auxiliary qubits.*

A common setting where this theorem can be used is if qubits in one register control operations in another register, such as the modular multiplication step in Shor's algorithm. This result was already discovered by Moore and Nilsson, though they used a $\mathcal{O}(\log n)$-deep implementation of the fanout gate based on cascading CNOT-gates [MN01]. Note that for every set of pairwise-commuting gates, a gate K changing the basis must exist [HJ85, Theorem 1.3.19].

Using the fanout gate, we obtain LAQCC-implementations for multiple complex quantum gates, as we outline below. Most gates follow from work by Takahashi and Tani [TT13]. We only consider the action on n-qubit computational basis states. The action of the gates on arbitrary states follows by linearity.

From Section 7.3 we know that any permutation $\pi \in S_n$ is in LAQCC.

$$\mathrm{Permutation}(\pi)_n : |y_1\rangle \dots |y_n\rangle \mapsto \left|y_{\pi(1)}\right\rangle \dots \left|y_{\pi(n)}\right\rangle$$

Høyer and Špalek provided the first constant-depth quantum implementation of the OR-gate with one-sided error [HŠ05]. Takahashi and Tani later improved

[12] This is expected, as efficiently simulating Clifford circuits is a complete problem for $\oplus\mathsf{L}$.

the construction of the OR-gate to be exact [TT13]. From the OR-gate we also directly arrive at the AND- and Equal$_i$-gate:

$$\begin{aligned}
\mathrm{OR}_n &: |y_1\rangle \dots |y_n\rangle |x\rangle \mapsto |y_1\rangle \dots |y_n\rangle |\mathrm{OR}_n(y) \oplus x\rangle \\
\mathrm{AND}_n &: |y_1\rangle \dots |y_n\rangle |x\rangle \mapsto |y_1\rangle \dots |y_n\rangle |\mathrm{AND}_n(y) \oplus x\rangle \\
\mathrm{Equal}_i &: |j\rangle |b\rangle \mapsto \begin{cases} |j\rangle |1 \oplus b\rangle & \text{if } j = i \\ |j\rangle |b\rangle & \text{else.} \end{cases}
\end{aligned}$$

We will omit the subscript n if the number of qubits on which they operate is clear. All three gates have width $\mathcal{O}(n \log n)$. The OR-gate follows from Theorem 1 in [TT13]. The AND-gate follows by negating all inputs and outputs of the OR-gate. The Equal$_i$-gate follows from the AND-gate, combined with a negation of the inputs that correspond to zeros in the binary representation of i.

By modifying the circuit for the OR-gate, we additionally obtain an implementation for the Exact$_t$-gate with the same circuit size (see also [HŠ05, Theorem 4.6]). This circuit outputs 1 if precisely t of the inputs are 1, and outputs 0 otherwise.

$$\mathrm{Exact}_t : |x\rangle |b\rangle \mapsto \begin{cases} |x\rangle |b \oplus 1\rangle & \text{if } |x| = t \\ |x\rangle |b\rangle & \text{else,} \end{cases}$$

As we have LAQCC-circuits for the OR- and AND-gate, we have LAQCC-circuits for all AC^0-circuits, including circuits for modular addition and circuits that check (in)equalities. All three circuits have width $\mathcal{O}(n^2)$.

$$\begin{aligned}
\mathrm{Add}_n &: |x\rangle |y\rangle \mapsto |x\rangle |y + x \bmod 2^n\rangle \\
\mathrm{Equality} &: |x\rangle |y\rangle |b\rangle \mapsto \begin{cases} |x\rangle |y\rangle |b \oplus 1\rangle & \text{if } x = y \\ |x\rangle |y\rangle |b\rangle & \text{else,} \end{cases} \\
\mathrm{Greaterthan} &: |x\rangle |y\rangle |b\rangle \mapsto \begin{cases} |x\rangle |y\rangle |b \oplus 1\rangle & \text{if } x > y \\ |x\rangle |y\rangle |b\rangle & \text{else.} \end{cases}
\end{aligned}$$

The Equality-gate is obtained by subtracting the first register from the second and computing the OR of the second register in the auxillary register and negating it. The Greaterthan-gate follows by a similar construction: Use a single auxiliar qubit in the $|0\rangle$-state to the second register and interpret y as a $n+1$-bit integer; Subtract the first register from the second and apply a CNOT-gate from this auxiliary qubit to the target output qubit. If x is larger than y, then the most significant bit of the second register (the auxillary one) is in the one state.

Høyer and Špalek also showed how fanout gates help implement quantum subroutines in constant depth, such as the quantum Fourier transform defined in Equation (1.13) with circuit size $\mathcal{O}(n^3 \log n)$ [HŠ05, Theorem 4.12]. With this

gate, we can obtain a constant-depth implementation for the Hammingweight-gate. This gate computes the binary representation of the Hamming weight of a string x in a separate register using a $\mathcal{O}(n^2)$ wide circuit [TT13, Lemma 4]. Note that this gate is strictly stronger than the Exact_t-gate, as it actually computes the Hamming weight, instead of determining if it is equal to some integer.

$$\text{Hammingweight} : |x\rangle_n |0\rangle_{\log(n)} \mapsto |x\rangle_n ||x|\rangle_{\log(n)}$$

Takahashi and Tani combined the Hammingweight-gate with the Exact_t-gate to obtain a circuit for the Threshold_t-gate with circuit width $\mathcal{O}(n \log n)$ if $t \leq \log n$ and $\mathcal{O}(n\sqrt{t \log n})$ for $\log n \leq t \leq \lceil \frac{n}{2} \rceil$ [TT13, Theorem 2]. The circuit size for $t > \lceil \frac{n}{2} \rceil$ follows from the circuit size of the Threshold_{n-t}-gate.

$$\text{Threshold}_t : |x\rangle |b\rangle \mapsto \begin{cases} |x\rangle |b \oplus 1\rangle & \text{if } \sum_i x_i \geq t \\ |x\rangle |b\rangle & \text{else.} \end{cases}$$

Now, as the Threshold_t-gate is in LAQCC, any TC^0-circuit is also in LAQCC. Additionally, the Threshold-gate can be turned into a weighted Threshold-gate by incorporating the weights in the rotation angles used in the implementation. Note that we can also compute the Threshold_t-gate using the Greaterthan-gate, where the second register is the state $|t\rangle$. However, this gives worse scaling of the circuit size than a direct Threshold_t-gate implementation.

The final routine we discuss is a consequence of a quantum algorithm by Long, which generalizes Grover's algorithm. Grover's algorithm searches a database and returns a marked item with high probability. The algorithm by Long does the same, but with unit probability [Lon01]. We modify the algorithm by Long not to search a database, but instead to prepare quantum state:

7.4.2. Lemma. *Let $\mathcal{C}$ be a set of 2^n quantum states that forms a basis for all n-qubit quantum states. Let $\mathcal{G}$ and $\mathcal{B}$ partition $\mathcal{C}$, such that $\frac{|\mathcal{G}|}{|\mathcal{C}|} = c$ is a known constant. Suppose U implements in constant depth the map*

$$U : |y\rangle |b\rangle \mapsto \begin{cases} |y\rangle |b \oplus 1\rangle & \text{if } y \in \mathcal{G} \\ |y\rangle |b\rangle & \text{if } y \in \mathcal{B} \end{cases}.$$

Then, there exists a LAQCC*-circuit that prepares the state $\frac{1}{\sqrt{|\mathcal{G}|}} \sum_{y \in \mathcal{G}} |y\rangle$.*

Proof: Write $|\mathcal{G}\rangle = \frac{1}{\sqrt{|\mathcal{G}|}} \sum_{y \in \mathcal{G}} |y\rangle$ and $|\mathcal{B}\rangle = \frac{1}{\sqrt{|\mathcal{B}|}} \sum_{y \in \mathcal{B}} |y\rangle$. Then we have $\langle \mathcal{G} | \mathcal{B} \rangle = 0$, as $\mathcal{B}$ and $\mathcal{G}$ partition $\mathcal{C}$.

The circuit by Long produces a quantum state that, upon measurement, returns a uniform random element from $\mathcal{G}$. Omitting that final measurement thus prepares the state $|\mathcal{G}\rangle$ as desired.

The corresponding circuit is indeed in LAQCC: First, prepare a uniform superposition $|s\rangle = \sum_{i=0}^{2^n-1} |i\rangle$. Then, iteratively reflect over the state $|\mathcal{B}\rangle$ using U, and reflect over the uniform superposition state $|s\rangle$. Both reflections have a LAQCC implementation and we only need to apply them a constant number of iterations.

The first reflection implements the map $I - (1 - e^{i\phi}) |\mathcal{G}\rangle \langle\mathcal{G}|$ and the second reflection the map $I - (1 - e^{i\phi}) |s\rangle \langle s|$, where ϕ depends on the number of target states and the known constant c, see also [Lon01]. The implementation for both reflections follows the same lines: Mark the state $|\mathcal{G}\rangle$ and $|s\rangle$ using an auxillary qubit; Use an R_Z-gate to apply the phase $(1 - e^{i\phi})$; Uncompute the auxiliary qubit. The first reflection uses the unitary U, which by definition has a constant-depth implementation. The second reflection uses an Exact_0-gate with negated output and all inputs conjugated by Hadamard gates.

The circuit only uses operations from LAQCC. Additionally, the total number of iterations is $\mathcal{O}(\sqrt{N/m}) = \mathcal{O}(\sqrt{|\mathcal{C}|/|\mathcal{G}|}) = \mathcal{O}(\sqrt{c})$, from which the constant depth of the whole circuit follows. □

7.5 Complexity results for LAQCC

LAQCC seems more powerful than conventional computing alone. Yet, most power seems to come from the conventional intermediate computations. This section first proves an upper bound on the power of LAQCC, before showing that LAQCC-circuits are unlikely to be efficiently simulatable.

Any LAQCC-circuit A can be written as a composition of quantum layers U_i, measurements M_i and conventional layers C_i:

$$A = M_k U_k C_k \dots M_i U_i C_i \dots M_1 U_1 C_1, \tag{7.2}$$

for some constant k. Any unitary U_i corresponds to a QNC^0-circuit and any C_i corresponds to an NC^1-circuit. The measurements M_i can measure any subset of the qubits. By the principle of deferred measurements, we can always postpone them to the end of the circuit using CNOT-gates and fresh auxiliary qubits [NC10, Section 4.4], which gives the following lemma.

7.5.1. Lemma. *For any* LAQCC*-circuit A there is a* QNC^1*-circuit B without intermediate measurements that outputs the same state as A.*

Proof: The LAQCC-circuit A contains the layers C_i that use the intermediate measurement results as input. Delaying the measurements until the end of the circuit by applying a CNOT-gate from the qubit to a fresh auxiliary qubit, replaces the conventional output wires by quantum wires.

Next, we can replace any NC^1-circuit by a QNC^1-circuit that uses at most $\text{poly}(n)$ auxiliary qubits: 1) Replace all OR-gates by AND- and NOT-gates; 2) Replace

all AND-gates by Toffoli-gates, where the third input is a clean auxiliary qubit; and, 3) Replace all NOT-gates by X-gates.

Hence, for every conventional circuit C_i, we can obtain a quantum circuit V_i, such that V_i computes the same output. The quantum circuit $B = U_k V_k \dots U_1 V_1$ then proves the lemma. □

The power of LAQCC is thus bounded. This makes one wonder if these circuits admit efficient conventional simulations. Even in that case, LAQCC-circuits can still have value as "fast" alternatives for state preparation routines. By linking the class of Instantaneous Quantum Polynomial-time (IQP)-circuits, first introduced in [SB09], to LAQCC-circuits, we can show that such an efficient simulation for *all* LAQCC-circuits is unlikely.

7.5.2. Definition (Definition 2 [NM14]). An IQP-circuit on n qubits is a quantum circuit such that: each gate in the circuit is diagonal in the Pauli-Z basis, the input state is $|+\rangle^{\otimes n}$, and the output is the result of a measurement in the Pauli-X basis on a specified set of output qubits.

7.5.3. Lemma. *Any* IQP*-circuit has an equivalent* LAQCC*-circuit.*

Proof: Gates diagonal in the Pauli-Z basis commute in the Pauli-X basis. Therefore, prepare $|+\rangle^{\otimes n}$ by a single layer of Hadamard gates on all qubits. Then, parallelize the gates from the IQP-circuit using Theorem 7.4.1. Measurements in the Pauli-X basis correspond to a Hadamard gate followed by a measurement in the Pauli-Z basis. This construction gives the equivalent LAQCC-circuit. □

Bremner, Jozsa, and Shepherd showed that efficient weak simulation of all possible IQP-circuits up to a small multiplicative error implies a collapse of the polynomial hierarchy [BJS10]. As a direct corollary, LAQCC-circuits are unlikely to be efficiently simulatable using conventional computers, unless the polynomial hierarchy collapses. A circuit family is weakly simulatable if its output distribution can be sampled by purely conventional means in polynomial time given the description of the circuit family.

7.5.4. Lemma (Corollary 1 [BJS10]). *If the output probability distributions generated by uniform families of* IQP*-circuits could be weakly simulated to within multiplicative error* $1 \leq c < \sqrt{2}$ *then the polynomial hierarchy collapses to the third level, in particular,* $\mathsf{PH} = \Delta_3^p$.

7.6 Complexity results for powerful LAQCC

Based on the previous section, LAQCC seems more powerful than conventional computing alone. The conventional intermediate computations provide LAQCC with significant power. This added power raises the question what other capabilities are possible if we extend the conventional computational power beyond NC^1.

In this section, we consider a different instance of $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$ where we allow for more powerful quantum and conventional computations, as well as more alternations between them, giving us the class LAQCC^*.

7.6.1. Notation. The class LAQCC^* refers to the instance

$$\mathsf{LAQCC}(\mathsf{QPoly}(n), \mathsf{ALL}, \mathrm{poly}(n)).$$

That is, LAQCC^* refers to the class of circuits that alternate a polynomial number of times between polynomial-sized quantum circuits and arbitrary powerful conventional computations, together with feed forward of the conventional information to future quantum operations. The quantum computations are restricted to all single-qubit gates and the two-qubit CNOT-gate.

Note that LAQCC^* can trivially solve decision problems by simply using the unbounded conventional computational power and then loading the result in a quantum state. Therefore, we are mainly interested in the computational power of LAQCC^* with respect to state preparation. The definition of LAQCC^* directly defines $\mathsf{StateLAQCC}^*_\varepsilon$ for $\varepsilon \geq 0$.

7.6.2. Remark. For any nonzero ε, we can restrict ourselves to finite universal gate sets. The Solovay-Kitaev theorem [Kit97; NC10] states that any multi-qubit unitary can be approximated to within precision δ by a quantum circuit with size depending only on δ. Hence, any LAQCC^*-circuit using a continuous gate set can be approximated by a LAQCC^*-circuit using a universal finite gate set.

We will compare LAQCC^* with the class of polynomial-sized quantum circuits equipped with an additional post-selection gate. Even though this gate has no physical realization, as measurements outcomes cannot be chosen, it is a useful gate in analyzing different algorithms.

7.6.3. Definition. The class $\mathsf{PostQPoly}$ consists of all QPoly-circuits that produce a quantum state $\alpha |1\rangle |\psi\rangle + \beta |0\rangle |\perp\rangle$, where $|\psi\rangle$ is the relevant quantum state and $|\perp\rangle$ can be any arbitrary quantum state. The success of the quantum circuit is conditioned on the first qubit being in the $|1\rangle$-state.

We can decompose any LAQCC^*-circuit as

$$\Pi_{i=0}^{\mathrm{poly}(n)} M_i U_i(y_i) C_i(x_i) \, |0\rangle^{\otimes \mathrm{poly}(n)} \, . \tag{7.3}$$

Again, M_i denotes the i-th measurement, and U_i and C_i denote the i-th quantum and conventional layer, respectively. The $x_i \in \{0,1\}^*$ denote the outcome of M_i and $y_i \in \{0,1\}^*$ the bitstring outputted by C_i. Note, all x_i and y_i have length at most polynomial in n. We will use this decomposition to prove a similar inclusion as in Lemma 7.5.1.

7.6.4. Theorem. *For every $\varepsilon \geq 0$ it holds that*

$$\mathsf{StateLAQCC}^*_\varepsilon \subseteq \mathsf{StatePostQPoly}_\varepsilon$$

Proof: Fix $\varepsilon \geq 0$ and a positive integer n and let $|\psi\rangle \in \mathsf{StateLAQCC}^*_\varepsilon$. By definition, there exists a LAQCC^*-circuit $A = \Pi_{i=0}^{\mathrm{poly}(n)} M_i U_i(y_i) C_i(x_i)$ that prepares a state $|\phi\rangle$ such that $|\langle\phi|\psi\rangle| \geq \varepsilon$.

Let $B = \Pi_{i=0}^{\mathrm{poly}(n)} \mathrm{Equal}_{x_i}(x_i) U_i(y_i) |0\rangle^{\otimes \mathrm{poly}(n)}$, where the y_i are hardwired inputs to the quantum circuits, based on measurement outcome x_i. This $\mathsf{PostQPoly}$-circuit prepares $|\psi\rangle$ with unit probability, conditioned on the auxillary qubit being in the $|1\rangle$-state.

The Equal_{x_i}-gate replaces the measurement M_i by checking if the qubits that would be measured are in $|x_i\rangle$-state. The outcomes are stored in an auxiliary qubit. This gives $\mathrm{poly}(n)$ auxiliary qubits, one for each replaced measurement layer. Then apply an AND-gate on these auxiliary qubits and store the result in another auxiliary qubit. Condition the output of the circuit is then conditioned on this final auxiliary qubit. □

Note that every measurement has at least one possible output string x_i. Figure 7.2 sketches the idea behind the proof.

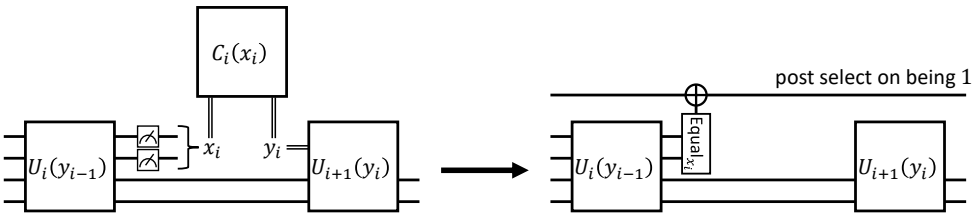


Figure 7.2: Graphical representation of transforming a LAQCC^*-circuit for generating $|\psi\rangle$ into a $\mathsf{PostQPoly}$-circuit.

7.7 Reflections and outlook

This chapter introduced the LAQCC-model that alternates between conventional and quantum operations. The LAQCC-model generalizes already existing models by explicitly bounding the conventional intermediate computing power.

We focused on a specific instance of LAQCC that alternates between QNC^0-circuits and NC^1-circuits. This instance generalizes the $\mathsf{QNC}^0[\oplus]$-circuits encountered in the first part of this work. The conventional computations help to implement a quantum fanout gate, which is often used as a building block for constructing more complex gates. Future research can focus on what other problems we can offload to the conventional computer. Another promising direction is to explore LAQCC-instances with increased quantum or conventional computational power, or more alternations between the two. Finally, future research can help quantify the relation between different LAQCC-instances and other known complexity classes.

At the end of the chapter, we took a first step in this direction by considering a powerful instance called LAQCC^*. This instance alternates between arbitrary conventional computations and polynomial-sized quantum circuits. This class naturally solves any solvable decision problem using the unbounded conventional computations. With respect to state preparation, we however have the inclusion

$$\mathsf{StateLAQCC}^*_\varepsilon \subseteq \mathsf{StatePostQPoly}_\varepsilon.$$

We conjecture that this inclusion is strict for any $\varepsilon > 0$.

As a first step towards proving this conjecture, one might consider an oracular separation, using, for instance, the oracle Aaronson and Kuperberg used to separate QMA and QCMA [AK07]. For every state $|\psi\rangle$, they define an oracle O_ψ such that $O_\psi : |\psi\rangle = -|\psi\rangle$, while for every $|\phi\rangle \perp |\psi\rangle$ $O_\psi : |\phi\rangle = |\phi\rangle$.

Interestingly, a PostQPoly-circuit can prepare $|\psi\rangle$ exactly using a single oracle query. There always exists a computational basis state $|i\rangle$ with nonzero overlap with $|\psi\rangle$. Starting with that state and implementing O_ψ as a bit-flip oracle gives

$$\alpha \left|\psi^\perp\right\rangle |0\rangle + \beta |\psi\rangle |1\rangle ,$$

for some $\alpha, \beta \in \mathbb{C}$ and $\beta \neq 0$. Conditioned on the auxiliary qubit, this circuit indeed prepares $|\psi\rangle$.

To prove that LAQCC^*-circuits cannot approximate $|\psi\rangle$, we have two promising directions. The first uses ε-nets to show that the number of possible quantum states is double exponential in n [NC10, Section 4.5.4]. However, counting the number of LAQCC^*-circuits, we find that $\mathsf{StateLAQCC}^*_\varepsilon$ contains only an exponential number of quantum states. When counting the number of LAQCC^*-circuits, we can omit the intermediate computations as their output is only used to control future quantum operations. We can then use that the oracle is close to the identity operator, and oracle calls only marginally help to approximate $|\psi\rangle$. We can omit an oracle call at the cost of a small error. However, if we want $\varepsilon = 1/\mathrm{poly}(n)$, we find that these small errors stack too much with the multiple oracle calls.

The second direction uses quantum state based on incompressible strings. These strings have no short program that returns them. Kolmogorov complexity dictates that strings exist that admit no efficient description [LV08]. Let z be an incompressible string of length $|z| = N(3+\varepsilon)n$ for $N = 2^n$. Interpret z as a concatenation of N substrings of length $(3+\varepsilon)n$ each: $z = [z_0, z_1 \ldots z_i \ldots, z_{N-1}]$. The following $(4+\varepsilon)n$-qubit quantum state encodes z

$$|z\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle |z_i\rangle .$$

A LAQCC^*-circuit that prepares $|z\rangle$ now gives an efficient description of a routine that returns the string z, something which we conjectured impossible due to z being incompressible. It is again unclear how oracle calls can help approximate $|z\rangle$.

Chapter 8

State preparation by LAQCC

This chapter proposes LAQCC-circuits for three types of non-stabilizer states. First, we give a LAQCC-circuit to prepare a uniform superposition over an arbitrary number of states. Next, we use an uncompress-compress method to prepare the W-state. We conclude with two protocols to prepare the Dicke state, uniform superpositions of all n-bitstrings with Hamming weight k. The first protocol prepares in constant depth Dicke states for $k = \mathcal{O}(\sqrt{n})$. The second protocol prepares Dicke states for all k in logarithmic depth.

8.1 Chapter overview

The previous chapter introduced the LAQCC-model, where quantum and conventional computations alternate to solve a problem or prepare some quantum state. This chapter considers the capability of this LAQCC-model to prepare quantum states often used in other algorithms and for benchmarking quantum devices.

First, Section 8.2 gives in Theorem 8.2.1 a LAQCC-circuit to prepare a uniform superposition of an arbitrary number of states. The circuit uses Lemma 7.4.2. This state is often used as starting state by other algorithms.

Section 8.3 gives a LAQCC-circuit for preparing the W-state. This circuit uses the circuit for the uniform superposition. Theorem 8.3.1 shows how to obtain the W-state using an uncompress-compress method that efficiently maps between the binary number representation and the one-hot number representation. The first representation corresponds to the initial uniform superposition obtained from Theorem 8.2.1, whereas the second representation corresponds to the W-state.

Next, Section 8.4 uses the circuit for the W-state to prepare the Dicke-(n, k) state for $k = \mathcal{O}(\sqrt{n})$. Specifically, the circuit for the Dicke state uses k parallel instances of the LAQCC-circuit for the W-state, all using the same target register. Additional LAQCC-operations ensure that all parallel instances target different qubits in the target register. The result is summarized in Theorem 8.4.1.

Section 8.5 then presents a method to prepare the Dicke state for arbitrary k. The protocol, summarized in Theorem 8.5.1, maps between the combinatorial number representation (which uniquely labels Dicke states for fixed k) and the factoradic representation. As this mapping requires logarithmic depth, we use a new instance called LAQCC-LOG.

8.2 Uniform superposition of size q

Many quantum algorithms start with a uniform superposition. Often, a uniform superposition over 2^n states is used. This state is obtained by applying a Hadamard gate on all n qubits. In practice, uniform superpositions over fewer states might be more useful. However, preparing the superposition

$$\frac{1}{\sqrt{q}} \sum_{i=0}^{q-1} |i\rangle$$

for arbitrary q is difficult. As a first step, we can consider a probabilistic approach:

1. Create a superposition $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$ with $n = \lceil \log_2(q) \rceil$ qubits;
2. Mark the states $i < q$ using an auxiliary qubit;
3. Measure the auxiliary qubit.

With probability at least one half, the correct state is prepared. The measurement result indicates whether we have the correct state. The second step uses a Greaterthan-gate, using $\mathcal{O}(\lceil \log_2(q) \rceil^2)$ qubits, with the second register initialized in the $|q\rangle$-state. However, this approach is non-unitary due to the measurements.

We can modify this probabilistic circuit into a deterministic circuit. The next section uses the deterministic circuit to prepare the W-state. Note that if q is a power of 2, the circuit simplifies to a layer of single-qubit Hadamard gates.

8.2.1. Theorem. *There exists a* LAQCC*-circuit that prepares the uniform superposition on q states. This circuit uses $\mathcal{O}(\lceil \log_2(q) \rceil^2)$ qubits.*

Proof: Let $n = \lceil \log_2(q) \rceil$ and define $\mathcal{G} = \{i \mid 0 \leq i < q\}$ and $\mathcal{B} = \{i \mid q \leq i < 2^n\}$. Applying Lemma 7.4.2 with the unitary

$$U_q : |y\rangle |b\rangle \mapsto \begin{cases} |y\rangle |b \oplus 1\rangle & \text{if } y < q, \\ |y\rangle |b\rangle & \text{if } y \geq q, \end{cases}$$

gives the desired state. This circuit is in LAQCC, as the unitary corresponds to a Greaterthan-gate. □

8.3 W-state

This section considers the W_n-state (or the W-state, if n is implicit) and presents a LAQCC-circuit that prepares it. The W-state is a uniform superposition over all n-bit strings of Hamming weight 1:

$$|W\rangle = \frac{1}{\sqrt{n}} \sum_i |e_i\rangle ,$$

where $|e_i\rangle$ is the state with a one on the i-th position and zeros elsewhere. The main theorem of this section gives a LAQCC-circuit that prepares the W-state:

8.3.1. Theorem. *There exists a* LAQCC*-circuit that prepares* $|W\rangle$*. This circuit uses* $\mathcal{O}(n \log(n) \log\log(n))$ *qubits placed in a grid of size* $n \times \mathcal{O}(\log(n) \log\log(n))$*.*

A natural bijection exists between the states $|e_i\rangle$ and a uniform superposition over n states. If we can find a unitary that implements this bijection, we can prepare the W-state. Hence, we wish to find a unitary that implements the map

$$|i\rangle_{\log n} |0\rangle_n \mapsto |0\rangle_{\log n} |e_i\rangle_n . \tag{8.1}$$

This unitary maps the binary representation of i (using $\log n$ qubits) to its one-hot representation (using n qubits). We refer to registers with a binary representation as index registers and those with a one-hot representation as system registers. The mapping between these two representations naturally defines two operations:

$$\textbf{Uncompress}: \quad |i\rangle_{\log n} |0\rangle_n \mapsto |i\rangle_{\log n} |e_i\rangle_n , \tag{8.2}$$

$$\textbf{Compress}: \quad |i\rangle_{\log n} |e_i\rangle_n \mapsto |0\rangle_{\log n} |e_i\rangle_n . \tag{8.3}$$

Combining these two operations implements the unitary given in Equation (8.1). Both operations have a LAQCC-implementation, as the next two lemmas show.

8.3.2. Lemma. *There exists a* LAQCC*-circuit using* $\mathcal{O}(n \log(n) \log\log(n))$ *qubits that, for any* n*, implements the* ***Uncompress*** *operation:*

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log n} |0\rangle_n \mapsto \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log n} |e_i\rangle_n . \tag{8.4}$$

The qubits are placed in a grid of size $n \times \mathcal{O}(\log(n) \log\log(n))$*.*

Proof: Interpret the right-most column of the grid as the system qubits. Adjacent to each system qubit are $\log n$ index qubits. Additionally, every row has access to

$\mathcal{O}(\log\log n)$ auxiliary qubits. The LAQCC-circuit then consists of the following three steps (where we only show the index and system qubits):

$$\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|i\rangle_{\log n}|0\rangle_{\log n}^{\otimes n-1}|0\rangle_n \xrightarrow{(1)} \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|i\rangle_{\log n}^{\otimes n}|0\rangle_n$$
$$\xrightarrow{(2)} \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|i\rangle_{\log n}^{\otimes n}|e_i\rangle_n$$
$$\xrightarrow{(3)} \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|i\rangle_{\log n}|0\rangle_{\log n}^{\otimes n-1}|e_i\rangle_n$$

Step (1) Use fanout gates to copy the first index register; Step (2) In parallel for all i, apply Equal_i-gates from the i-th index register to the corresponding system qubit. This creates the state $|e_i\rangle$ in the system register; Step (3) Use fanout gates to disentangle the index registers. □

Figure 8.1 shows the steps of the **Uncompress** operation graphically for $n = 4$. We explicitly leave out auxiliary qubits in the shown circuits.

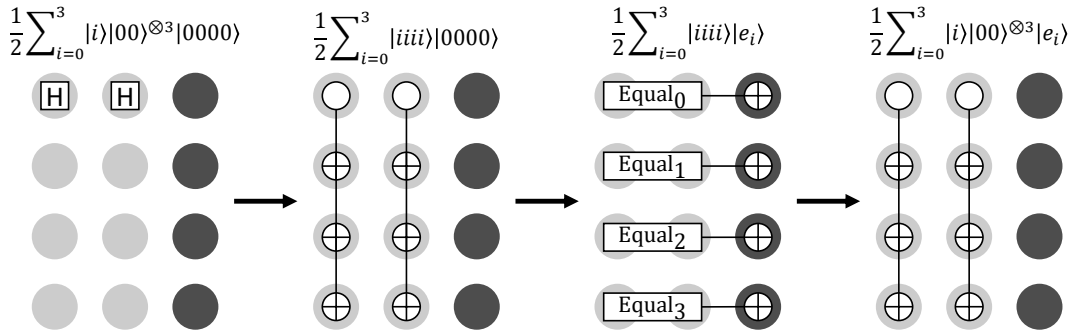


Figure 8.1: Circuit for the **Uncompress** operation for $n = 4$. Shown is a grid of 12 qubits: 8 light gray index qubits, and 4 dark gray system qubits. Each grid represents a single time step.

8.3.3. Lemma. *There exists a* LAQCC*-circuit that, for any n, implements the* ***Compress*** *operation:*

$$\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|i\rangle_{\log n}|e_i\rangle_n \mapsto \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}|0\rangle_{\log n}|e_i\rangle_n\,.$$

This circuit uses $\mathcal{O}(n\log n)$ qubits placed in a grid pattern of size $n \times \log n$.

Proof: The **Compress** operation uses parallel $CNOT$-gates controlled by the system register to uncompute the index registers. These $CNOT$-gates commute

for different indices in the system register and hence by Theorem 7.4.1 they can be applied in parallel. The **Compress** operation consists of five steps:

$$\begin{aligned}
\frac{1}{\sqrt{n}}\sum_{i=0}^{n}|i\rangle_{\log n}|0\rangle_{\log n}^{\otimes n-1}|e_i\rangle_n &\xrightarrow{(1)} \frac{1}{n}\sum_{i,j=0}^{n}(-1)^{i\cdot j}|j\rangle_{\log n}|0\rangle_{\log n}^{\otimes n-1}|e_i\rangle_n \\
&\xrightarrow{(2)} \frac{1}{n}\sum_{i,j=0}^{n}(-1)^{i\cdot j}|j\rangle_{\log n}^{\otimes n}|e_i\rangle_n \\
&\xrightarrow{(3)} \frac{1}{n}\sum_{i,j=0}^{n}|j\rangle_{\log n}^{\otimes n}|e_i\rangle_n \\
&\xrightarrow{(4)} \frac{1}{n}\sum_{i,j=0}^{n}|j\rangle_{\log n}|0\rangle_{\log n}^{\otimes n-1}|e_i\rangle_n \\
&\xrightarrow{(5)} \frac{1}{\sqrt{n}}\sum_{i=0}^{n}|0\rangle_{\log n}^{\otimes n}|e_i\rangle_n
\end{aligned}$$

Step (1) Apply Hadamard gates to the first index register, changing to the Hadamard basis, in which the NOT-gate is diagonal; Step (2) Use fanout gates to copy the first index register; Step (3) Apply controlled-Z-gates, controlled by system qubit i and with targets those index qubits in the i-th index register that correspond to the ones in the binary representation of i; Step (4) Use fanout gates to disentangle the index registers; and, Step (5) Apply Hadamard gates to clean the index register.

The controlled-Z-gates depend on the binary representation of i and precisely cancel the phase factors $(-1)^{i\cdot j}$ and disentangle the index registers from the system register. □

Figure 8.2 shows the steps of the **Compress** operation graphically for $n = 4$. We again omit the auxiliary qubits from the shown circuits.

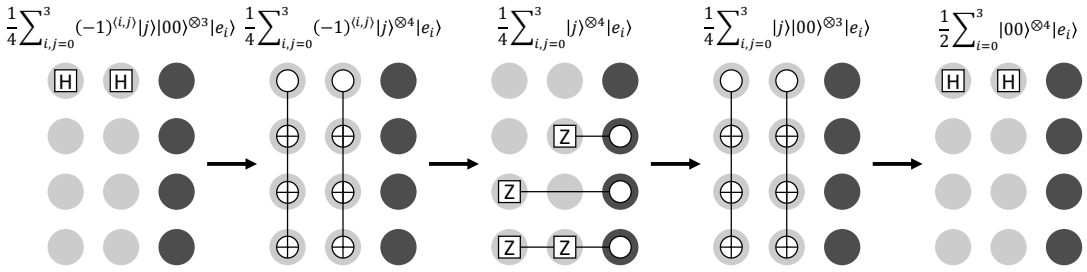


Figure 8.2: Circuit for the **Compress** operation for $n = 4$. Shown is a grid of 12 qubits: 8 light gray index qubits, and 4 dark gray system qubits. Each grid represents a single time step of the **Compress** operation.

We now combine these two lemmas to prove the main result of this section.

Proof of Theorem 8.3.1: The LAQCC-circuit consists of the following three steps:

$$
\begin{aligned}
|0\rangle_{\log n}^{\otimes n} |0\rangle_n &\to \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |0\rangle^{\otimes n-1} |0\rangle , && \text{(Theorem 8.2.1)} \\
&\to \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |0\rangle^{\otimes n-1} |e_i\rangle , && \text{(Lemma 8.3.2)} \\
&\to \frac{1}{\sqrt{n}} \sum_{i=0}^{n} |0\rangle^{\otimes n} |e_i\rangle . && \text{(Lemma 8.3.3)}
\end{aligned}
$$

□

8.4 Dicke states for small k

This section generalizes Theorem 8.3.1 to Dicke-(n, k) states, uniform superpositions over n-bit strings of Hamming weight k:

$$
|D_k^n\rangle = \binom{n}{k}^{-1/2} \sum_{x \in \{0,1\}^n : |x| = k} |x\rangle . \tag{8.5}
$$

Other works also considered efficient state preparation routines for Dicke states. Bärtschi and Eidenbenz for instance presented a circuit of linear depth and width in n that prepares the Dicke-(n, k) state, independent of k [BE19]. Their method uses a recursive relation for the Dicke-state:

$$
|D_k^n\rangle = \alpha_{k,n} \left|D_k^{n-1}\right\rangle \otimes |0\rangle + \beta_{k,n} \left|D_{k-1}^{n-1}\right\rangle \otimes |1\rangle .
$$

Our method, the main result of this section, instead extends the compress-uncompress method used for the W-state to small k:

8.4.1. Theorem. *For any n and $k = \mathcal{O}(\sqrt{n})$, there exists a* LAQCC*-circuit preparing the Dicke-(n, k) state, $|D_k^n\rangle$, using $\mathcal{O}(n^3)$ qubits.*

The main idea is to apply k parallel **Uncompress** operations. The parallel **Uncompress** operations might target the same system register qubit. We thus have to filter these cases. Following the lines of the birthday paradox, we find that coinciding indices occur rarely for $k = \mathcal{O}(\sqrt{n})$. Lemma 7.4.2 allows boosting the amplitude of the correct states to obtain the Dicke state.

We again assume the qubits to be laid out in a grid, the main difference being that we now have k index registers denoted by subscripts $i \in [k]$, each consisting of $\log n$ qubits. The first grid in Figure 8.3 gives an example of the initial layout of the grid for $n = 4$ and $k = 2$.

The algorithm consists of four operations:

$$
\begin{aligned}
&|0\rangle_1 \dots |0\rangle_k |0\rangle \\
&\quad \to \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle_1 \dots |j_k\rangle_k |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle \qquad \textbf{(Filling)} \qquad (8.6)\\
&\quad \to \sqrt{\frac{(n-k)!}{n!}} \sum_{j_1 \neq \dots \neq j_k} |j_1\rangle_1 \dots |j_k\rangle_k |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle \qquad \textbf{(Filtering)} \qquad (8.7)\\
&\quad \to \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1<\dots<j_k} |j_1\rangle_1 \dots |j_k\rangle_k |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle \qquad \textbf{(Ordering)} \qquad (8.8)\\
&\quad \to \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1<\dots<j_k} |0\rangle_1 \dots |0\rangle_k |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle \qquad \textbf{(Cleaning)} \qquad (8.9)
\end{aligned}
$$

The following four lemmas will give a LAQCC-circuit for each of these operations. First, the **Filling** operation shown in Equation (8.6) fills the system register.

8.4.2. Lemma. *There exists a* LAQCC*-circuit for the* ***Filling*** *operation using* $\mathcal{O}(kn \log(n) \log\log(n))$ *qubits.*

Proof: The **Filling** operation uses k parallel instances of the **Uncompress** operation from Lemma 8.3.2. These k operations commute and hence have a parallel implementation by Theorem 7.4.1. This parallelization requires k parallel system registers. In total, the **Filling** operation consists of five steps:

$$
\begin{aligned}
&|0\rangle_1 \dots |0\rangle_k |0\rangle^{\otimes k} \\
&\quad \xrightarrow{(1)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle_1 \dots |j_k\rangle_k \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} |l\rangle |0\rangle^{\otimes k-1} \\
&\quad \xrightarrow{(2)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle_1 \dots |j_k\rangle_k \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} |l\rangle^{\otimes k} \\
&\quad \xrightarrow{(3)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle_1 \dots |j_k\rangle_k \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} (-1)^{(2^{j_1}+\dots+2^{j_k})\cdot l} |l\rangle^{\otimes k} \\
&\quad \xrightarrow{(4)} \frac{1}{n^{k/2}} \sum_{j_1,;s j_k=0}^{n-1} |j_1\rangle_1 \dots |j_k\rangle_k \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} (-1)^{(2^{j_1}+\dots+2^{j_k})\cdot l} |l\rangle |0\rangle^{\otimes k-1} \\
&\quad \xrightarrow{(5)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle_1 \dots |j_k\rangle_k |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle |0\rangle^{\otimes k-1}
\end{aligned}
$$

Step (1) Use Theorem 8.2.1 to create uniform superpositions of size n in all index registers and of size 2^n in the system register; Step (2) Copy the registers using fanout gates; Step (3) Apply phase-versions of Equal$_{j_i}$-gates for all j_i.

These phase-versions apply a Z-gate instead of an X-gate to the target qubit; Step (4) Use fanout gates to clean the copies of the system register; Step (5) Apply Hadamard gates on the system register. This gives the sum of the one-hot representations of the index registers in the system register.

We use kn Equal_i-gates, each using $\mathcal{O}(\log(n)\log\log(n))$ qubits. This gives a total of $\mathcal{O}(kn\log(n)\log\log(n))$ qubits required to implement the **Filling** operation. □

Figure 8.3 shows these five steps graphically. We omitted the auxiliary qubits and included only the auxiliary system register for clarity.

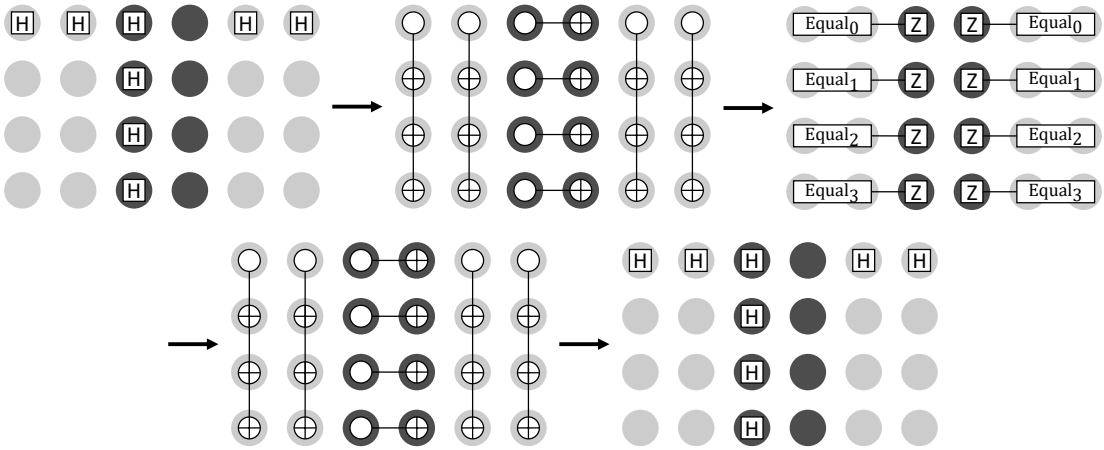


Figure 8.3: Circuit for the **Filling** operation for $n = 4$ and $k = 2$. Shown is a grid of 24 qubits: 16 light gray index qubits and 8 dark gray system qubits. Each grid represents a single step.

The **Filling** operation can have multiple index registers with the same value, resulting in a state in the system register with Hamming weight less than k. The next **Filtering** operation shown in Equation (8.7) removes these collisions, ensuring that every state in the system register has Hamming weight k.

8.4.3. Lemma. *There exists a* LAQCC*-circuit that implements the* ***Filtering*** *operation using* $\mathcal{O}(n\log n)$ *qubits.*

Proof: The final state produced by the **Filling** operation splits in two substates, based on whether all indices j_i are different. Let

$$|\psi\rangle = \sum_{j_1\neq\ldots\neq j_k} |j_1\rangle_1 \ldots |j_k\rangle_k |e_{j_1}\oplus\cdots\oplus e_{j_k}\rangle ,$$

then we can write the output of the **Filling** operation as

$$\frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle_1 \ldots |j_k\rangle_k |e_{j_1}\oplus\cdots\oplus e_{j_k}\rangle = \alpha|\psi\rangle + \beta|\psi^\perp\rangle ,$$

where $\left|\psi^{\perp}\right\rangle$ contains all terms with overlapping indices j_i. As a result, the states in $\left|\psi^{\perp}\right\rangle$ have Hamming weight strictly smaller than k. Of all n^k possible values of the index registers, only $n!/(n-k)!$ have all index register values different.

By the birthday paradox we have that for any integers n and $k < \frac{n}{2}$ it holds that

$$\frac{n!}{n^k(n-k)!} > e^{-\frac{2k^2}{n}}. \tag{8.10}$$

In fact, we have the simple computation

$$\begin{aligned}
\frac{n!}{n^k(n-k)!} &= e^{\sum_{i=1}^{k} \log(1-\frac{i}{n})} \\
&> e^{\sum_{i=1}^{k} \frac{-i}{n-i}} \\
&> e^{-\sum_{i=1}^{k} \frac{i}{n-k}} \\
&> e^{-\frac{k^2}{n-k}} \\
&> e^{-\frac{2k^2}{n}},
\end{aligned}$$

where in the first inequality we use that $\log(1+x) \geq \frac{x}{1+x}$ for $x > -1$. As a result, we obtain the lower bound $|\alpha|^2 > \exp(\frac{-2k^2}{n})$, which is constant for $k = \mathcal{O}(\sqrt{n})$.

Now note that a unitary U_{flag} that flag $|\psi\rangle$ is implemented using an Exact_t-gate. The output of this gate is 1, precisely if t of the inputs are 1, which happens precisely if every index register has a different value. We thus have the map

$$\begin{aligned}
&\frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle_1 \ldots |j_k\rangle_k |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle |0\rangle \\
&\quad \to \frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle_1 \ldots |j_k\rangle_k |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle \left|\mathbf{1}_{|e_{j_1}\oplus\cdots\oplus e_{j_k}|=k}\right\rangle \\
&\quad = \alpha |\psi\rangle |1\rangle + \beta \left|\psi^{\perp}\right\rangle |0\rangle .
\end{aligned}$$

Applying Lemma 7.4.2 with unitary U_{flag} allows us to amplify α to 1 in constant depth and using $\mathcal{O}(n \log n)$ qubits, to give the state

$$\sqrt{\frac{(n-k)!}{(n)!}} \sum_{j_1 \neq \ldots \neq j_k} |j_1\rangle_1 \ldots |j_k\rangle_k |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle .$$

□

All states in the system register now have the correct Hamming weight. However, it is remains unclear which index register corresponds to which 1 in the system register, as a permutation of the index registers gives the same state in the system register. The **Ordering** operation (Equation (8.8)) resolves this redundancy by imposing an order on the index registers.

8.4.4. Lemma. *There exists a* LAQCC*-circuit that implements the* ***Ordering*** *operation using* $\mathcal{O}(k^2 \log^2 n)$ *qubits.*

Proof: Start with $k-1$ auxiliary qubits per index register. For index register i, use the first $j < i$ auxiliary qubits to store the outcome of a Greaterthan-gate evaluated on the j-th and i-th index registers. For the auxiliary qubits $j \geq i$, store the outcome of a Greaterthan-gate evaluated on the $j+1$-th and i-th index registers. Every Greaterthan-gate uses $\mathcal{O}(\log^2 n)$ qubits, and a total of $\mathcal{O}(k^2)$ Greaterthan-gates are evaluated. We use fanout gates on the index registers to parallelize the Greaterthan-gates. Omitting a scaling factor and the system register, these parallel Greaterthan-gates implement the map

$$\sum_{j_1 \neq \ldots \neq j_k} |j_1\rangle_1^{\otimes k-1} |0\rangle^{\otimes k-1} \ldots |j_k\rangle_k^{\otimes k-1} |0\rangle^{\otimes k-1}$$
$$\rightarrow \sum_{j_1 \neq \ldots \neq j_k} \left[|j_1\rangle_1^{\otimes k} |\mathbf{1}_{j_1 > j_2}\rangle \ldots |\mathbf{1}_{j_1 > j_k}\rangle \right] \ldots \left[|j_k\rangle_k^{\otimes k} |\mathbf{1}_{j_k > j_1}\rangle \ldots \left|\mathbf{1}_{j_k > j_{k-1}}\right\rangle \right].$$

Each $\mathbf{1}_{j_k > j_{k'}}$ acts as an indicator variable for the event $\{j_k > j_{k'}\}$.

Next, measure the auxiliary qubits and add all measurement outcomes to determine the Hamming weights. These Hamming weights impose an ordering on the index registers. Assume without loss of generality that the imposed ordering is $j_1 < \ldots < j_k$, otherwise a LAQCC-circuit exists to SWAP the index registers accordingly. Adding the measurement results is in AC^0 and sorting the measurement results is in TC^0 [Siu+93]. As both classes are subsets of NC^1, the conventional controller can impose an ordering and then determine and apply the permutation to obtain the desired state

$$\binom{n}{k}^{-1/2} \sum_{j_1 < \cdots < j_k} |j_1\rangle_1 \ldots |j_k\rangle_k |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle .$$

□

Note that we can also compute the Hammingweight-gate and the permutation in superposition. As quantum operations are in general expensive, it is better to offload these comptuations to the conventional controller.

Similar to the **Compress** operation in the W-state protocol, we now have to clean the index registers. The **Cleaning** operation, shown in Equation (8.9), cleans the index registers, taking into account the added ordering of the index registers. Suppose the m-th qubit of the system register is a 1. If this is the t-th one in the string, then the t-th index register has value m. Computing the Hamming weight of the first $t-1$ qubits gives precisely the required information to know which index register to uncompute.

8.4.5. Lemma. *There exists a* LAQCC*-circuit that implements the* ***Cleaning*** *operation using* $\mathcal{O}(n^3)$ *qubits.*

Proof: Compute in parallel, the Hamming weight of the first i system register qubits, for $i \in [n-1]$. Provided that the j-th system register qubit is in the $|1\rangle$-state, the Hamming weight of the first $j-1$ qubits determines which of the index registers has to be uncomputed. Compute the Hammingweight-gate requires $\mathcal{O}(n^2)$ qubits. We store the Hamming weight in an auxiliary register of size $\log k$.

Using these auxiliary registers, we can then clean the k index registers, similar to the **Compress** method of Lemma 8.3.3, combined with the information stored in the auxiliary registers. Cleaning index register j consist of fives steps and requires a copy of the system register and a copy of each of the n auxiliary Hammingweight-registers.

1. Apply Hadamard gates to bring the index register to phase space;
2. Apply fanout gates to copy the index register;
3. Apply n parallel Z-gates on the index register to clean it. These Z-gates are controlled by one system register qubit and by the Hammingweight-register, such that there have been precisely $j-1$ ones in the system register already;
4. Apply fanout gates to clean the index register copies;
5. Apply Hadamard gates to reset the index register qubits to the $|0\rangle$-state.

The cost of the Hammingweight-computation dominates the requirements on the number of qubits, giving the $\mathcal{O}(n^3)$ qubit requirement for the **Cleaning** step. □

Figure 8.4 shows the circuit to compute the Hamming weight of the system register qubits for $n = 4$. Note that we need only three computations, as system qubit i uses the Hamming weight of the first $i-1$ system qubits. We omitted the index registers, and instead show the auxiliary registers that hold the Hammingweight values. Figure 8.5 shows the steps taken to clean the j-th index register. The **Cleaning** operation uses the Hamming weight information.

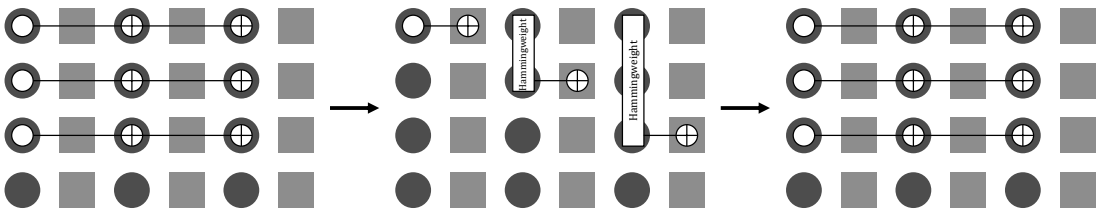


Figure 8.4: Circuit to compute the Hamming weight of the first qubits. The dark gray dots represent system qubits, and the gray squares denote auxiliary registers of $\log k$ qubits each.

The proof of the theorem now follows by combining these four lemmas:

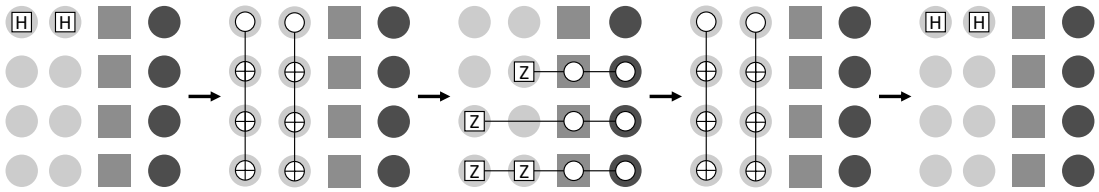


Figure 8.5: Circuit to clean index register j. The black dots represent qubits in the system register and the light gray dots the index register and its copies. The gray squares denote auxiliary registers of $\log k$ qubits each. Each grid represents a single time step.

Proof of Theorem 8.4.1: The circuit that prepares the Dicke-state for $k = \mathcal{O}(\sqrt{n})$ follows directly from the circuits presented in Lemmas 8.4.2 to 8.4.5. These circuits use at most $\mathcal{O}(n^3)$ qubits completing the proof. □

Bärtschi and Eidenbenz posed a conjecture on the optimal depth of quantum circuits that prepare the Dicke-(n, k) state. They presented an algorithm for generating Dicke-(n, k) states in depth $\mathcal{O}(k \log \frac{n}{k})$, given all-to-all connectivity, and conjectured that this scaling is optimal for constant k. Our result shows that there is a LAQCC-circuit in this regime, but does not disprove their conjecture. However, as the circuits shown here are also accessible in QNC^1 by Lemma 7.5.1, there also exists a quantum circuit of depth $\mathcal{O}(\log n)$ for $k = \mathcal{O}(\sqrt{n})$ that achieves better scaling for non-constant k, i.e., when $k = \omega(1)$.

8.5 Dicke states for all k

The previous section presented a LAQCC-circuit that prepares the Dicke-(n, k) state for $k = \mathcal{O}(\sqrt{n})$. That method does not scale to larger k, because the birthday-paradox argument fails for larger k. This section shows how to prepare Dicke-(n, k) states for all k using a logarithmic number of alternations. We let LAQCC-LOG refer to the instance $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(\log n))$.

One way of studying the creation of Dicke states is by looking at efficient algorithms that convert numbers from one representation to another. An example is the **Uncompress-Compress** method in the W-state protocol that converts numbers from a binary representation to a one-hot representation. Dicke states generalize the W-state, hence the one-hot representation no longer suffices for preparing the state. Instead, we use a construction based on number conversion between the combinatorial and the factoradic representation. The main theorem of this section uses an efficient conversion between these two representations.

8.5.1. Theorem. *For $k \leq n$ positive integers, there exists a* LAQCC-LOG*-circuit for preparing the Dicke-(n, k) state using $\mathcal{O}(\mathrm{poly}(n))$ qubits.*

The next sections introduce the two representations, as well as quantum circuits that convert one representation into the other.

8.5.1 Combinatorial number system

Beckenbach showed that any integer $m \geq 0$ can be written as a sum of k binomial coefficients [Bec64]. This decomposition is even unique for fixed k, as the next lemma shows:

8.5.2. Lemma ([Bec64]). *For all integers $m \geq 0$ and $k \geq 1$, there exists a unique decreasing sequence of integers $c_k, c_{k-1}, \ldots, c_1$ with $c_j > c_{j-1}$ and $c_1 \geq 0$ such that*

$$m = \binom{c_k}{k} + \binom{c_{k-1}}{k-1} \cdots \binom{c_1}{1} = \sum_{i=1}^{k} \binom{c_i}{i}.$$

From this lemma, we naturally arrive at a definition for the combinatorial number representation:

8.5.3. Definition. Let $k \in \mathbb{N}$ be a constant. Any integer $m \in \mathbb{N}$ can be represent by a unique string of integers $(c_k, c_{k-1} \ldots, c_1)$, according to Lemma 8.5.2. We call this string the *index representation* of m and denote it by $m^{indx(k)}$.

The bit string containing k ones at indices $(c_k, \ldots, c_1)$ is the m-th bit string with k ones in the lexicographical order. This bit string is called the *combinatatorial representation*. We denote the m-th bit string with k ones as $m^{comb(k)}$.

The W-state protocol used the conversion between the binary representation of a number m and its combinatorial representation $m^{comb(1)}$.

Preparing the Dicke-(n, k) state requires the following steps. Given positive integers k and n, prepare the superposition

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{i=0}^{\binom{n}{k}-1} |i\rangle |0\rangle ;$$

Map between the representation i and $i^{comb(k)}$ to obtain the state

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{i=0}^{\binom{n}{k}-1} |i\rangle \left|i^{comb(k)}\right\rangle ;$$

Map between the representation $i^{comb(k)}$ and i to clean the label register

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{i=0}^{\binom{n}{k}-1} |0\rangle \left|i^{comb(k)}\right\rangle = |D_k^n\rangle .$$

The second mapping uses Lemma 8.5.2. This calculation requires iterative multiplication and addition, both of which are in TC^0 [Vol99], hence this calculation is in TC^0.

The first mapping, from binary to combinatorial representation for given k, has a simple greedy iterative algorithm: On input m, find the biggest c_k such that $m \geq \binom{c_k}{k}$ and subtract this from m: $\tilde{m} = m - \binom{c_k}{k}$. This gives c_k and a residual $\tilde{m}$. Repeat this process for $\tilde{m}$: Find the largest c_j such that $\tilde{m} \geq \binom{c_j}{j}$ and update residual $\tilde{m} = \tilde{m} - \binom{c_j}{j}$, until all c_j are found.

This greedy algorithm is inherently linear in k as it requires all previously found $\{c_i\}_{i=j}^{k}$ to find c_{j-1}. Hence, it is not immediately obvious if and how to achieve this mapping in constant or even logarithmic depth.

8.5.2 Factoradic representation

A number representation closely related to the combinatorial number representation is the *factoradic representation*. This number system uses factorials instead of binomials to represent integers.

8.5.4. Definition. A sequence $y = (y_{n-1}, y_{n-2}, \ldots, y_0)$ of integers, such that $j \geq y_j \geq 0$ is called an *n-factoradic* (or simply a factoradic). The elements of an n-factoradic are called *n-digits*. An n-factoradic y can represent a number m between 0 and $n! - 1$, in the following way

$$m = \sum_{j}^{n} y_j \cdot j!. \tag{8.11}$$

We call the n-factoradic y the *factoradic representation* of an integer $m \leq n! - 1$. Denote $\mathrm{Fact}(n)$ as the set of all n-factoradics.

The following lemma shows that every integer $m \in \{0, \ldots, n! - 1\}$ has a unique factoradic representation given by Equation (8.11).

8.5.5. Lemma. *For $k \geq 0$ it holds that:*

$$\sum_{i=0}^{k} i \cdot i! = (k+1)! - 1.$$

Proof: We prove the lemma by induction. The base case $k = 0$ follows naturally, as $0 \cdot 0! = 1! - 1$.

Now assume the lemma holds for some integer j, then

$$\sum_{i=0}^{j+1} i \cdot i! = (j+1) \cdot (j+1)! + \sum_{i=0}^{j} i \cdot i = (j+1) \cdot (j+1)! + (j+1)! - 1 = (j+2)! - 1,$$

which completes the proof. □

As the factoradic representation represents a unique integer, we can use it as a number system. The next section shows how to convert between the factoradic representation and the combinatorial number representation.

8.5.3 Mapping between representations

The next lemma gives a logspace algorithm to convert a factoradic representation to its combinatorial representation.

8.5.6. LEMMA. *There is a logspace algorithm $\mathcal{A}$ that, given $k \in \{0, \ldots, n\}$, and a uniformly random n-factoradic, outputs a uniformly random n-bit string of Hamming weight k.*

Proof: Let $y = (y_{n-1}, \ldots, y_0)$ be an n-factoradic. The logspace algorithm $\mathcal{A}$ will then output an n-bit string $y^{comb(k)} = y_{n-1}^{comb(k)} \ldots y_0^{comb(k)} \in \{0,1\}^n$ of Hamming weight k, one bit at a time, from left to right, according to the following rule: Let $H_{>n-j} = \sum_{i=n-j+1}^{n-1} y_i^{comb(k)}$ be the Hamming weight of the bits produced before we reach bit $n-j$. Then $y_{n-j}^{comb(k)}$ is given by:

$$(\mathcal{A}(y))_{n-j} = y_{n-j}^{comb(k)} = \begin{cases} 1 & \text{if } y_{n-j} < k - H_{>n-j} \\ 0 & \text{otherwise} \end{cases}. \tag{8.12}$$

This conversion requires comparing an n-digit with a constant and the Hamming weight of a bit string. The only information that $\mathcal{A}$ needs to remember, as it goes from bit $n-j+1$ to bit $n-j$, is the Hamming weight $H_{>n-j}$ of the bits it produced so far, which requires logarithmic storage space.

Note that the number of factoradic n-digit strings that map to the same combinatorial bit string is always $k!(n-k)!$: Let $y^{comb(k)} \in \{0,1\}^n$ have Hamming weight k. For any bit position $y_{n-j}^{comb(k)}$, there are $n-j+1-(k-H_{>n-j})$ possible choices for the n-digit y_{n-j} such that $y_{n-j}^{comb(k)} = 0$. For the leftmost index $n-j$ such that $y_{n-j}^{comb(k)} = 0$, it holds that $H_{>n-j} = j-1$. There then are $n-k$ possible n-digits y_{n-j} such that $y_{n-j}^{comb(k)} = 0$. Then, for the second leftmost index $n-j$ such that $y_{n-j}^{comb(k)} = 0$ it holds that $H_{>n-j} = j-2$, hence there are $n-k-1$ possible n-digits y_{n-j} such that $y_{n-j}^{comb(k)} = 0$. This argument repeats for all k indices. This results in $(n-k)!$ different possible choices for the $(n-k)$-many n-digits such that $y^{comb(k)} = 0$.

Similarly, for the leftmost position $n-j$ where $y_{n-j}^{comb(k)} = 1$, there are k possible choices for the n-digit y_{n-j} that given $y_{n-j}^{comb(k)} = 1$. The second leftmost position $n-j$ gives $k-1$ possible choices, and so forth, for a total of $k!$ possible choices of the k-many n-digits where $y^{comb(k)} = 1$.

We conclude that, for every n-bit string $y^{comb(k)} \in \{0,1\}^n$ of Hamming weight k, there are exactly $k!(n-k)!$ n-factoradics y such that $\mathcal{A}(y) = y^{comb(k)}$. □

This lemma gave a logspace algorithm to convert a uniformly random n-factoradic to a uniformly random n-bit string of Hamming weight k, for any k. As logspace is contained in TC^1 [Joh90], the algorithm $\mathcal{A}$ has a parallel log-depth implementation, provided one has access to threshold gates. As LAQCC-LOG contains threshold gates, any TC^1-circuit has an equivalent LAQCC-LOG-circuit:

8.5.7. Corollary. *The following map can be implemented in* LAQCC-LOG*.*

$$\frac{1}{\sqrt{n!}} \sum_{y \in Fact(n)} |y\rangle\,|0\rangle \to \frac{1}{\sqrt{n!}} \sum_{y \in Fact(n)} |y\rangle\,|\mathcal{A}(y)\rangle\,.$$

The next lemma gives a TC^0-circuit that implements the inverse of $\mathcal{A}$.

8.5.8. Lemma. *Given as input an n-bit string $y^{comb(k)}$ of Hamming weight k, a uniformly-random k-factoradic, and a uniformly-random $(n-k)$-factoradic. There exists a TC^0-circuit that on this input, when given this input, outputs a uniformly random n-factoradic y such that $\mathcal{A}(y) = y^{comb(k)}$.*

Proof: The conversion can be done in parallel, generating an n-digit for every bit in $y^{comb(k)} = (y_{n-1} \dots y_0) \in \{0,1\}^n$. Recall that we are given as input a uniformly-random k-factoradic $X_{k-1}, \dots, X_0$ and a uniformly-random $(n-k)$-factoradic $Z_{n-k-1}, \dots, Z_0$.

For every bit position $n-j$, for $j \in [n]$, calculate the Hamming weight of the bits from $n-j+1$ to $n-1$: $H_{>n-j} = \sum_{i=j+1}^{n-1} y_i^{comb(k)}$. Recall that iterated addition is in TC^0 [Vol99].

If $y_{n-j}^{comb(k)} = 1$, set $y_{n-j} = X_{k-H_{>n-j}}$. This gives a uniform random n-digit between 0 and $k - H_{>n-j} - 1$. If $y_{n-j}^{comb(k)} = 0$, set $y_{n-j} = k - H_{>n-j} + Z_{n-k-H_{>n-j}}$. Note that this gives a uniform random n-digit between $k - H_{>n-j}$ and $n-j$. By construction, it now follows that $\mathcal{A}(y) = y^{comb(k)}$. Computing each n-digit in this way requires summation and indexing, both of which are in AC^0. □

8.5.9. Remark. The above algorithm establishes a bijection $(y^{comb(k)}, Z, X) \leftrightarrow y$ between triples $(y^{comb(k)}, Z, X)$, where $y^{comb(k)} \in \{0,1\}^n$ has Hamming weight k, $Z \in \mathrm{Fact}(n-k)$ and $X \in \mathrm{Fact}(k)$, and an n-factoradic $y \in \mathrm{Fact}(n)$. Let $(\mathcal{A}(y), \mathcal{Z}(y), \mathcal{X}(y))$ be the image of an n-factoradic y under this bijection. The previous lemma shows that one can compute y from $(y^{comb(k)}, Z, X)$ in TC^0. Then the map $(\mathcal{A}(y), y) \mapsto (\mathcal{A}(y), y, \mathcal{Z}(y), \mathcal{X}(y))$ is also in TC^0. Indeed, to find $\mathcal{Z}(y)$ and $\mathcal{X}(y)$, we need only invert the two defining equalities $y_{n-j} = X_{k-H_{>n-j}}$ and $y_{n-j} = k - H_{>n-j} + Z_{n-k-H_{>n-j}}$.

8.5.10. Corollary. *There exists a* LAQCC*-circuit for the map*

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{y^{comb(k)}} |0\rangle \left|y^{comb(k)}\right\rangle \to \frac{1}{\sqrt{n!}} \sum_{y \in Fact(n)} |y\rangle \left|\mathcal{A}(y)\right\rangle ,$$

where $y^{comb(k)}$ *ranges over all* n*-bit strings of Hamming weight* k*.*

Proof: The map consists of three steps:

$$\begin{aligned}
&\binom{n}{k}^{-\frac{1}{2}} \sum_{y^{comb(k)}} \left|y^{comb(k)}\right\rangle |0\rangle |0\rangle |0\rangle \\
\xrightarrow{(1)}\ &\binom{n}{k}^{-\frac{1}{2}} \sum_{y^{comb(k)}} \left|y^{comb(k)}\right\rangle \left(\bigotimes_{j=0}^{n-k-1} \sum_{i=0}^{j} |i\rangle\right) \left(\bigotimes_{j=0}^{k-1} \sum_{i=0}^{j} |i\rangle\right) |0\rangle \\
=\ &\frac{1}{\sqrt{n!}} \sum_{y^{comb(k)}} \left|y^{comb(k)}\right\rangle \left(\sum_{Z \in \mathrm{Fact}(n-k)} |Z\rangle\right) \left(\sum_{X \in \mathrm{Fact}(k)} |X\rangle\right) |0\rangle \\
\xrightarrow{(2)}\ &\frac{1}{\sqrt{n!}} \sum_{y \in \mathrm{Fact}(n)} |\mathcal{A}(y)\rangle \left|\hat{Z}(y)\right\rangle \left|\hat{X}(y)\right\rangle |y\rangle \\
\xrightarrow{(3)}\ &\frac{1}{\sqrt{n!}} \sum_{y \in \mathrm{Fact}(n)} |\mathcal{A}(y)\rangle |0\rangle |0\rangle |y\rangle
\end{aligned}$$

The first step uses Theorem 8.2.1 to prepare a uniform superposition over all n-factoradics. The second step follows directly from Lemma 8.5.8. The third step follows directly from Remark 8.5.9. The above steps implicitly use that the inverse of a LAQCC-operation is also in LAQCC. Even though it is unclear if this inverse-property holds in general, it does hold for the used LAQCC-operations. The only non-reversible operations used are measurements in the fanout gate construction. The fanout gate itself is reversible. □

We now have all necessary tools to prove the main theorem of this section, a LAQCC-LOG-circuit that prepares the Dicke-(n,k) state for arbitrary $k \leq n$.

Proof of Theorem 8.5.1: The LAQCC-LOG-circuit combines the circuits resulting from Lemma 8.5.6 and Lemma 8.5.8 and consists of three steps:

$$\begin{aligned}
|0\rangle^{\otimes n \log n} |0\rangle^{\otimes n} &\xrightarrow{(1)} \frac{1}{\sqrt{n!}} \left(\bigotimes_{j=0}^{n-1} \sum_{i=0}^{j} |i\rangle\right) |0\rangle^{\otimes n} = \sum_{y \in \mathrm{Fact}(n)} |y\rangle |0\rangle^{\otimes n} \\
&\xrightarrow{(2)} \frac{1}{\sqrt{n!}} \sum_{y \in \mathrm{Fact}(n)} |y\rangle |\mathcal{A}(y)\rangle \\
&\xrightarrow{(3)} \binom{n}{k}^{-\frac{1}{2}} \sum_{y \in \mathrm{Fact}(n)} |0\rangle |\mathcal{A}(y)\rangle = |D_k^n\rangle .
\end{aligned}$$

The first step uses Theorem 8.2.1 to prepare a uniform superposition over all n-factoradics. The second step uses the circuit given in Corollary 8.5.7. The third step uses the inverse of the circuit given in Corollary 8.5.10. □

8.6 Reflections and outlook

This chapter introduced novel state-preparation protocols for three types on non-stabilizer states: The uniform superposition, the W-state and the Dicke state. For the uniform superposition, we used an exact version of Grover's search routine and LAQCC-circuits given in the previous chapter.

For the W-state, we used an uncompress-compress method, which efficiently maps between the binary representation and the one-hot representation of integers. We then extended this method to also work for Dicke-(n, k) states for $k = \mathcal{O}(\sqrt{n})$. For arbitrary k, we used a mapping between two different representations, namely the combinatorial number representation and the factoradic representation. This second method does require a logarithmic number of alternations between the quantum and conventional circuits.

Future research can focus on improving the circuits given in this chapter. For the Dicke state for instance, one can try and find a constant-depth circuit that works for any k. Alternatively, one might try to implement an uncompress-compress method similar to the one used for the W-state. The current approach is inspired by that method, but uses multiple intermediate steps for it.

Another direction for future research is to focus on preparing other types of quantum states. These states can be close to the states we considered, such as many-body scar states [Buh+24, Section 4.5] or weighted superpositions of different Dicke states [Bon+23], or states with certain characteristics, such as sparsity or some symmetry [Sun+23; RLR24; LL24; MTS24].

Next, our protocols can improve in the circuit size. They are of constant depth, but require wide circuits to be implemented. Similarly, the constants in the protocols might be improvable.

Finally, it is worthwhile to show that states in general can or cannot be prepared by certain LAQCC-instances. Such a result could then indicate if our protocol for the Dicke state for arbitrary k is optimal. Such a result can additionally imply separations between certain complexity classes.

Chapter 9

Error analysis

This chapter compares LAQCC-circuits that prepare the GHZ state and the W-state with standard implementations. We determine the theoretical success probability for both and compare them.

9.1 Chapter overview

One of the main reasons to consider the class LAQCC was to offload computations to a conventional controller and assure that qubits are idle only briefly. As an example, when preparing a GHZ state on n qubits, a standard approach using an all-to-all connectivity uses n qubits, n quantum gates, and has depth $\lceil \log(n) \rceil + 1$, for a total circuit size of $\mathcal{O}(n \log n)$. The LAQCC-approach that prepares the GHZ state instead uses $2n - 1$ qubits, around $4n$ quantum gates, and has constant depth, for a total circuit size of $\mathcal{O}(n)$.

Based on the number of qubits required and the number of quantum gates applied, the standard approach seems favorable. On the other hand, based on the circuit size, the LAQCC-circuit seems favorable. The LAQCC-circuit is significantly more dense, meaning that qubits are idle only briefly.

In this chapter, we explore which approach works best, based on the success probabilities of the quantum gates and the probabilities that qubits decohere while idling. This chapter presents a first-order comparison between standard circuits and LAQCC-circuits for preparing the GHZ state and the W-state.

First, in Section 9.2, we introduce the error models used in the remainder of this chapter. Next, Section 9.3 will provide the analysis for the GHZ state. For the

GHZ state we consider a standard approach using an all-to-all connectivity and using a linear nearest-neighbor connectivity, and we consider a LAQCC-approach. Additionally, we also derive expressions for the success probability of hybrid versions of a standard approach combined with a LAQCC-approach. Next, we compare these success probabilities and, using some reductions, obtain a bound on when one protocol performs better than the other in terms of the success probabilities of the individual terms. Theorem 9.3.1 summarizes the results on which protocol performs best when.

Next, Section 9.4 gives an implementation of both a standard approach and a LAQCC-circuit on quantum hardware. We use these implementations to compare the performance of the two approaches with each other and with the outcomes expected by the theoretical analysis.

Afterwards, in Section 9.5, we perform the same theoretical analysis for the W-state as we did the the GHZ state. First, we derive an expression for the success probability of a standard approach and of a LAQCC-approach. Next, we compare these approaches to determine under what circumstances, the LAQCC-approach performs best. Due to the qubit requirement of the LAQCC-approach, we have not implemented this circuit.

9.2 Error model

In the next section, we analyze the protocols by comparing the success probability of these circuits in terms of the success probabilities of the individual terms in the circuits. Computing this overall success probability requires an error model that describes the behavior of the quantum circuit in case qubits decohere.

We consider a worst-case error model where every error corresponds to an independent uniformly random gate being applied to the qubit(s)[13]. Qubits can decohere both while idling and while a gate is applied to it. In both cases, a random unitary is applied to that qubit.

In this error model, the probability that two errors cancel each other is 0. Suppose an error B occurred and suppose a gate G is applied after which another error D occurs. The probability that the two errors cancel corresponds to the probability of the event

$$D \cdot G \cdot B = G.$$

This means that we should have $D = GB^\dagger G^\dagger$. As both sides of the equality correspond to independent Haar-random unitaries, the probability that they are equal is zero. Note that taking $G = I$, the identity gate, corresponds to the situation where no gate is applied and a qubit was instead idling.

[13]Formally, the unitaries are taken uniformly at random with respect to the Haar measure.

An error for measurement gates corresponds to a situation where the measurement outcome differs from the actually measured state. In the LAQCC-circuits, the measurement outcomes are used to control future quantum gates. Hence, if the incorrect control is used, the wrong gate can be applied. We thus 'need' one specific error to cancel he measurement error and have the circuit still outputting the correct quantum state.

In this error model, all errors are independent. The circuits we consider in this section have at most polynomial size. The set of errors that can possibly be canceled is therefore also polynomial in size. Yet, the group of unitary matrices from which the errors are sampled corresponds to a continuum of matrices. That is, the group of unitary matrices is parametrized by three continuous parameters. Hence, the set of unitary matrices that can cancel a previous error has measure 0. As a result, in the subsequent analysis, we only have to derive expressions for the situation where all gates succeed and no errors occur at all.

We determine the success probability of quantum circuits based on the success probability of the elementary operations used in the circuit. These elementary operations are single-qubit gates, two-qubit CNOT-gates, and measurements, as well as idling terms for other qubits during these operations. Additionally, we have the idling term during the conventional computations. Table 9.1 summarizes these terms together with their meaning.

Success term	Probability that ...
p_s	a single-qubit gate succeeds
p_d	a two-qubit gate succeeds
p_m	a measurement returns the correct value
p_c	the intermediate computation returns the correct value
p_{ix}	a qubit remains coherent, while idling during an x-operation

Table 9.1: The considered success probabilities when analyzing state preparation protocols in the remainder of this chapter.

We explicitly assume that intermediate conventional computations always return the correct answer, hence $p_c = 1$. Note that p_d relates to two qubits, whereas p_{id} concerns individual qubits.

In most quantum devices, multi-qubit gates have to be decomposed in single-qubit gates and CNOT-gates. Some devices have the controlled-Z-gate as native gate, instead of the CNOT-gate. The choice for one of these two gates negligibly affects the following success probabilities.

In the remainder, we will use the term P_X to denote the success probability of a subroutine X. Similarly, P_{iX} denotes the probability that a qubit remains coherent while idling during the execution of X.

Some protocols use controlled-U-gates, for some single-qubit gate U. Every such gate admits a decomposition in two CNOT-gates and three single-qubit gates, as Figure 9.1 shows graphically (see also [NC10, Corollary 4.2]). The gates A, B, and C are chosen such that the product ABC is the identity and $AXBXC = U$.

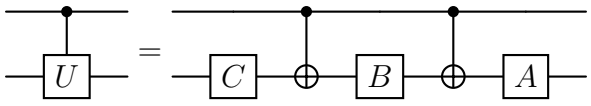


Figure 9.1: Identity for the controlled-U gate.

Using the success probabilities from Table 9.1 in the circuit shown in Figure 9.1, we obtain a success probability for the controlled-U-gate of

$$P_{cU} = p_s^3 p_{is}^3 p_d^2. \tag{9.1}$$

In the remainder of this chapter, we derive success probabilities for preparing the GHZ state and the W-state. The derived expressions depend on up to seven success probability variables and the input size n. Hence, for a first-order estimate of the relative theoretical performance of these protocols, we make some assumptions about the magnitude of the different terms.

We distinguish between cheap operations, where the probability of an error is low, and expensive operations, where the probability of an error is higher. This distinction applies to both the operations themselves and the corresponding idling times. In most hardware realizations, single-qubit gate errors are significantly lower than two-qubit gate errors. Similarly, the gate times for single-qubit gates are significantly shorter than those for two-qubit gates. We therefore assume that single-qubit gates and idling during single-qubit gates are cheap operations, while other operations are considered expensive.

In practice, this means that when comparing success probabilities, we assume $p_s \approx 1 \approx p_{is}$. Additionally, we assume that $p_d \approx p_m$, and $p_{id} \approx p_{im} \approx p_{ic}$. The assumptions are supported by observed success probabilities for quantum hardware, such as the error rates reported by IBM [IBM25].

When comparing protocols, we use approximate inequality signs ($\gtrsim$ and $\lesssim$) to indicate that we applied these assumptions on the success probabilities.

9.3 Error analysis for GHZ state preparation

In this section, we derive an expression for the success probability of preparing a GHZ state using a standard approach using an all-to-all connectivity, a standard

approach using a linear nearest-neighbor connectivity, and a LAQCC-approach. These two hardware connectivities cover most quantum hardware realizations. We also derive the success probability for a hybrid version that combines a standard approach with the LAQCC approach.

After deriving the success probabilities for the different approaches, we compare them to determine which protocol performs best in terms of the success probabilities of the individual parts of the circuit.

Informally, the LAQCC-approach performs exponentially better than the standard approach using an all-to-all connectivity if $p_d > p_{id}^{\Omega(\log n)}$. This means that the LAQCC-approach performs exponentially better if the probability of a CNOT-gate introducing an error is at most the probability that an error is introduced while a qubit is idling for $\Omega(\log n)$ CNOT-gates. For the standard approach using a linear nearest-neighbor connectivity, we find that the LAQCC-approach performs exponentially better than the standard approach if $p_d > p_{id}^{\Omega(n)}$. Both results are in line with what one might expect based on the circuit sizes of the different approaches. Theorem 9.3.1 provides a formal statement of the result for both comparisons.

9.3.1 Success probability of GHZ state preparation

Below we consider four possible approaches to prepare GHZ states and for each of them determine the success probability. The four approaches considered are:

1. Standard approach using an all-to-all connectivity;
2. Standard approach using a linear nearest-neighbor connectivity;
3. A LAQCC-approach;
4. A hybrid version of a standard and a LAQCC-approach.

We derive success probability expressions for preparing the GHZ state given by

$$\frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right).$$

Standard approach using an all-to-all connectivity

Figure 9.2 shows a quantum circuit to prepare the GHZ state for $n = 8$ using an all-to-all connectivity. In every subsequent time step, twice as many qubits can be targeted.

Let $k = \lfloor \log_2 n \rfloor$ and let $m = n - 2^k$. In the first layer, only a single one-qubit gate is applied. Afterwards, at time step $t = i$, 2^{i-1} two-qubit gates are applied, while all other qubits remain idle. After time step $t = k$, a total of $2^k - 1$ qubits

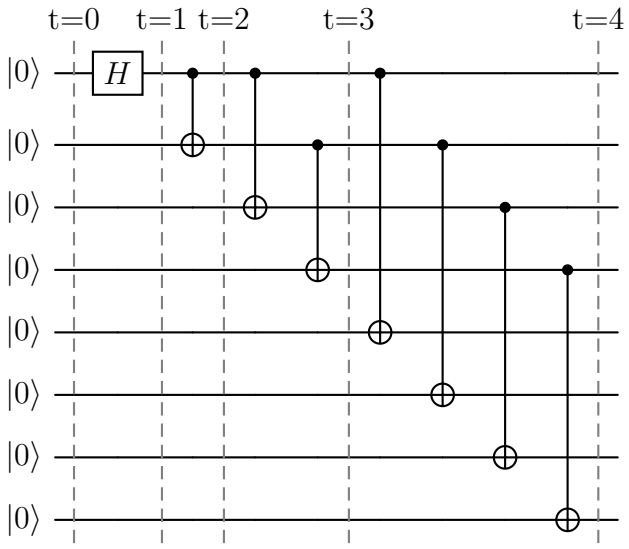


Figure 9.2: Circuit for preparing the GHZ state using an all-to-all connectivity for $n = 8$. The dotted lines indicate time steps and which gates can be applied in parallel.

have been targeted by a CNOT-gate. If n is a power of two, the state has now been prepared. Otherwise, a single extra layer is necessary with m CNOT-gates.

For the success probability $P_{GHZ_n,all}$, we then arrive at the expression:

$$\begin{aligned} P_{GHZ_n,all} &= p_s p_{is}^{n-1} \left(\Pi_{t=1}^{k} p_d^{2^{t-1}} p_{id}^{n-2^t} \right) p_d^m p_{id}^{(n-2m)(\lceil \log_2 n \rceil - k)} \\ &= p_s p_{is}^{n-1} p_d^{n-1} p_{id}^{nk-2^{k+1}+2+n(\lceil \log_2 n \rceil - \lfloor \log_2 n \rfloor)-2m(\lceil \log_2 n \rceil - \lfloor \log_2 n \rfloor)} \\ &= p_s p_{is}^{n-1} p_d^{n-1} p_{id}^{n\lceil \log_2 n \rceil - 2(n-m)+2-2m(\lceil \log_2 n \rceil - \lfloor \log_2 n \rfloor)} \\ &= p_s p_{is}^{n-1} p_d^{n-1} p_{id}^{n(\lceil \log_2 n \rceil - 2)+2}. \end{aligned} \tag{9.2}$$

As $m(\lceil \log_2 n \rceil - \lfloor \log_2 n \rfloor - 1) = 0$, most terms in the exponent of p_{id} cancel.

To determine the probability that a qubit remains coherent while a GHZ state on n qubits is prepared, we note that the circuit uses a one single-qubit gate and then $\lceil \log_2 n \rceil$ layers of CNOT-gates. Combined, we have the success probability for idling qubits of

$$p_{iGHZ_n,all} = p_{is} p_{id}^{\lceil \log_2 n \rceil}.$$

Standard approach using a linear nearest-neighbor connectivity

Figure 9.3 shows a quantum circuit to prepare the GHZ state for $n = 6$ using a linear nearest-neighbor connectivity. The key difference from the previous approach is that qubits can now only interact with their direct neighbors, and hence at most two CNOT-gates per layer can be applied.

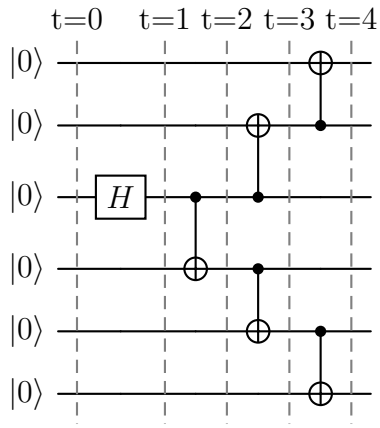


Figure 9.3: Circuit for preparing the GHZ state using a linear nearest-neighbor connectivity for $n = 6$. The dotted lines indicate time steps and which gates can be applied in parallel.

The circuit consists of the following steps: First, apply a single-qubit Hadamard gate on qubit $\left\lfloor \frac{n+1}{2} \right\rfloor$ and then a CNOT-gate from that qubit to its direct neighbor with a higher index. Let $k = \left\lfloor \frac{n}{2} \right\rfloor$ and note that we can now have $k-1$ time steps consisting of 2 CNOT-gates each. For odd n, we require a final layer consisting of a single CNOT-gate to include the last qubit in the GHZ state. We multiply the term in the exponent corresponding to the last layer with $n-2k$, as the term then vanishes for even n.

The overall probability of correctness $P_{GHZ,linear}$ is then given by

$$\begin{aligned} P_{GHZ_n,linear} &= p_s p_{is}^{n-1} p_d p_{id}^{n-2} \left(p_d^2 p_{id}^{n-4}\right)^{k-1} \left(p_d p_{id}^{n-2}\right)^{n-2k} \\ &= p_s p_{is}^{n-1} p_d^{n-1} p_{id}^{n(\lceil n/2 \rceil - 2)+2}. \end{aligned} \tag{9.3}$$

Note that for $n \leq 6$, the performance using an all-to-all connectivity and a linear nearest-neighbor connectivity is the same. For $n \geq 7$, the approach using an all-to-all connectivity has higher success probability as it has fewer idling qubits.

The idling term looks similar for this approach. Only the exponent of p_{id} differs, corresponding to the extra layers of CNOT-gates applied:

$$p_{iGHZ_n,linear} = p_{is} p_{id}^{\lceil n/2 \rceil}.$$

LAQCC-approach

Corollary 7.3.6 implies the existence of a LAQCC-circuit that prepares a GHZ state. In fact, we already saw such a circuit in Figure 4.1 in Section 4.3. Figure 9.4 shows the same circuit as in Figure 4.1 for $n = 3$ with the time steps explicitly shown.

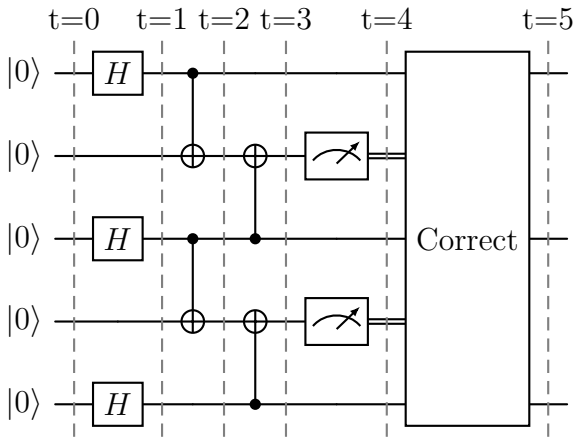


Figure 9.4: GHZ state preparation using a LAQCC circuit.

The LAQCC-circuit uses $2n-1$ qubits and the circuit depth remains constant, even with growing n. The success probability at time $t=4$ is given by

$$p_s^n p_{is}^{n-1} p_d^{2(n-1)} p_{id}^2 p_m^{n-1} p_{im}^n.$$

A prefix sum computation on the measurement results indicates which qubits need correction. In the worst case, half of the qubits require a Pauli-X correction. Assuming $p_s \leq p_{is}$, which is valid as letting a qubit remain idle is generally better than manipulating it, we can lower bound the success probability:

$$P_{GHZ_n,\mathsf{LAQCC}} \geq p_s^{n+\lfloor n/2 \rfloor} p_{is}^{n+\lceil n/2 \rceil - 1} p_d^{2(n-1)} p_{id}^2 p_m^{n-1} p_{im}^n p_{ic}^n. \tag{9.4}$$

A qubit idling while a GHZ state is prepared by a LAQCC-circuit has success probability

$$P_{iGHZ,\mathsf{LAQCC}} \geq p_{is}^2 p_{id}^2 p_{im} p_{ic}.$$

Hybrid approach

We now combine the different state preparation approaches: First, use a standard approach to prepare k small GHZ states. Then, use a LAQCC-approach to join these small GHZ states together.

We assume that k perfectly divides n, such that $n = kg$ for some positive integer g. The same analysis can also be performed when we prepare GHZ states of different sizes and then combine them. However, the analysis will be more involved.

When we use a LAQCC-protocol to join the k GHZ states, all other qubits remain idle. The total success probability is given by

$$P_{GHZ_g}^k P_{iGHZ_g}^k P_{GHZ_k,\mathsf{LAQCC}} P_{iGHZ,\mathsf{LAQCC}}^{n-k}.$$

Note that in the worst case, we still have to correct at most half of the total number of qubits, as a correction term in the LAQCC-protocol also has to be applied to all qubits in the corresponding GHZ state. Hence, the error terms in the circuit are slightly worse than what this expression alone suggests.

Depending on the connectivity used, we get one of two expressions for the success probability. For the hybrid LAQCC-all-to-all approach we obtain

$$P_{GHZ_{n,k},\textit{hybrid-all}} \geq p_s^{2k+\lfloor n/2 \rfloor} p_{is}^{3n-k+\lceil n/2 \rceil -1} p_d^{n+k-2} p_{id}^{(n+k)\lceil \log_2 g \rceil +2} p_m^{k-1} p_{im}^n p_{ic}^n. \quad (9.5)$$

For the hybrid LAQCC-linear nearest-neighbor approach we obtain

$$P_{GHZ_{n,k},\textit{hybrid-linear}} \geq p_s^{2k+\lfloor n/2 \rfloor} p_{is}^{3n-k+\lceil n/2 \rceil -1} p_d^{n+k-2} p_{id}^{(n+k)\lceil g/2 \rceil +2} p_m^{k-1} p_{im}^n p_{ic}^n. \quad (9.6)$$

9.3.2 Comparing GHZ state preparation approaches

In this section we compare the success probabilities for the different approaches to prepare the GHZ state. The main theorem proven in this section is the following.

9.3.1. THEOREM. *Let $\varepsilon > 0$ be a constant. If $p_d \gtrsim (1+\varepsilon) p_{id}^{\frac{n}{n-1}(\lceil \log_2 n \rceil /2-2)}$, then $P_{GHZ_n,\mathsf{LAQCC}} \gtrsim (1+\varepsilon)^{2(n-1)} P_{GHZ_n,all}$.*

If $p_d \gtrsim (1+\varepsilon) p_{id}^{\frac{n}{n-1}(\lceil n/2 \rceil /2-2)}$, then $P_{GHZ_n,\mathsf{LAQCC}} \gtrsim (1+\varepsilon)^{2(n-1)} P_{GHZ_n,linear}$.

In each of the following subsections, we start with a comparison of the success probabilities and then derive the expression for p_d and p_{id} accordingly. The theorem follows directly from these comparisons.

All-to-all versus LAQCC

To determine when

$$P_{GHZ_n,\mathsf{LAQCC}} \geq P_{GHZ_n,all} \quad (9.7)$$

holds, we compare Equations (9.2) and (9.4) to find that the inequality holds, precisely if

$$p_s^{n+\lfloor n/2 \rfloor -1} p_{is}^{\lceil n/2 \rceil} p_d^{n-1} p_m^{n-1} p_{im}^n p_{ic}^n \geq p_{id}^{n(\lceil \log_2 n \rceil -2)}.$$

Applying the assumptions on the success probabilities (discussed in Section 9.2), we see that this expression reduces to

$$p_d^{2(n-1)} \gtrsim p_{id}^{n(\lceil \log_2 n \rceil -4)}.$$

Hence, the LAQCC-approach performs better if

$$p_d \gtrsim p_{id}^{\frac{n}{n-1}(\lceil \log_2 n \rceil /2-2)}. \quad (9.8)$$

That is, the LAQCC-approach outperforms the standard approach using an all-to-all connectivity if the probability of a CNOT-gate introducing an error is at most the probability of a qubit idling during $\frac{n}{n-1}(\lceil \log_2 n \rceil /2 - 2)$ CNOT-gates picking up an error.

Now note that we can rewrite Equation (9.4) in terms of Equation (9.2), using the assumptions on the success probabilities, as

$$P_{GHZ_n,\mathsf{LAQCC}} \gtrsim p_d^{2(n-1)} p_{id}^{n(4-\lceil \log_2 n \rceil)} P_{GHZ_n,all}. \tag{9.9}$$

Hence, for $p_d = (1+\varepsilon) p_{id}^{\frac{n}{n-1}(\lceil \log_2 n \rceil /2-2)}$ we obtain

$$P_{GHZ_n,\mathsf{LAQCC}} \gtrsim (1+\varepsilon)^{2(n-1)} P_{GHZ_n,all}, \tag{9.10}$$

proving the first statement of Theorem 9.3.1.

Note that for large n, Equation (9.8) reduces to

$$p_d \gtrsim p_{id}^{\Omega(\log_2 n)}, \tag{9.11}$$

corresponding with what we expected based on the circuit size.

Linear versus LAQCC

Similar to the previous section, to determine when

$$P_{GHZ_n,\mathsf{LAQCC}} \geq P_{GHZ_n,linear} \tag{9.12}$$

holds, we compare Equations (9.3) and (9.4). We then see that the inequality holds precisely if

$$p_s^{n+\lfloor n/2 \rfloor -1} p_{is}^{\lceil n/2 \rceil} p_d^{n-1} p_m^{n-1} p_{im}^n p_{ic}^n \geq p_{id}^{n(\lceil n/2 \rceil -2)}.$$

Applying the assumptions on the success probabilities, we see that this expression reduces to

$$p_d^{2(n-1)} \gtrsim p_{id}^{n(\lceil n/2 \rceil -4)}.$$

Hence, the LAQCC-approach performs better if

$$p_d \gtrsim p_{id}^{\frac{n}{n-1}(\lceil n/2 \rceil /2-2)}. \tag{9.13}$$

That is, the LAQCC-approach outperforms the standard protocol using a linear nearest-neighbor connectivity if the probability of a CNOT-gate introducing an error is at most the probability of a qubit idling during $\frac{n}{n-1}(\lceil n/2 \rceil /2-2)$ CNOT-gates picking up an error.

Now note that we can rewrite Equation (9.4) in terms of Equation (9.3), using the assumptions on the success probabilities, as

$$P_{GHZ_n,\mathsf{LAQCC}} \gtrsim p_d^{2(n-1)} p_{id}^{n(4-\lceil n/2 \rceil)} P_{GHZ_n,linear}. \tag{9.14}$$

Hence, for $p_d = (1+\varepsilon) p_{id}^{\frac{n}{n-1}(\lceil n/2 \rceil/2-2)}$ we obtain

$$P_{GHZ_n,\mathsf{LAQCC}} \gtrsim (1+\varepsilon)^{2(n-1)} P_{GHZ_n,linear}, \tag{9.15}$$

proving the second statement of Theorem 9.3.1.

Note that for large n, Equation (9.13) reduces to

$$p_d \gtrsim p_{id}^{\Omega(n)}, \tag{9.16}$$

corresponding with what we expected based on the circuit size.

Comparison with the hybrid approach

We now compare the two standard approaches with their corresponding hybrid version. In the hybrid approach, k GHZ states of g qubits each are combined using the LAQCC-approach to prepare a GHZ state on $n = kg$ qubits. Therefore, we express the relative performance of the approaches in terms of k and g.

For the hybrid approach using an all-to-all connectivity, we find that it outperforms the standard approach using the same connectivity if

$$P_{GHZ_{n,k},hybrid\text{-}all} \geq P_{GHZ_n,all}.$$

Using Equations (9.2) and (9.5), this expression simplifies to

$$p_s^{2k+\lfloor n/2 \rfloor -1} p_{is}^{2n-k+\lceil n/2 \rceil} p_d^{k-1} p_m^{k-1} p_{im}^n p_{ic}^n \geq p_{id}^{n(\lceil \log_2 n \rceil - \lceil \log_2 g \rceil -2) - k\lceil \log_2 g \rceil}.$$

Again applying the assumptions on the success probabilities, we see that this expression reduces to

$$p_d \gtrsim p_{id}^{\frac{n}{2(k-1)}(\lceil \log_2 n \rceil - \lceil \log_2 g \rceil -4) - \frac{k}{2(k-1)} \lceil \log_2 g \rceil}. \tag{9.17}$$

Let $\varepsilon > 0$ and $p_d = (1+\varepsilon) p_{id}^{\frac{n}{2(k-1)}(\lceil \log_2 n \rceil - \lceil \log_2 g \rceil -4) - \frac{k}{2(k-1)} \lceil \log_2 g \rceil}$, then

$$P_{GHZ_{n,k},hybrid\text{-}all} \geq (1+\varepsilon)^{2(k-1)} P_{GHZ_n,all}. \tag{9.18}$$

Hence, the hybrid approach performs exponentially better than the standard approach.

For the hybrid approach using a linear nearest-neighbor connectivity, we find that it outperforms the standard approach using the same connectivity if

$$P_{GHZ_{n,k},hybrid\text{-}linear} \geq P_{GHZ_n,linear}.$$

Using Equations (9.3) and (9.6), this expression simplifies to

$$p_s^{2k+\lfloor n/2 \rfloor -1} p_{is}^{2n-k+\lceil n/2 \rceil} p_d^{k-1} p_m^{k-1} p_{im}^n p_{ic}^n \geq p_{id}^{n(\lceil n/2 \rceil - \lceil g/2 \rceil - 2) - k\lceil g/2 \rceil}.$$

Again applying the assumptions on the success probabilities, we see that this expression reduces to

$$p_d \gtrsim p_{id}^{\frac{n}{2(k-1)}(\lceil n/2 \rceil - \lceil g/2 \rceil - 4) - \frac{k}{2(k-1)}\lceil g/2 \rceil}. \tag{9.19}$$

Let $\varepsilon > 0$ and $p_d = (1+\varepsilon) p_{id}^{\frac{n}{2(k-1)}(\lceil n/2 \rceil - \lceil g/2 \rceil - 4) - \frac{k}{2(k-1)}\lceil g/2 \rceil}$, then

$$P_{GHZ_{n,k},\textit{hybrid-linear}} \geq (1+\varepsilon)^{2(k-1)} P_{GHZ_n,\textit{linear}}. \tag{9.20}$$

Hence, the hybrid approach performs exponentially better than the standard approach.

For both hybrid approaches, we can obtain an informal estimate similar to Equations (9.11) and (9.16). Using that $n = kg$, $\lceil x \rceil \approx x \approx \lfloor x \rfloor$ for any $x \in \mathbb{R}$, and $\frac{x}{x-1} \approx 1$ for large $x \in \mathbb{R}$, we see that Equations (9.17) and (9.19) simplify to

$$p_d \gtrsim p_{id}^{\Omega(g \log_2 k)} \tag{9.21}$$

and

$$p_d \gtrsim p_{id}^{\Omega(ng)}, \tag{9.22}$$

respectively.

Note that for $g = \mathcal{O}(1)$ (and hence $k = \Theta(n)$), the hybrid approaches perform similarly to the LAQCC-approach.

9.4 Implementation on quantum hardware

In the previous sections we compared three GHZ state generation approaches. In this section, we implement the approaches on quantum hardware and compare the results with our theoretical results. We first discuss the nuances and practicalities of implementing a protocol on currently available quantum hardware. Then, we present the results for the selected quantum device. Given the worst-case error model considered in the previous section, we expect to see differences when analyzing quantum hardware implementations.

9.4.1 Setup and implementation details

We expect that our theoretical bounds will give a lower bound on the actual success probabilities, as we consider a worst-case error model. In practice, errors may be less severe. In some cases, errors might even cancel each other out.

We consider the IBM Brisbane device of IBM, which is based on superconducting technology [IBM25]. The device has 127 qubits and is freely available via IBM's online quantum computing platform. Given the device's topology and the qubit requirements of the LAQCC-approach, we can prepare a GHZ state on at most $n = 55$ qubits.

We implement a standard approach and the LAQCC-approach. For each approach, we give the measurement outcomes as a quasi-probability distribution based on 4,096 samples. We also provide the expected success probabilities based on Equations (9.3) and (9.4) and using the parameters of the device.

Note that the measurements used to obtain this quasi-probability distribution can introduce errors themselves. Furthermore, we cannot detect phase errors, as measurements are performed in the Pauli-Z basis. However, as both approaches apply n measurements at the end of the circuit, we ignore resulting decoherence for now, expecting its impact on both outcomes to be approximately the same.

We expect differences between the hardware results and the theoretical results for multiple reasons. First, the worst-case error model used likely gives an upper bound on errors in practice. Second, quantum hardware systems typically have a limited set of native gates, requiring that some gates used in an algorithm are decomposed into native gates. Third, conventional pre- and post-processing techniques can help reduce the circuit depth and the errors in the circuits.

In a noiseless situation, we expect the circuits to give a near-uniform distribution between the all-zeros and all-ones outcomes upon measurement. In a noisy setting, we expect to often find the two correct measurement outcomes, but we also expect to find other measurement results due to implementation imperfections.

For the theoretical success probabilities, we use the success probabilities of the individual terms in a circuit. Most of these terms are provided by the quantum hardware provider in terms of error probabilities. However, the idling terms for CNOT-gates and measurements are not provided and must be derived manually.

We compute the idling success probability for CNOT-gate and for measurements using the relaxation time and dephasing time, given by decay constants T_1 and T_2, respectively. These constants define the probability that a qubit remains in its correct state. Specifically, the probability that a qubit initially in the $|1\rangle$-state remains in that state after time t is given by e^{-t/T_1}. Similarly for a qubit initially in the $|+\rangle$-state, the probability is determined by the T_2 decay constant. In general, T_1 is larger than T_2. Therefore, we will compute both p_{id} and p_{im} with the T_2 decay constant.

Each qubit has its own T_1 and T_2 constants, and similarly, every measurement and every quantum gate has its own success probability. Therefore, we take the median over all available values for every success probability term.

The IBM Brisbane device supports the CZ-gate instead of a CNOT-gate as native two-qubit gate. By conjugating the target qubit of a CNOT-gate with Hadamard gates, we obtain a CZ-gate. We therefore compute $p_d = p_{d,CZ} p_s^2 p_{is}^2$.

We now give the code used to generate the results on quantum hardware. The current quantum hardware only allows control of future quantum gates by measurement outcomes. We cannot perform computations on these outcomes, making the current LAQCC-implementation suboptimal.

```
from qiskit import QuantumCircuit, QuantumRegister,
                                    ClassicalRegister
from qiskit_ibm_runtime import QiskitRuntimeService, SamplerV2
from qiskit.transpiler.preset_passmanagers import
                                    generate_preset_pass_manager

API_token = "<your token here>"
backend_name = "ibm_brisbane"
n_qubits = 10
service = QiskitRuntimeService(channel="ibm_quantum", token=
                                    API_token)
backend = service.backend(backend_name)
pm = generate_preset_pass_manager(backend=backend,
                                    optimization_level=1)

#%% LAQCC circuit
qrm = QuantumRegister(n_qubits)
qrx = QuantumRegister(n_qubits - 1)
crx = ClassicalRegister(n_qubits-1, name="intermediate_result")
crm = ClassicalRegister(n_qubits, name="final_result")
qcircuit_LAQCC = QuantumCircuit(qrm,qrx, crx, crm, name="GHZ")
for i in range(n_qubits):
    qcircuit_LAQCC.h(qrm[i])
for i in range(n_qubits-1):
    qcircuit_LAQCC.cx(qrm[i], qrx[i])
for i in range(1,n_qubits):
    qcircuit_LAQCC.cx(qrm[i], qrx[i-1])

for i in range(n_qubits - 1):
    qcircuit_LAQCC.measure(qrx[i], crx[i])

for i in range(n_qubits-1):
    with qcircuit_LAQCC.if_test((crx[i], 1)):
        for j in range(i+1, n_qubits):
            qcircuit_LAQCC.x(qrm[j])

qcircuit_LAQCC.measure(qrm, crm)

isa_circuit_LAQCC = pm.run(qcircuit_LAQCC)
sampler = Sampler(backend)
job = sampler.run([(isa_circuit_LAQCC)])
result_LAQCC = job.result()
```

```
#%% Standard circuit
qrm = QuantumRegister(n_qubits)
crm = ClassicalRegister(n_qubits)

start_qubit = (n_qubits + 1) // 2 - 1
k = n_qubits//2
qcircuit_standard = QuantumCircuit(qrm, crm, name="GHZ")

qcircuit_standard.h(start_qubit)
qcircuit_standard.cx(start_qubit, start_qubit +1)
for i in range(k-1):
    qcircuit_standard.cx(start_qubit-i, start_qubit-i-1)
    qcircuit_standard.cx(start_qubit+1+i, start_qubit+2+i)

if n_qubits - 2*k: # Check if we need a final layer
    qcircuit_standard.cx(1, 0)

qcircuit_standard.measure(qrm, crm)

isa_circuit_standard = pm.run(qcircuit_standard)
sampler = Sampler(backend)
job = sampler.run([(isa_circuit_standard)])
result_standard = job.result()
```

We now proceed by comparing the theoretical success probabilities from Equations (9.3) and (9.4) with the implementation results. As mentioned, we expect these success probabilities to give a lower bound on the actual success probabilities. Additionally, we give the expected running time of each circuit based on the obtained gate and measurement times. Finally, we show the measurement results for different values of n for both approaches, allowing us to compare practical performance and observe how the success probabilities change as n grows.

9.4.2 IBM Brisbane device

This section presents the results of the hardware experiments run on the IBM Brisbane device. The IBM Brisbane device has 127 qubits, allowing for the preparation of a GHZ state on at most 55 qubits. Table 9.2 summarizes the success probabilities of different gates of this device, using a T_2 value of 131.71μs.

Equations (9.3) and (9.4) give the following two success probabilities for the largest GHZ state preparable on this device

$$\begin{aligned} P_{Brisbane,GHZ_{55},standard} &= p_s p_{is}^{54} p_d^{54} p_{id}^{1432} \\ &= 4.52 \cdot 10^{-4} \\ P_{Brisbane,GHZ_{55},\mathsf{LAQCC}} &\geq p_s^{82} p_{is}^{83} p_d^{108} p_{id}^{2} p_m^{54} p_{im}^{55} p_{ic}^{55} \\ &= 4.82 \cdot 10^{-2}. \end{aligned}$$

Success term	Value	Obtained via
p_s	$1 - 2.530 \cdot 10^{-4}$	Provided by IBM
p_{is}	$1 - 2.530 \cdot 10^{-4}$	Provided by IBM
p_d	$1 - 9.442 \cdot 10^{-3}$	Provided by IBM and computed with p_s
p_{id}	$1 - 4.998 \cdot 10^{-3}$	Based on a gate time of 660 ns
p_m	$1 - 1.600 \cdot 10^{-2}$	Provided by IBM
p_{im}	$1 - 9.822 \cdot 10^{-3}$	Based on a measurement time of 1300 ns
p_{ic}	$1 - 9.822 \cdot 10^{-3}$	Equal to p_{im}

Table 9.2: Success probabilities of IBM Brisbane device, based on calibration on December 13, 2024 at 09.30 (UTC+2).

We see a factor 100 difference in the theoretical success probabilities with a worst-case error model, favoring the LAQCC-approach.

To determine the time duration of both approaches, we estimate the single-qubit gate time using p_s and T_2 to be approximately 33 ns. Using a two-qubit gate time of 660 ns and a measurement time of 1300 ns, we find running times

$$T_{Brisbane,GHZ_{55},standard} = 33\text{ns} + 28 * 660\text{ns} = 18.51\mu\text{s}$$
$$T_{Brisbane,GHZ_{55},\text{LAQCC}} = 2(33 + 660 + 1300)\text{ns} = 3.99\mu\text{s}.$$

Hence, we expect the LAQCC-approach to have a shorter running time.

Below, we give the results of the hardware implementations for different n. For small n, all measurement outcomes are given. For larger n, we instead give aggregated results, grouping the results of strings with the same Hamming weight.

Based on the success probabilities shown in Table 9.2 and the theoretical relation between p_d and p_{id} shown in Theorem 9.3.1, we expect the LAQCC-approach to outperform the standard approach for $n \geq 15$.

Figure 9.5 shows the results for small n. The standard approach slightly outperforms the LAQCC-approach, as expected based on the success probabilities.

Figure 9.6 shows the aggregated results for $n = 20$ and $n = 25$, where measurement outcomes with the same Hamming weight are grouped. The aggregated results show the magnitude of the errors. With the LAQCC-approach, both expected outcomes are measured, but the results are somewhat uniform. In contrast, the standard approach did not return the all-ones string in any measurement, but does show two distinct peaks near the low and high Hamming weight outcomes.

Figure 9.7 shows the results for large n. The LAQCC-approach appears to produce a normal distribution, as seen for $n = 25$. This suggests the LAQCC-approach samples from a uniform superposition over all bit strings. The results for the standard approach show similarities with the expected distribution, with a little more weight towards the two extremes of the distribution.

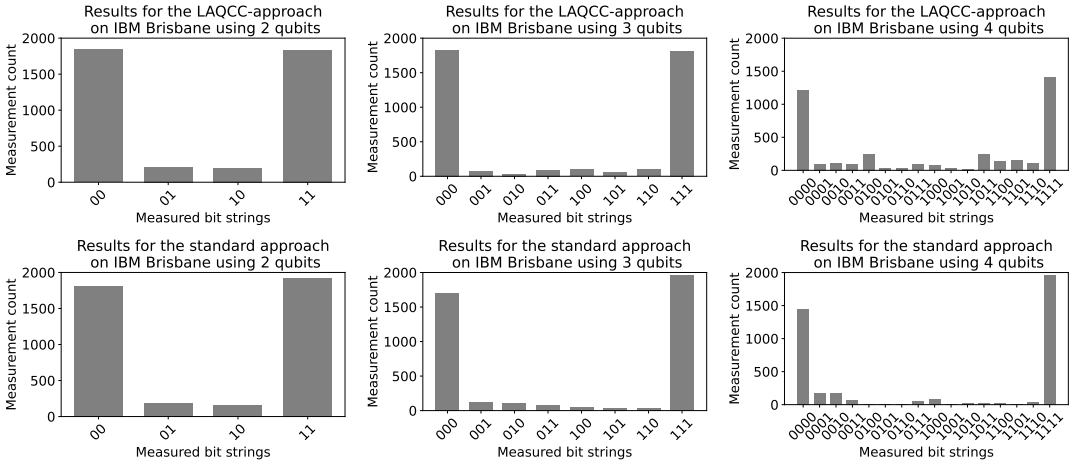


Figure 9.5: Measurement results for preparing a GHZ state on few qubits on the IBM Brisbane device using the LAQCC-approach and the standard approach. Horizontally, the different measurement results are shown and the height of the bars shows how often that measurement result is found.

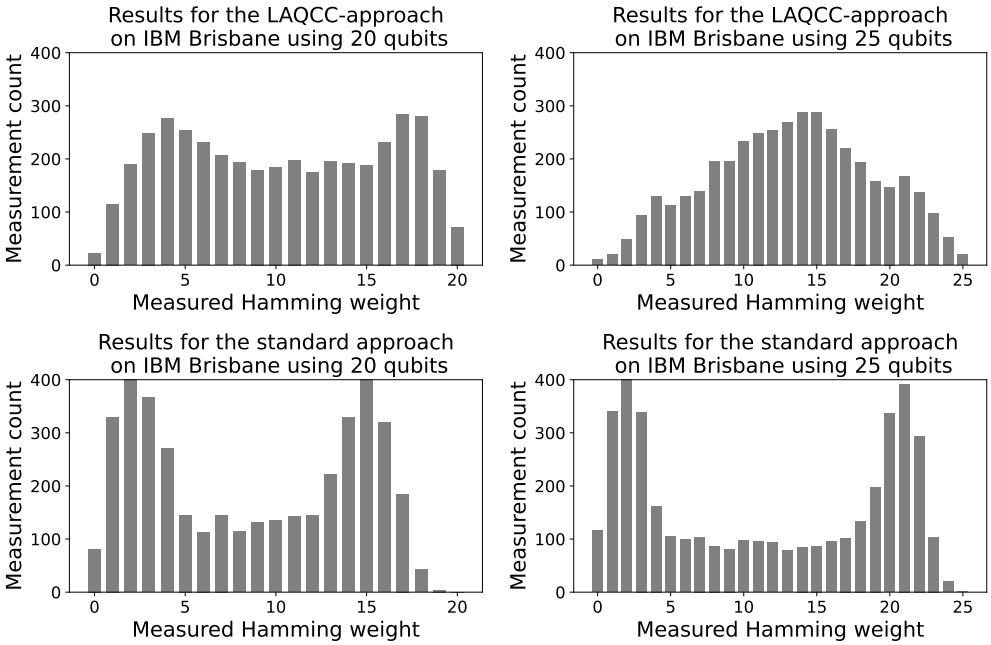


Figure 9.6: Measurements results for preparing a GHZ state on $n = 20$ and $n = 25$ qubits on the IBM Brisbane device using the LAQCC-approach and the standard approach. Horizontally, the different measured Hamming weights are shown and the height of the bars shows how often that Hamming weight was found.

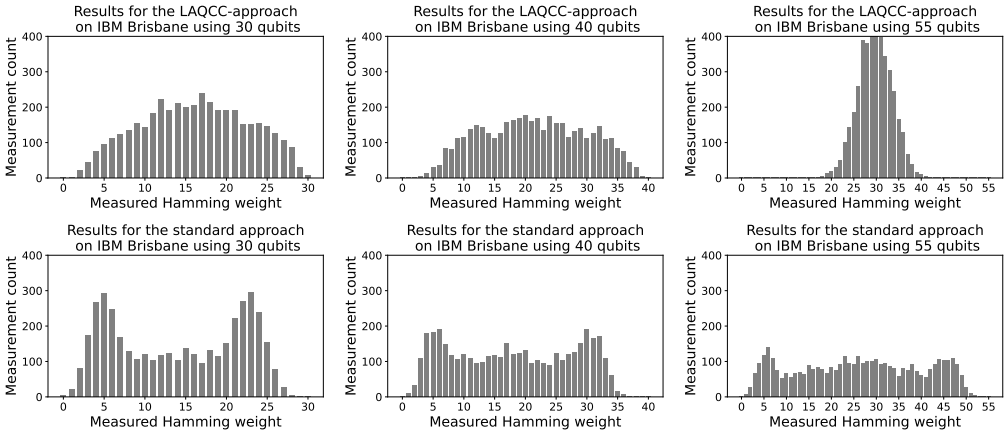


Figure 9.7: Measurement results for preparing a GHZ state on $n = 30$, $n = 40$, and $n = 55$ qubits on the IBM Brisbane device using the LAQCC-approach and the standard approach. Horizontally, the different measured Hamming weights are shown and the height of the bars shows how often that Hamming weight was found.

9.5 Error analysis for W-state preparation

In this section, we derive an expression for the success probability of preparing W-states with a standard approach using a linear nearest-neighbor connectivity and with the LAQCC-approach from Section 8.3. The standard approach has depth $\mathcal{O}(n)$ and uses n qubits, whereas the LAQCC-approach uses $\mathcal{O}(n \log(n) \log\log(n))$ qubits and has constant depth. Based on the circuit sizes, our intuition suggests that the LAQCC-approach performs better if

$$p_d \gtrsim p_{id}^{\Omega(n/(\log(n)\log\log(n)))}.$$

In the next sections we will derive success probabilities for both approaches and will see that our intuition is indeed correct, as summarized in the next theorem.

9.5.1. Theorem. *Let $n = 2^k$ for some integer k and let $\varepsilon > 0$ be a constant. If $p_d = (1+\varepsilon)p_{id}^{3n/(59\log_2(n)\log_2\log_2(n))}$, then, with respect to the most significant terms, $P_{W,\mathsf{LAQCC}} \gtrsim (1+\varepsilon)^{59n/(\log_2(n)\log_2\log_2(n))} P_{W,direct}$.*

Section 9.5.1 presents success probabilities for some LAQCC-subroutines. Sections 9.5.2 and 9.5.3 give the success probabilities for preparing the W-state using a LAQCC-approach and using a standard approach. Finally, Section 9.5.4 compares the two approaches and proves the theorem.

9.5.1 Success probability for different subroutines

The LAQCC-approach to prepare the W-state uses multiple subroutines, such as the fanout gate and the OR-gate. This section provides the success probability

for these subroutines. Specifically, this section gives the success probability for the fanout gate and parity gate on n qubits, the OR-reduction introduced by Høyer and Špalek [HŠ05] and the exact OR-gate by Takahashi and Tani [TT13].

Fanout and parity gate

A fanout gate on n qubits has one control qubit and $n-1$ target qubits. Our implementation requires $3n-1$ qubits and is inspired by the non-local CNOT-gate by Yimsiriwattana and Lomonaco [YL04]. The circuit first prepares a GHZ state on n qubits using $2n-1$ qubits and then uses this GHZ state to apply parallel gate teleportation to implement the fanout gate. Figure 9.8 gives the corresponding circuit for $n = 3$, the time steps indicate which gates can be applied in parallel.

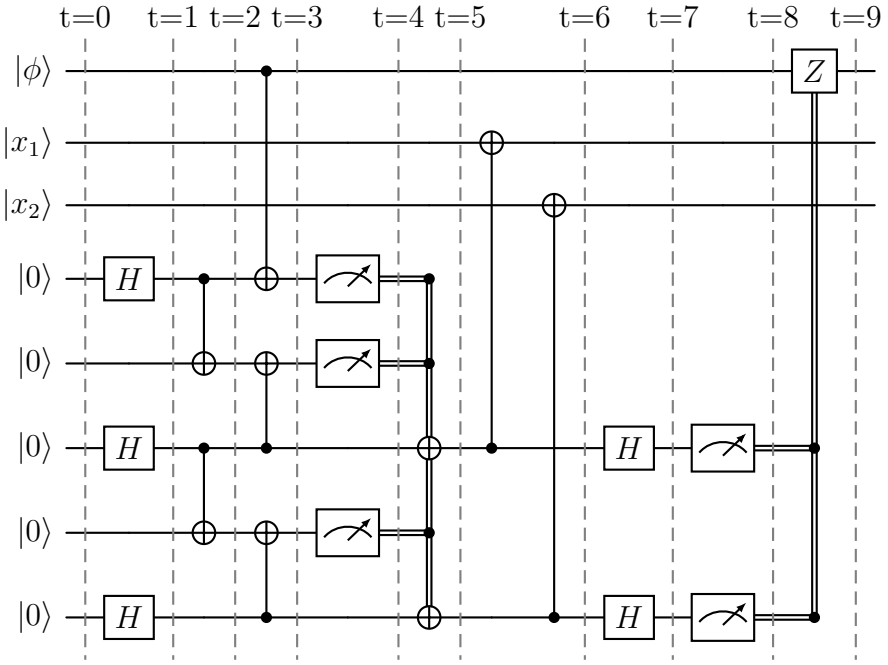


Figure 9.8: Implementation of a quantum fanout gate with the GHZ state preparation expanded. The time steps indicate which gates can be applied in parallel.

We can extend the circuit shown in Figure 9.8 to general n and count the gates and idling terms to obtain the following expression for the success probability:

$$P_{Fanout_n} \geq p_s^{2n+\lceil (n-1)/2 \rceil} p_{is}^{5n+\lfloor (n-1)/2 \rfloor - 2} p_d^{3n-2} p_{id}^{2n+1} p_m^{2n-1} p_{im}^{3n-1} p_{ic}^{3n-1}. \tag{9.23}$$

The fanout gate and the parity gate are closely related, as the parity gate corresponds to a fanout gate with each qubit conjugated by Hadamard gates. By including the Hadamard gates on the first n qubits in the fanout-gate circuit, we

reduce the number of idling qubits, while retaining the same circuit depth. This gives a success probability for the parity gate on n qubits of

$$P_{Parity_n} \geq p_s^{4n+\lceil (n-1)/2 \rceil -1} p_{is}^{3n+\lfloor (n-1)/2 \rfloor -1} p_d^{3n-2} p_{id}^{2n+1} p_m^{2n-1} p_{im}^{3n-1} p_{ic}^{3n-1}. \tag{9.24}$$

As the circuits for both the fanout gate and the parity gate have the same depth and apply the same type of operations, we see that a qubit idling during the execution of either of the two gates has the same success probability. Counting the gates gives a success probability for an idling qubit of

$$P_{iFanout_n} = P_{iParity_n} = p_{is}^4 p_{id}^3 p_{im}^2 p_{ic}^2. \tag{9.25}$$

OR-reduction

The OR-reduction introduced by Høyer and Špalek prepares a state on $\mathcal{O}(\log n)$ qubits, such that evaluating an OR-gate on this reduced state gives the same output as the OR-gate evaluated on the initial n qubits [HŠ05, Lemma 5.1].

Let c be any positive integer and $\varphi \in [0, 2\pi)$, define the state

$$\left|\mu_\varphi^c\right\rangle = \frac{1 + e^{i\varphi c}}{2} |0\rangle + \frac{1 - e^{i\varphi c}}{2} |1\rangle .$$

We obtain this state by computing: $\left|\mu_\varphi^c\right\rangle = HR_Z(\varphi c)H |0\rangle$. The OR-reduction uses these $\left|\mu_\varphi^c\right\rangle$ states.

Given input $x_1, \ldots, x_n$, the OR-reduction prepares $t = \lceil \log_2(n+1) \rceil$ states $\left|\mu_{\varphi_k}^{|x|}\right\rangle$ for $\varphi_k = \frac{2\pi}{2^k}$ and $k \in [t]$. We can prepare the states $\left|\mu_{\varphi_k}^{|x|}\right\rangle$ for each k in parallel using fanout gates. Figure 9.9 shows the circuit for a single k, with the time steps again indicating which gates can be applied in parallel.

We can parallelize this circuit for every k using fanout gates. The n fanout gates of length t for copying the input qubits can be applied in parallel with the t fanout gates of length n for copying the auxiliary qubits. As a result, we have no idling terms for the fanout gates. Additionally, the first and last Hadamard gate in Figure 9.9 can be included in the construction of the fanout gates. This gives a total success probability of:

$$\begin{aligned} P_{OR_n\text{-reduction}} &= P_{Fanout_t}^{2n} P_{Fanout_n}^{2t} \left(P_{cR_Z}^n\right)^t \\ &\geq p_s^{11nt+2(n\lceil (t-1)/2 \rceil + t\lceil (n-1)/2 \rceil)+2t} p_{is}^{23nt+2n\lfloor (t-1)/2 \rfloor + 2t\lfloor (n-1)/2 \rfloor - 4(n+t)} \\ &\quad p_d^{14nt-4(n+t)} p_{id}^{8nt+2(n+t)} p_m^{8nt-2(n+t)} p_{im}^{12nt-2(n+t)} p_{ic}^{12nt-2(n+t)} \end{aligned} \tag{9.26}$$

We now use this expression to determine the success probability for the OR-gate.

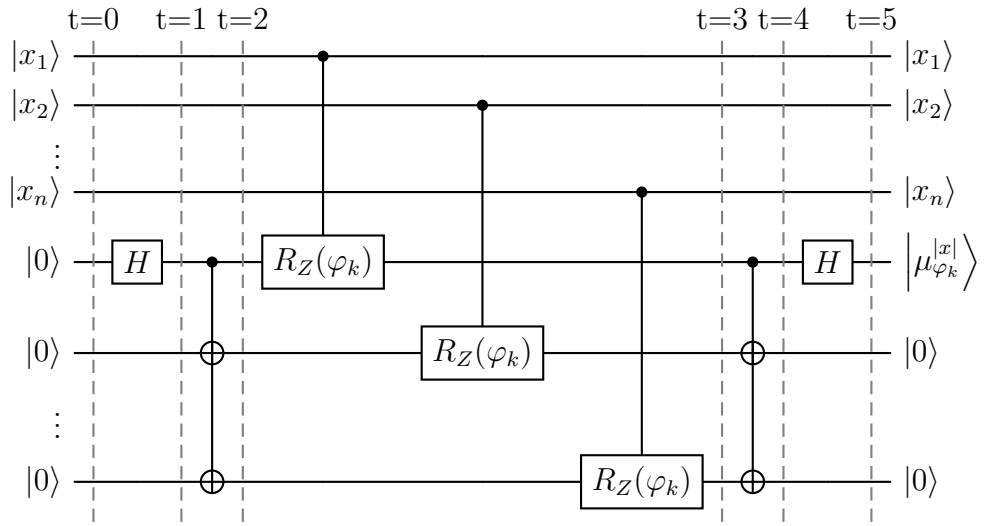


Figure 9.9: LAQCC-circuit that prepares the state $\left|\mu_{\varphi_k}^{|x|}\right\rangle$. These states are used to implement the OR-reduction. The dotted lines indicate time steps and which gates can be applied in parallel.

OR-gate

Takahashi and Tani presented an exponential-sized circuit for the OR-gate, which they applied to a state of logarithmic size [TT13]. Combined, this gives a circuit of polynomial size. Using the Fourier inversion formula shown in Equation (1.11) we see that for a bit string $x \in \mathbb{F}_2^n$ we can write

$$\mathrm{OR}_n(x) = \frac{1}{2^{n-1}} \sum_{a \in \mathbb{F}_2^n \setminus \{0^n\}} \mathrm{PA}_n^a(x),$$

where $\mathrm{PA}_n^a(x) = \oplus_{j=0}^{n-1} a_i x_i$ is the parity of x, weighted by a nonzero vector a. This weighted-parity gate is implemented by a standard parity gate on a subset of the inputs. The exponential-sized circuit for the OR_n-gate consists of three steps:

1. Simultaneously,
   (a) Copy the input state $2^n - 1$ times and compute $\mathrm{PA}_n^a(x)$ for every nonzero n-bit string a;
   (b) Prepare a GHZ state on $2^n - 1$ qubits;
2. Apply $R_Z(\pi/2^{n-1})$-gates to the qubits in the GHZ state and controlled by the qubits that hold the $\mathrm{PA}_n^a(x)$ results;
3. Apply a fanout gate of length $2^n - 1$ to the GHZ state together with a Hadamard gate to uncopy the GHZ state and obtain the OR of the input in a single auxiliary qubit.

We then need to uncompute the auxiliary registers by running the first two steps. This introduces an extra factor two in the exponents in the overall success probability. Furthermore, in this protocol, we can choose to prepare the GHZ states only when they are needed, thereby omitting idling terms for the GHZ states.

In total, we will have n parity gates with a single target, $\binom{n}{2}$ parity gates with two targets, and in general $\binom{n}{k}$ parity gates with k targets, where k corresponds to the Hamming weight of a. A lower bound the success probability follows by replacing every parity gate by the success probability for the parity gate on all inputs, resulting in an easier expression.

Let n be the length of the input and $t = \lceil \log_2(n+1) \rceil$. The success probability for the OR-gate is then given by

$$\begin{aligned} P_{OR_n} &= P^2_{OR_n\text{-}reduction} P^{2t}_{Fanout_{2^t-1}} \left(\Pi_{k=1}^{t} \binom{t}{k} P_{Parity_k} \right)^2 P^2_{GHZ_{2^t-1},\mathsf{LAQCC}} P^{2^t-1}_{cR_Z} p_s^1 \\ &\geq P^2_{OR_n\text{-}reduction} P^{2t}_{Fanout_{2^t-1}} P^{2(2^t-1)}_{Parity_t} P^2_{GHZ_{2^t-1},\mathsf{LAQCC}} P^{2^t-1}_{cR_Z} p_s^1 \\ &\geq p_s^{22nt+2(2n-2^t-1)\lceil (t-1)/2 \rceil + 4t\lceil (n-1)/2 \rceil + 2t\left\lceil (2^{t-1}-1)/2 \right\rceil + 2\left\lceil (2^t-1)/2 \right\rceil + 10t\cdot 2^t + 3\cdot 2^t - 4t - 2} \\ &\quad p_{is}^{2(2n+2^t-1)\lfloor (t-1)/2 \rfloor + 4t\lfloor (n-1)/2 \rfloor + 2t\left\lfloor (2^{t-1}-1)/2 \right\rfloor + 2\left\lfloor (2^t-1)/2 \right\rfloor} \\ &\quad p_{is}^{46nt-8n-18t+11t\cdot 2^t+3\cdot 2^t-5} p_d^{28nt-8n-18t+9t\cdot 2^t+2\cdot 2^t-6} p_{id}^{16nt+4n+2t+6t\cdot 2^t+2\cdot 2^t+2} \\ &\quad p_m^{16nt-4n-10t+6t\cdot 2^t-2} p_{im}^{24nt-4n-12t+9t\cdot 2^t} p_{ic}^{24nt-4n-12t+9t\cdot 2^t}. \end{aligned} \tag{9.27}$$

Note that this expression holds for arbitrary n. For $n = 2^k$, the expression simplifies, as in that case $t = \lceil \log_2(n+1) \rceil = k+1$, which gives

$$\begin{aligned} P_{OR_n} \geq\ & p_s^{42nk+50n-4k+2(2k-2n+1)\lceil k/2 \rceil + 6(k+1)\lceil (n-1)/2 \rceil - 6} \\ & p_{is}^{68nk+68n-18k+2(4n-1)\lfloor k/2 \rfloor + 6(k+1)\lfloor (n-1)/2 \rfloor - 25} p_d^{46nk+42n-18k-24} \\ & p_{id}^{28nk+36n+2k+4} p_m^{28nk+24n-10k-12} p_{im}^{42nk+38n-12k-12} p_{ic}^{42nk+38n-12k-12}. \end{aligned} \tag{9.28}$$

9.5.2 LAQCC-approach

This section gives the success probability for preparing the W-state on $n = 2^k$ qubits, for some integer k, using the LAQCC-approach presented in Section 8.3. We obtain the success probability for the W-state by determining them for the **Uncompress** and **Compress** methods and then taking the product of the two. Following the steps outlined in Lemmas 8.3.2 and 8.3.3 gives:

$$P_{\mathbf{Uncompress}_n} = p_s^k p_{is}^{nk+n-k} P^{2k}_{Fanout_n} P^{2n}_{iFanout} \left(\Pi_{i=0}^{n-1} P_{Equal_i} \right). \tag{9.29}$$

$$P_{\mathbf{Compress}_n} = p_s^{2k} p_{is}^{2(nk+n-k)} P^{2k}_{Fanout_n} P^{2n}_{iFanout} \left(\Pi_{i=0}^{n-1} P_{cZ,target_i} \right). \tag{9.30}$$

The success probability for the Equal_i-gate is lower bounded by the success probability of the OR_k-gate with all k input qubits conjugated with X-gates. We

can incorporate these X-gates in the circuit for the OR_k-gate. The controlled-Z-gates with target i correspond to a fanout gate with targets on the qubits corresponding to the ones in the binary representation of i, and with the targets conjugated by Hadamard gates. Hence, we can lower bound the success probability of these controlled-Z-gates with target i by the success probability of a fanout gate of length $k+1$ with the target qubits conjugated by Hadamard gates. Summarizing, we have that for every $i \in \mathbb{F}_2^k$

$$P_{Equal_i} \geq P_{OR_k} p_s^{2k} p_{is}^2, \tag{9.31}$$

$$P_{cZ,target_i} \geq P_{Fanout_{k+1}} p_s^{2k} p_{is}^2. \tag{9.32}$$

We can now obtain a lower bound on the success probability of preparing the W-state by multiplying the expressions of Equations (9.29) and (9.30), applying the lower bounds described in Equations (9.31) and (9.32) and using the expression for the OR-gate given in Equation (9.27). We use $t = \lceil \log_2(k+1) \rceil$ and obtain

$$\begin{aligned}
P_{W,\mathsf{LAQCC}} &= P_{\mathbf{Uncompress}_n} P_{\mathbf{Compress}_n} \\
&\geq p_s^{4nk+3k} p_{is}^{3nk+7n-3k} P_{Fanout_n}^{4k} P_{Fanout_{k+1}}^{n} P_{iFanout}^{4n} P_{OR_k}^{n} \\
&\geq p_s^{22nkt+14nk+2n\left\lceil (2^t-1)/2 \right\rceil + n(3\cdot 2^t + \lceil k/2 \rceil) + 2nt(5\cdot 2^t-2)+3k+4k\lceil (n-1)/2 \rceil + 2n(2k-2^t-1)\lceil (t-1)/2 \rceil} \\
&\quad p_s^{4nt\lceil (k-1)/2 \rceil + 2nt\left\lceil (2^{t-1}-1)/2 \right\rceil + 2n\left\lceil (2^t-1)/2 \right\rceil} p_{is}^{46nkt+20nk+3n\cdot 2^t-18nt+21n-11k+n\lfloor k/2 \rfloor} \\
&\quad p_{is}^{2n(2k+2^t-1)\lfloor (t-1)/2 \rfloor + 4nt\lfloor (k-1)/2 \rfloor + 2nt\left\lfloor (2^{t-1}-1)/2 \right\rfloor + 2n\left\lfloor (2^t-1)/2 \right\rfloor + 11nt\cdot 2^t + 4k\lfloor (n-1)/2 \rfloor} \\
&\quad p_d^{28nkt+7nk+9nt(2^t-2)+2n\cdot 2^t-5n-8k} p_{id}^{16nkt+14nk+2nt(3\cdot 2^t+1)+2n\cdot 2^t+17n+4k} \\
&\quad p_m^{16nkt+6nk+2nt(3\cdot 2^t-5)-n-4k} (p_{im}p_{ic})^{24nkt+11nk+3nt(3\cdot 2^t-4)+10n-4k}
\end{aligned} \tag{9.33}$$

Note that this expression is quite involved with dependencies on both n, $k = \log_2(n)$ and $t = \lceil \log_2(k+1) \rceil$. We furthermore use ceil- and floor-functions. We can approximate Equation (9.33) using $\lceil x \rceil \approx x \approx \lfloor x \rfloor$ for $x \in \mathbb{R}$. In this case, we see that $2^t \approx k$. We then obtain

$$\begin{aligned}
P_{W,\mathsf{LAQCC}} \gtrsim\ & p_s^{71nkt/2+37nk/2+3nk-8nt-n+k} p_{is}^{125nkt/2+47nk/2-22nt+21n-13k} \\
& p_d^{37nkt+9nk-18nt-5n-8k} p_{id}^{22nkt+16nk+2nt+17n+4k} \\
& p_m^{22nkt+6nk-10nt-n-4k} (p_{im}p_{ic})^{33nkt+11nk-12nt+10n-4k}
\end{aligned} \tag{9.34}$$

9.5.3 Direct method

The most direct method of preparing a W-state is by applying successive controlled rotations, using controlled-R_Y-gates (Equation (1.6)), where the angle of the gates depends on the qubit index. Figure 9.10 shows the circuit for $n = 4$. Each square $1/n$ denotes an $R_Y(\theta)$-gate with argument $\theta = -2 \arccos \sqrt{1/n}$. The controlled-$R_Y$-gate reduce to a CNOT-gate for $n = 2$.

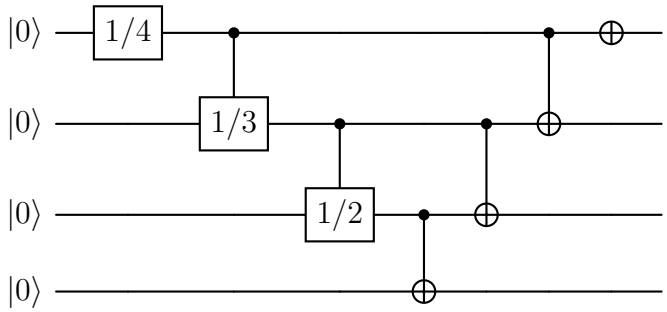


Figure 9.10: Exact circuit for preparing the W-state for $n = 4$. Every gate parametrized by $1/n$ denotes a controlled-R_Y-gate with argument $\theta = -2 \arccos \sqrt{1/n}$.

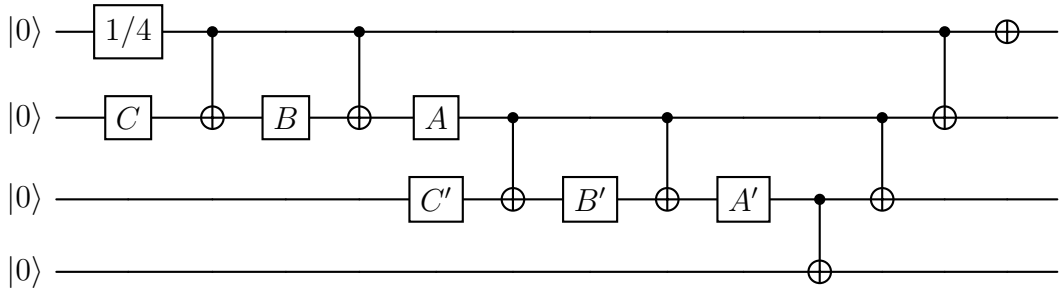


Figure 9.11: Exact decomposed circuit for preparing the W-state for $n = 4$, where every controlled R_Y-gate is replaced by single-qubit gates and CNOT-gates.

The quantum circuit shown in Figure 9.10, when extended to arbitrary n, indeed prepares the W-state on n qubits. Just before the first CNOT-gate, the quantum circuit corresponds to the state

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |1\rangle^i |0\rangle^{n-i} .$$

Each CNOT-gate will then correctly set one additional qubit, until we have the desired W-state. The idea behind this circuit is to iteratively "pass on" part of the amplitude to the remaining unset qubits and thereby correctly set all qubits.

The circuit shown in Figure 9.10 uses controlled-R_Y-gates, which most quantum hardware devices do not directly support. We can decompose the controlled-R_Y-gates using the decomposition shown in Figure 9.1. In our case, the exact single-qubit gates used in the decomposition are irrelevant, as we assume the same success probability for every single-qubit gate. Figure 9.11 shows the resulting decomposed quantum circuit to prepare the W-state for $n = 4$. This figure shows that some gates can be applied in parallel. Counting the depth gives $n-2$ groups

of four layers each, with two single-qubit gates, a CNOT-gate, one single-qubit gate and again a CNOT-gate. Next, there is one single-qubit gate, $n-1$ layers of CNOT-gates and a final single-qubit gate, for a total depth of $5n-7$ for $n \geq 2$. The total success probability is given by

$$P_{W,direct} = p_s^{3n-4} p_{is}^{n(2n-5)+4} p_d^{3n-5} p_{id}^{n(3n-11)+10}. \tag{9.35}$$

9.5.4 Comparison success probability

To determine when the LAQCC-approach outperforms the standard approach, we have to determine when

$$P_{W,\mathsf{LAQCC}} \geq P_{W,direct}. \tag{9.36}$$

Comparing the success probabilities given in Equations (9.33) and (9.35) shows that this inequality holds approximately if the following inequality holds

$$\begin{aligned} &p_s^{71nkt/2+37nk/2+3nk-8nt-4n+k+4} p_d^{37nkt+9nk-18nt-8n-8k+5} \\ &p_m^{22nkt+6nk-10nt-n-4k} (p_{im}p_{ic})^{33nkt+11nk-12nt+10n-4k} \\ &\quad \geq p_{is}^{2n^2-125nkt/2-47nk/2+22nt-26n+13k+4} p_{id}^{3n^2-22nkt-16nk-2nt-28n-4k+10}. \end{aligned}$$

Applying the assumptions on the success probabilities discussed in Section 9.2, shows that this inequality reduces to

$$p_d^{59nkt+15nk-28nt-9n-12k+5} \gtrsim p_{id}^{3n^2-88nkt-38nk+22nt-48n+4k+10}.$$

For a first estimate on when the LAQCC-approach outperforms the standard direct approach, we only keep the most significant terms. Using that $k = \log_2 n$ and $t \approx \log_2 \log_2 n$ shows that the LAQCC-approach performs best if

$$p_d \gtrsim p_{id}^{3n/(59 \log_2 n \log_2 \log_2 n)}.$$

Let $\varepsilon > 0$, now if $p_d = (1+\varepsilon) p_{id}^{3n/(59 \log_2 n \log_2 \log_2 n)}$, then we see that $P_{W,\mathsf{LAQCC}} \gtrsim (1+\varepsilon)^{59n \log_2 n \log_2 \log_2 n} P_{W,direct}$, finishing the proof of Theorem 9.5.1 and also showing our intuition was indeed correct.

9.6 Reflections and outlook

In this chapter, we considered the success probability of different approaches to prepare the GHZ state and the W-state. We used a worst-case error model, where errors correspond to Haar random unitary matrices being applied to the gates. A single error causes the task of perfectly preparing the quantum states to fail, as multiple errors cancel with probability zero. As such, we derived expressions for the probability that none of the qubits decoheres, either while idling or while being manipulated by a quantum gate.

For the GHZ state, we derived success probabilities for two standard protocols, using either an all-to-all connectivity or a linear nearest-neighbor connectivity, as well as for a LAQCC-approach. We also did this for a hybrid version of a standard approach and a LAQCC-approach. Next, we compared the derived success probabilities to determine which approach performs best when. Conditional on assumptions on the magnitude of the success probabilities of the individual terms in the quantum circuit, we find that the LAQCC-approach performs exponentially better than the standard approach using an all-to-all connectivity if $p_d \gtrsim (1+\varepsilon)p_{id}^{\Omega(\log n)}$, and similarly for the linear nearest-neighbor connectivity if $p_d \gtrsim (1+\varepsilon)p_{id}^{\Omega(n)}$. Theorem 9.3.1 gives the exact constants. The theorem says that the LAQCC-approach outperforms the standard approaches if the probability that a qubit decoheres while idling for $\Omega(\log n)$, respectively, $\Omega(n)$ two-qubit gate durations is larger than the probability that a single two-qubit gate is erroneous. Both results are in line with what our intuition tells us based on the circuit sizes.

For the GHZ state we implemented both the LAQCC-approach and the standard approach using a linear nearest-neighbor connectivity. We found that for small problem instances, both approaches perform similarly. For larger n, the LAQCC-approach returns an approximately normal distribution (when aggregating the results based on the Hamming weight). The standard approach gives results that show some similarities with the expected output distribution: two large groups of outputs having low or high Hamming weight and some samples in between. Based on the results, we saw that the used error model was suboptimal and differs from practical error models. Note that we could only use the intermediate measurement outcomes to control future quantum operations, and not to perform computations with them before controlling quantum operations.

Next, we performed a similar analysis for the W-state. The LAQCC-approach has constant depth and circuit size $\mathcal{O}(n \log n \log\log n)$. The standard direct approach uses n qubits and has depth $\mathcal{O}(n)$. Comparing the derived success probabilities shows that the LAQCC-approach exponentially outperforms the standard approach if $p_d \gtrsim (1+\varepsilon)p_{id}^{\Omega(n/(\log(n)\log\log(n)))}$. Theorem 9.5.1 gives the constants for the most significant term in the exponent.

Multiple directions for future research exist. First, the same analysis can be used to compare different quantum approaches to prepare other quantum states. We expect similar relations between p_d and p_{id}, depending on the circuit sizes and the number of CNOT-gates. Ideally, one would prove such a relation on a higher level, obtaining a result for multiple quantum state preparation routines.

Second, we used a worst-case error model in our analysis. The hardware implementation demonstrated that the error model is suboptimal and differs from practical error models. Specifically, the used error model does not allow for errors, whereas in practice, some errors can be tolerated or might even cancel.

Furthermore, conventional post-processing techniques exist to mitigate the effect of errors [TBG17; Kan+19; Cai+23].

A similar analysis with a more realistic error model is an interesting extension of this chapter. The depolarizing and dephasing channels applied to the quantum gates are an example of a more realistic error model. With these error models, gates are replaced by standard probability distributions over different gates, and two errors might cancel. These error models give a more realistic picture of the performance of a quantum circuit, but complicate the analysis. We then need both upper and lower bounds on the success probabilities to compare them.

Another aspect in which we can extend the error model is by dropping the constraint of independent errors. Kalai argues that the noise models underlying quantum computing are complex and highly entangled [Kal16]. He states that error models with independent errors are oversimplified and cannot approximate the behavior for large systems well. Correlated noise is also seen with metronomes initialized randomly but placed on the same surface. Over time, these metronomes will synchronize [PRK01], something that cannot be explained by (noise) models of individual metronomes alone. The computation of the success probability will become extremely complex with error models where errors depend on the state of other qubits or on operations applied to other qubits.

A third direction for future research is to make the circuits themselves more realistic. In general, quantum gates have to be decomposed into the native gate set of a device, giving an overhead currently not addressed in the success probability expressions. Additionally, some quantum hardware devices limit on the number of parallel gates, for instance due to crosstalk between qubits. This crosstalk is magnified when two qubits close to each other are simultaneously manipulated by two different quantum gates [Zha+22]. Such limits on the number of parallel gates significantly reduces the power of the quantum device. For instance, the advantage of the LAQCC-approach significantly diminishes if only a constant number of gates can be applied in parallel. This limitation might be one of the reasons why the LAQCC-approach performs worse in practice than expected.

Fourth, determine the output states in a different way. We measured the states in the Pauli-Z basis, which cannot detect phase errors. By instead applying state tomography, for instance, by measuring the expectation value of different Pauli observables, we can also detect phase errors. The overhead of state tomography is however large and proves infeasible for large system sizes.

In line with this extension, we arrive at the fifth possible direction for future research: relax the requirements on the output of the circuits. Currently, the output state must match the target state exactly; otherwise, the circuit failed. In practice, a high overlap with the target state suffices. This relaxation directly opens the way to more possible quantum circuits.

Approximating the target state opens the way to probabilistic algorithms. For instance, approximately preparing the W-state is possible by applying n single-qubit R_Y-gates with parameter $\theta = \arccos\left(\sqrt{\frac{n-1}{n}}\right)$ and then measuring the parity of these gates. Upon measuring an odd parity, the superposition collapses to a superposition over all bit strings of odd Hamming weight, with those having Hamming weight 1 having the highest amplitudes. Choosing a smaller θ reduce the probability of finding an odd parity. However, once an odd parity is measured, the resulting state better approximates the W-state.

Allowing the output state to have high overlap with the target state, gives room to simplify the assumptions on the gate set. The current analysis assumes that all single-qubit gates are available. In practice, most gates have to be approximated, for instance using the Solovay-Kiteav theorem, which gives a small overhead. Similarly, some gates are easier to implement than others, such as the simple Pauli-gates and the harder T-gate. Hence, we can imagine that we have different success probabilities for different gates being applied. Note that this idea aligns with the third direction for future research mentioned.

An aspect we only briefly touched upon, but which might prove vital in future applications of quantum computing is the time aspect. Especially on the near-term, quantum computers will most likely only solve small tasks and have to finish their computations quickly before the qubits decohere completely. Furthermore, with short quantum computation times, the cost of having to rerun a quantum circuit because it failed is small. The LAQCC-circuit provides one way to overcome this barrier by giving low-depth quantum circuits with intermediate conventional computations. With the uncertain development of future quantum computers, we can imagine that short quantum circuits can still prove useful, even if fault-tolerant quantum computation is out of reach.

List of symbols

The following list gives symbols and notation used throughout this work:

- $[n] = \{1, \ldots, n\}$
- $\omega_p = e^{i2\pi/p}$
- $\Delta_h f(x) = f(x+h) - f(x)$ the additive derivative of $f : \mathbb{F}_p^n \to \mathbb{C}$
- $\Delta_h f(x) = f(x+h)\overline{f(x)}$ the multiplicative derivative of $f : \mathbb{F}_p^n \to \mathbb{C}$
- $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$ the inner product of vectors $x, y \in \mathbb{F}_p^n$
- $\oplus$ represents addition modulo 2
- $\mathbb{D}$ represents the complex unit disc

The following definitions concern the behavior of functions $f, g : \mathbb{N} \to \mathbb{R}$:

- $f = \mathcal{O}(g)$ means there exist constants $c, n_0 > 0$ such that $|f(n)| \leq cg(n)$ for all $n \geq n_0$;
- $f = \Omega(g)$ means there exist constants $c, n_0 > 0$ such that $|f(n)| \geq cg(n)$ for all $n \geq n_0$;
- $f = o(g)$ means for every constant $c > 0$, there exists a constant $n_0 > 0$ such that $|f(n)| < cg(n)$ for all $n \geq n_0$;
- $f = \omega(g)$ means for every constant $c > 0$, there exists a constant $n_0 > 0$ such that $|f(n)| > cg(n)$ for all $n \geq n_0$;
- $f = \Theta(g)$ means $f = \mathcal{O}(g)$ and $f = \Omega(g)$;

- $\mathrm{D}\mathcal{O}_\varepsilon(f)$ denotes a depends of $\mathcal{O}(f)$ on a constant ε, similarly for Ω, o, ω and Θ;
- $\widetilde{\mathcal{O}}$ hides a dependency on logarithmic factors of the argument, similarly for Ω, o, ω and Θ;
- $\mathrm{poly}(f)$ denotes some unspecified polynomial in f;
- $\exp(f)$ denotes some unspecified exponential in f.

Finally,

- $\mathrm{Pol}_{\leq d}(\mathbb{F}_p^n, \mathbb{F}_p^k)$ denotes the space of all polynomial maps $\phi : \mathbb{F}_p^n \to \mathbb{F}_p^k$ of degree at most d;
- $\mathrm{arank}_d(\phi)$ denotes the analytic rank of polynomial maps $\phi \in \mathrm{Pol}_{\leq d}(\mathbb{F}_p^n, \mathbb{F}_p^k)$ (Theorem 3.3.2);
- $\mathrm{arank}(T)$ denotes the analytic rank of a tensor T (Theorem 3.7.1);
- $\mathrm{prank}(T)$ denotes the partition rank of a tensor T (Theorem 3.7.2).

Samenvatting

Adaptieve Kwantumcomputers: decoderen en toestanden maken

Het eerste idee van computers stamt uit het einde van de negentiende eeuw. In de loop der jaren is er aanzienlijke vooruitgang geboekt in de fysieke realisaties van computers. Tegenwoordig kunnen we ons nauwelijks een wereld zonder computers voorstellen en gebruiken we ze voor veel verschillende dingen in ons dagelijks leven. Er wordt veel onderzoek verricht naar het vergroten van de rekenkracht van computers en het verkennen van nieuwe manieren om berekeningen uit te voeren. Een veelbelovende manier is kwantumcomputers. Toekomstige kwantumcomputers hebben de potentie om specifieke problemen aanzienlijk sneller op te lossen dan huidige methoden. Deze kwantumcomputers zullen moeten samenwerken met een gewone computer om effectief te kunnen functioneren.

Huidige kwantumcomputers zijn nog in ontwikkeling en hebben op dit moment beperkte mogelijkheden. De interactie met een gewone computer kan nu echter al de capaciteit van kwantumcomputers verbeteren, met name door sommige berekeningen uit te laten voeren door een gewone computer. Kwantumcomputers die samenwerken met gewone computers om berekeningen uit te voeren, worden *adaptieve kwantumcomputers* genoemd. Dit werk formaliseert een model dat deze adaptieve kwantumcomputers beschrijft. Aangezien kwantumcomputers nog in ontwikkeling zijn, richten we ons op berekeningen die na een vast aantal stappen eindigen. Waarschijnlijk maakt dit de implementatie van de berekeningen in de praktijk makkelijker. Dit werk is verdeeld in twee hoofdonderdelen.

Het eerste deel toont aan dat adaptieve kwantumcomputers krachtiger zijn dan standaardcomputers. Het richt zich op het praktische probleem van het verkrijgen van informatie uit gecorrumpeerde digitale gegevens. Gewone computers hebben

moeite om dit soort informatie te verkrijgen in een vast aantal berekeningsstappen. Het bewijs hiervoor maakt gebruik van een structuur-versus-willekeur-methode, waarbij het probleem opgesplitst wordt in een gestructureerd en een willekeurige component. De kracht van adaptieve kwantumberekeningen blijkt uit een specifiek voorbeeld waarbij informatie wordt verkregen uit gecorrumpeerde digitale gegevens. Daarnaast kunnen adaptieve kwantumcomputers ook gewone berekeningen verbeteren, zelfs als deze niet beperkt zijn tot een vast aantal berekeningsstappen.

Het tweede deel beschrijft hoe adaptieve kwantumberekeningen niet-adaptieve kwantumberekeningen kunnen verbeteren, bijvoorbeeld door verschillende kwantumberekeningen sneller uit te voeren. Met behulp van deze snellere gewone berekeningen kunnen adaptieve kwantumcomputers ook kwantumtoestanden efficiënter maken dan niet-adaptieve tegenhangers. Kwantumtoestanden vormen de computationele eenheden van een kwantumcomputer, waardoor het maken ervan inherent kwantum is. Dit werk presenteert efficiënte adaptieve kwantumberekeningen om kwantumtoestanden zoals de uniforme superpositietoestand, de GHZ-toestand, de W-toestand en de Dicke-toestand te maken. Deze toestanden worden vaak gebruikt in andere kwantumalgoritmen, waardoor het hebben van efficiënte routines om deze toestanden te maken ook andere algoritmen efficiënter maakt. Dit werk eindigt met een vergelijking van deze adaptieve kwantumberekeningen met niet-adaptieve kwantumberekeningen, waarbij de prestaties zowel theoretisch als via kwantumhardware-implementaties worden geanalyseerd.

Abstract

Adaptive Quantum Computers: decoding and state preparation

The concept of computers dates back to the late 19th century. Over the years, significant progress has been made towards their physical realization. Today, we can hardly imagine a world without computers and we use them for a wide range of applications in our daily life. Extensive research efforts focus on enhancing the power of computers and exploring new ways of performing computations. One promising approach is quantum computing. Future quantum computers have the potential to solve specific problems significantly faster than current methods. These quantum computers will have to interact with a standard computer to operate effectively.

Current quantum computers are still under development and have limited capabilities. However, the interaction with a standard computer can already enhance their functionality, particularly by offloading certain computations to the standard computer. Quantum computers that interact with standard computers to perform computations are called *adaptive quantum computers*. This work formalizes a model that describes these adaptive quantum computers. As quantum computers are still under development, this work focuses on computations that terminate after a fixed number of steps, as that makes their implementation likely easier in practice. The work is divided into two main parts.

The first part shows that adaptive quantum computers are more powerful than standard computers. It focuses on the practical problem of retrieving information from corrupted digital data. Standard computers struggle to retrieve such information within a fixed number of computation steps. The proof uses a structure-versus-randomness approach that splits the problem in a structured and

a random-like component. The potential of adaptive quantum computations follows from a specific example where information is retrieved from corrupted data. Additionally, adaptive quantum computers can even improve standard computations for this problem that are not constrained by a fixed number of computation steps.

The second part explores how adaptive quantum computations can improve non-adaptive quantum computations, for instance by performing various quantum computations faster. Using these faster computations, adaptive quantum computers can also prepare quantum states more efficiently than their non-adaptive counterparts. Quantum states describe the computational units of a quantum computer, making the task of preparing them inherently quantum. This work presents efficient adaptive quantum computations to prepare quantum states such as the uniform superposition state, the GHZ state, the W-state and the Dicke state. These states are often used in other quantum algorithms, so having efficient routines for preparing them also enhance the efficiency of other algorithms. This work concludes by comparing these adaptive quantum computations with non-adaptive ones, analyzing their performance both theoretically and through quantum hardware implementations.

Acknowledgments

Writing this acknowledgment section was one of the harder sections of this dissertation to write, as every time I thought I had somewhat of an overview of those who contributed, new names popped up in my head. Therefore, I would like to start by thanking everybody that contributed to this dissertation, either directly or indirectly.

First, I would like to thank Harry Buhrman, my promotor. From the start I valued you openness, directness, and the great ideas you had. You taught me to think beyond obvious solutions and how to link fields within quantum technology that initially seemed completely distinct. Our many white-board sessions on the couches, as well as our discussion on co-design and the (scientific) world helped shape the work in this dissertation and shape my views on the (future) developments of quantum computing. Thank you for being the supervisor you were and hopefully we continue to collaborate while you are at Quantinuum.

I also would like to show my deep gratitude to Jop Briët. Thank you for the many talks we have had on a wide range of topics, including our own research, other research topics, and non-work topics. I greatly value how you always made time for a chat or a discussion on research problems, even if they extended beyond the scope of our joint projects. Also thank you for introducing me to the field on higher-order Fourier analysis and additive combinatorics, two fields within mathematics that often have very simple statements and very complicated proofs.

My thanks to Frank Phillipson extend beyond this dissertation alone. Since my first steps within TNO (already nine and a half years ago), you provided a calm and realistic perspective on matters. At the same time, you always asked just the right questions to keep our research progressing (even, or especially, if I convinced

myself that we had solved it). I highly value you feedback and presence both within and outside our (applied) research endeavors.

Next, I would like to thank my committee members Steven Girvin, Guido Schaefer, Kareljan Schoutens, Florian Speelman and John van de Wetering, for carefully reading the book and providing valuable feedback.

The research in this dissertation was produced while being employed by TNO. TNO provided me the freedom to explore whether a PhD was the new challenge I was looking for. I am grateful for the research managers that helped me along the way: Paul de Jager, Daniëlle Keus, Annemieke Kips, Adri Krabbendam, Milena Kooij-Janic, Dick van Smirren and Daniël Worm. Daniëlle, and later Milena and Daniël, helped me shape my project work load such that my TNO and PhD work could be combined efficiently. Dick was always there for sought-after guidance, support, reflections on personal and professional development, or just a simple tea chat about trains.

I would also like to thank my other TNO colleagues. Alessandro Amadori, Thomas Attema (thank you for somewhat paving the way by going through the process two years earlier), John Beerends (due to our work on speech perception, an online video call will never be the same), Juan Boschero, Oskar van Deventer, Peter Elias-van den Berg, Maran van Heesch (giving the final nudge to contact Harry), Rob Kooij, Ward van der Schoot, Marc van Vliet, Robert Wezeman and many others created a great atmosphere. This atmosphere allowed me to pursue novel research ideas, even if they were nothing more than a hunch something might work.

Next to research, there was ample time for non-work discussions and tea chats that made the time at TNO even more enjoyable. Thanks to Bart Kamphorst, Maaike van Leuken, Jacintha Moons, Bert-Jan te Paske, Thomas Rooijakkers, Erieke Weitenberg, and many others for taking the time to discuss serious and slightly less serious business, or to just play a game (or two) of chess. Also thanks to Robert Seepers, the co-inventor of Chess-darts–*A healthy mind requires a healthy body. Chess-darts provides both*–for the many talks and chess games both within and outside TNO.

My thanks also goes to the QuSoft environment. The welcome felt like a warm bath, even when in the pandemic times most contacts were virtual. The time at QuSoft was filled with many fun, interesting and exciting moments, with the definite highlight being the yearly QuSoft retreats. The QuSoft time was filled with many pleasant interactions for which I would like to thank everybody. Thank you to Lynn Engelberts, Marten Folkertsma, Dyon van Vreumingen and Jordi Weggemans, the co-occupants of the Dutch corner office; let us hope that our research qualities far exceed our qualities in keeping the office plants alive for more than a month. Thanks to my coauthors Davi Castro-Silva, Marten Folkertsma

and Bruno Löff for our collaborations that resulted in work now part of this dissertation and to Amira Abbas, Chris Cade, Yanlin Chen, Francisco Escudero Gutíerrez, Jonas Helsen, Garazi Muguruza Lasa, Ido Niessen, Freek Witteveen and many others for insightful discussions that led to new research ideas.

I thank Marc Hoeijmans for helping design the cover of this work. Your enthusiasm and creativity helped to make my vague ideas on the cover into concrete visuals.

It is very easy to lose yourself in your research and forget about other things in life. Luckily, many people prevented this from happening by providing the necessary distractions outside of my PhD and TNO life.

More often than not, time outside of research was spent doing (something related to) judo. During this time, my mind could often wander freely, resulting in new ideas for algorithms or proofs. Judo brought me to many places in the world and gave me the privilege to meet many wonderful people. First of all, thanks to Sander Stammers for creating the perfect atmosphere for others to shine. I also thank Pia de Kramer, Tjeerd van de Put, and Peter Tümmers (I will definitely check hotel descriptions before booking from now on) for being the amazing teachers and friends they are, both on the tatami and outside. Thank you for sharing these moments and enjoying our shared passion.

A name missing in the previous paragraph is Erik Faes, the one with whom many of my judo adventures were carried out (and probably the one I have fallen for most often). You always listened to me going on about some proof I was working on or some algorithm I was explaining. Thank you for being the friend you are and the amazing adventures we had together.

I also thank my friends outside my judo endeavors, Bram, Joost and Niek; Kimberly; Laura and Renée; and, Anouk and Luuk. You provided relaxation and joy in various ways, including many board game nights.

Further, I would like to thank my family. My parents, Frank and Jacqueline, my sister, Lyane, and my grandparents, who always supported me in pursuing my dreams and ambitions. This support helped me to be the person I am today. Thank you for that.

My final thanks goes to Frédérique. The last years have been amazing and I cannot wait to have many more amazing moments together. The many times I was late due to a meeting running longer than expected or I had to finish just one thing late at night–which usually was not 'just one thing'–you were there with your support. Without you the time would have definitely not been this interesting, for that I cannot thank you enough.

Tilburg Niels M. P. Neumann
March, 2025.

Bibliography

[AA13] S. Aaronson and A. Arkhipov. "The Computational Complexity of Linear Optics". In: *Theory of Computing* 9 (2013), pp. 143–252. DOI: 10.4086/toc.2013.v009a004.

[AA15] S. Aaronson and A. Ambainis. "Forrelation: A Problem that Optimally Separates Quantum from Classical Computing". In: *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*. STOC '15. Portland, Oregon, USA: Association for Computing Machinery, 2015, pp. 307–316. ISBN: 9781450335362. DOI: 10.1145/2746539.2746547.

[Aar04] S. Aaronson. "Multilinear Formulas and Skepticism of Quantum Computing". In: *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*. STOC '04. Chicago, IL, USA: Association for Computing Machinery, 2004, pp. 118–127. ISBN: 1581138520. DOI: 10.1145/1007352.1007378.

[Aar10] S. Aaronson. "BQP and the polynomial hierarchy". In: *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*. STOC '10. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2010, pp. 141–150. ISBN: 9781450300506. DOI: 10.1145/1806689.1806711.

[AAS20] S. Aaronson, Y. Atia, and L. Susskind. *On the Hardness of Detecting Macroscopic Superpositions*. 2020. arXiv: 2009.07450 [quant-ph].

[AB09] S. Arora and B. Barak. *Computational complexity*. A modern approach. Cambridge University Press, Cambridge, 2009. ISBN: 978-0-521-42426-4. DOI: 10.1017/CBO9780511804090.

[AB97] D. Aharonov and M. Ben-Or. "Fault-tolerant quantum computation with constant error". In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*. STOC '97. El Paso, Texas, USA: Association for Computing Machinery, 1997, pp. 176–188. ISBN: 0897918886. DOI: 10.1145/258533.258579.

[ABD24] S. Arunachalam, S. Bravyi, and A. Dutt. *A note on polynomial-time tolerant testing stabilizer states*. 2024. arXiv: 2410.22220 [quant-ph].

[AC02] M. Adcock and R. Cleve. "A Quantum Goldreich-Levin Theorem with Cryptographic Applications". In: *STACS 2002*. Ed. by H. Alt and A. Ferreira. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 323–334. ISBN: 978-3-540-45841-8. DOI: 10.1007/3-540-45841-7_26.

[AC98] C. Adami and N. J. Cerf. "Quantum Computation with Linear Optics". In: *Selected Papers from the First NASA International Conference on Quantum Computing and Quantum Communications*. QCQC '98. Berlin, Heidelberg: Springer-Verlag, 1998, pp. 391–401. ISBN: 354065514X.

[Ach+24] R. Acharya et al. "Quantum error correction below the surface code threshold". In: *Nature* (Dec. 2024). ISSN: 1476-4687. DOI: 10.1038/s41586-024-08449-y.

[AD24] S. Arunachalam and A. Dutt. *Polynomial-time tolerant testing stabilizer states*. 2024. arXiv: 2408.06289 [quant-ph].

[Aha+07] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. "Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation". In: *SIAM Journal on Computing* 37 (2007), pp. 166–194. DOI: 10.1137/S0097539705447323.

[Ajt83] M. Ajtai. "Σ^1_1-formulae on finite structures". In: *Annals of Pure and Applied Logic* 24 (1983), pp. 1–48. ISSN: 0168-0072. DOI: 10.1016/0168-0072(83)90038-6.

[AK07] S. Aaronson and G. Kuperberg. "Quantum versus Classical Proofs and Advice". In: *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*. 2007, pp. 115–128. DOI: 10.1109/CCC.2007.27.

[AKG05] S. Aaronson, G. Kuperberg, and C. Granade. *The complexity zoo*. 2005. URL: https://www.complexityzoo.net/Complexity_Zoos.

[AKS04] M. Agrawal, N. Kayal, and N. Saxena. "PRIMES is in P". In: *Annals of Mathematics* 160 (Sept. 2004), pp. 781–793. ISSN: 0003-486X. DOI: 10.4007/annals.2004.160.781.

[Alm+17] C. G. Almudever, L. Lao, X. Fu, N. Khammassi, I. Ashraf, D. Iorga, S. Varsamopoulos, C. Eichler, A. Wallraff, L. Geck, A. Kruth, J. Knoch, H. Bluhm, and K. Bertels. "The engineering challenges in quantum computing". In: *Design, Automation & Test in Europe*

Conference & Exhibition (DATE), 2017. 2017, pp. 836–845. DOI: 10.23919/DATE.2017.7927104.

[Ara+21] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva. "A divide-and-conquer algorithm for quantum state preparation". In: *Sci Rep* 11 (2021), p. 6329. DOI: 10.1038/s41598-021-85474-1.

[Aru+19] F. Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574 (Oct. 2019), pp. 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5.

[AS06] P. Arrighi and L. Salvail. "Blind quantum computation". In: *International Journal of Quantum Information* 4 (Oct. 2006), pp. 883–898. DOI: 10.1142/s0219749906002171.

[Bab82] C. Babbage. "On the Theoretical Principles of the Machinery for Calculating Tables". In: *Babbage's Calculating Engines: Being a Collection of Papers Relating to them; their History and Construction*. Ed. by H. P. Babbage. Cambridge Library Collection - Mathematics. Cambridge University Press, 1882, pp. 216–219.

[Ban+19] T. Bannink, J. Briët, H. Buhrman, F. Labib, and T. Lee. "Bounding Quantum-Classical Separations for Classes of Nonlocal Games". In: *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*. Ed. by R. Niedermeier and C. Paul. Vol. 126. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 12:1–12:11. ISBN: 978-3-95977-100-9. DOI: 10.4230/LIPIcs.STACS.2019.12.

[Bar+13] R. Barends et al. "Coherent Josephson Qubit Suitable for Scalable Quantum Integrated Circuits". In: *Phys. Rev. Lett.* 111 (Aug. 2013), p. 080502. DOI: 10.1103/PhysRevLett.111.080502.

[BC24] J. Briët and D. Castro-Silva. "Random restrictions of high-rank tensors and polynomial maps". In: *Discrete Analysis* (Nov. 2024). DOI: 10.19086/da.124610.

[BCH06] D. Bacon, I. L. Chuang, and A. W. Harrow. "Efficient Quantum Circuits for Schur and Clebsch-Gordan Transforms". In: *Phys. Rev. Lett.* 97 (Oct. 2006), p. 170502. DOI: 10.1103/PhysRevLett.97.170502.

[BCN25] J. Briët, D. Castro-Silva, and N. M. P. Neumann. "Quadratic Fourier analysis in quantum algorithms". Unpublished, to be submitted. 2025.

[BDH24] Z. Bao, P. van Dordrecht, and J. Helsen. *Tolerant testing of stabilizer states with a polynomial gap via a generalized uncertainty relation*. 2024. arXiv: 2410.21811 [quant-ph].

[BE19] A. Bärtschi and S. Eidenbenz. "Deterministic Preparation of Dicke States". In: *Fundamentals of Computation Theory*. Springer International Publishing, 2019, pp. 126–139. ISBN: 978-3-030-25027-0.

[BE22] A. Bärtschi and S. Eidenbenz. "Short-Depth Circuits for Dicke State Preparation". In: *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2022, pp. 87–96. DOI: 10.1109/QCE53715.2022.00027.

[Bec64] E. F. Beckenbach. *Applied combinatorial mathematics*. New York, J. Wiley, 1964, pp. 27–30.

[Bel64] J. S. Bell. "On the Einstein Podolsky Rosen paradox". In: *Physics Physique Fizika* 1 (Nov. 1964), pp. 195–200. DOI: 10.1103/PhysicsPhysiqueFizika.1.195.

[Ben+19] A. Bene Watts, R. Kothari, L. Schaeffer, and A. Tal. "Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2019. Phoenix, AZ, USA: Association for Computing Machinery, 2019, pp. 515–526. ISBN: 9781450367059. DOI: 10.1145/3313276.3316404.

[Ben73] C. H. Bennett. "Logical Reversibility of Computation". In: *IBM Journal of Research and Development* 17 (1973), pp. 525–532. DOI: 10.1147/rd.176.0525.

[Ben80] P. A. Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *J Stat Phys* 22 (May 1980), pp. 563–591. DOI: 10.1007/BF01011339.

[Ben82] P. A. Benioff. "Quantum mechanical Hamiltonian models of discrete processes that erase their own histories: Application to Turing machines". In: *Int J Theor Phys* 21 (Apr. 1982), pp. 177–201. DOI: 10.1007/bf01857725.

[Bez+21] A. V. Bezvershenko, C.-M. Halati, A. Sheikhan, C. Kollath, and A. Rosch. "Dicke Transition in Open Many-Body Systems Determined by Fluctuation Effects". In: *Phys. Rev. Lett.* 127 (Oct. 2021), p. 173606. DOI: 10.1103/PhysRevLett.127.173606.

[BGK18] S. Bravyi, D. Gosset, and R. König. "Quantum advantage with shallow circuits". In: *Science* 362 (Oct. 2018), pp. 308–311. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aar3106.

[BJS10] M. J. Bremner, R. Jozsa, and D. J. Shepherd. "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy". In: *Proc. R. Soc. A.* 467 (Aug. 2010), pp. 459–472. DOI: 10.1098/rspa.2010.0301.

[BKP11] D. Browne, E. Kashefi, and S. Perdrix. "Computational Depth Complexity of Measurement-Based Quantum Computation". In: *Theory of Quantum Computation, Communication, and Cryptography*. Ed. by W. van Dam, V. M. Kendon, and S. Severini. Berlin, Heidelberg:

Springer Berlin Heidelberg, 2011, pp. 35–46. ISBN: 978-3-642-18073-6. DOI: 10.1007/978-3-642-18073-6_4.

[BMS17] M. J. Bremner, A. Montanaro, and D. J. Shepherd. "Achieving quantum supremacy with sparse and noisy commuting quantum computations". In: *Quantum* 1 (Apr. 2017), p. 8. ISSN: 2521-327X. DOI: 10.22331/q-2017-04-25-8.

[BN25] H. Buhrman and N. M. P. Neumann. "Error-analysis of state preparation protocols using LAQCC". Unpublished work, to be submitted. 2025.

[Bon+23] L. J. Bond, M. J. Davis, J. Minář, R. Gerritsma, G. K. Brennen, and A. Safavi-Naini. *Efficient State Preparation for Metrology and Quantum Error Correction with Global Control*. 2023. arXiv: 2312.05060 [quant-ph].

[BP23] A. Bene Watts and N. Parham. *Unconditional Quantum Advantage for Sampling with Shallow Circuits*. 2023. arXiv: 2301.00995 [quant-ph].

[Bra+20] S. Bravyi, D. Gosset, R. König, and M. Tomamichel. "Quantum advantage with noisy shallow circuits". In: *Nat. Phys.* 16 (2020). Preliminary version in FOCS'19, pp. 1040–1045. ISSN: 1745-2473, 1745-2481. DOI: 10.1038/s41567-020-0948-z.

[Bra+22] S. Brandhofer, D. Braun, V. Dehn, G. Hellstern, M. Hüls, Y. Ji, I. Polian, A. S. Bhatia, and T. Wellens. "Benchmarking the performance of portfolio optimization with QAOA". In: *Quantum Inf Process* 22 (Dec. 2022). DOI: 10.1007/s11128-022-03766-5.

[Bra13] T. Brandes. "Excited-state quantum phase transitions in Dicke superradiance models". In: *Phys. Rev. E* 88 (Sept. 2013), p. 032133. DOI: 10.1103/PhysRevE.88.032133.

[Bri+09] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. "Measurement-based quantum computation". In: *Nature Phys* 5 (Jan. 2009), pp. 19–26. ISSN: 1745-2481. DOI: 10.1038/nphys1157.

[Bri+24] J. Briët, H. Buhrman, D. Castro-Silva, and N. M. P. Neumann. "Noisy Decoding by Shallow Circuits with Parities: Classical and Quantum (Extended Abstract)". In: *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Vol. 287. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. DOI: 10.4230/LIPIcs.ITCS.2024.21.

[BRR17] M. Booth, S. P. Reinhardt, and A. Roy. *Partitioning Optimization Problems for Hybrid Classical/Quantum Execution*. Tech. rep. D-Wave Systems Inc., 2017, pp. 01–09.

[BS21] N. Bansal and M. Sinha. "k-forrelation optimally separates Quantum and classical query complexity". In: *Proceedings of the 53rd*

Annual ACM SIGACT Symposium on Theory of Computing. STOC 2021. Virtual, Italy: Association for Computing Machinery, 2021, pp. 1303–1316. ISBN: 9781450380539. DOI: 10.1145/3406325.3451040.

[BS94] A. Balog and E. Szemerédi. "A statistical theorem of set addition". In: *Combinatorica* 14 (Sept. 1994), pp. 263–268. ISSN: 1439-6912. DOI: 10.1007/bf01212974.

[BSS16] S. Bravyi, G. Smith, and J. A. Smolin. "Trading Classical and Quantum Computational Resources". In: *Phys. Rev. X* 6 (June 2016), p. 021043. DOI: 10.1103/PhysRevX.6.021043.

[Buh+24] H. Buhrman, M. Folkertsma, B. Loff, and N. M. P. Neumann. "State preparation by shallow circuits using feed forward". In: *Quantum* 8 (Dec. 2024), p. 1552. ISSN: 2521-327X. DOI: 10.22331/q-2024-12-09-1552.

[BV97] E. Bernstein and U. Vazirani. "Quantum Complexity Theory". In: *SIAM Journal on Computing* 26 (1997), pp. 1411–1473. DOI: 10.1137/S0097539796300921.

[Cai+23] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien. "Quantum error mitigation". In: *Rev. Mod. Phys.* 95 (Dec. 2023), p. 045005. DOI: 10.1103/RevModPhys.95.045005.

[CEB20] J. Cook, S. Eidenbenz, and A. Bärtschi. "The Quantum Alternating Operator Ansatz on Maximum k-Vertex Cover". In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE. 2020, pp. 83–92. DOI: 10.1109/QCE49297.2020.00021.

[Cer+21] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. "Variational quantum algorithms". In: *Nat Rev Phys* 3 (Sept. 2021), pp. 625–644. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00348-9.

[Che+24] S. Chen, W. Gong, Q. Ye, and Z. Zhang. *Stabilizer bootstrapping: A recipe for efficient agnostic tomography and magic estimation*. 2024. arXiv: 2408.06967 [quant-ph].

[CJL08] S. Clark, R. Jozsa, and N. Linden. "Generalized Clifford groups and simulation of associated quantum circuits". In: *Quantum Info. Comput.* 8 (Jan. 2008), pp. 106–126. ISSN: 1533-7146.

[Cla+69] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. "Proposed Experiment to Test Local Hidden-Variable Theories". In: *Phys. Rev. Lett.* 23 (Oct. 1969), pp. 880–884. DOI: 10.1103/PhysRevLett.23.880.

[Cle+04] R. Cleve, P. Høyer, B. Toner, and J. Watrous. "Consequences and limits of nonlocal strategies". In: *Proceedings. 19th IEEE An-*

nual Conference on Computational Complexity, 2004. IEEE. 2004, pp. 236–249. DOI: 10.1109/CCC.2004.1313847.

[Cle+13] R. Cleve, W. van Dam, M. Nielsen, and A. Tapp. "Quantum entanglement and the communication complexity of the inner product function". In: *Theoretical Computer Science* 486 (2013). Theory of Quantum Communication Complexity and Non-locality, pp. 11–19. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2012.12.012.

[Coj+21] A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden. "On the Possibility of Classical Client Blind Quantum Computing". In: *Cryptography* 5 (Jan. 2021), p. 3. DOI: 10.3390/cryptography5010003.

[Cop02] D. Coppersmith. *An approximate Fourier transform useful in quantum factoring.* First published as an IBM Internal Report in 1994. 2002. arXiv: quant-ph/0201067 [quant-ph].

[Cro+19] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta. "Validating quantum computers using randomized model circuits". In: *Phys. Rev. A* 100 (Sept. 2019), p. 032328. DOI: 10.1103/PhysRevA.100.032328.

[CSV21] M. Coudron, J. Stark, and T. Vidick. "Trading Locality for Time: Certifiable Randomness from Low-Depth Circuits". In: *Commun. Math. Phys.* 382 (2021), pp. 49–86. DOI: 10.1007/s00220-021-03963-w.

[CZ95] J. I. Cirac and P. Zoller. "Quantum Computations with Cold Trapped Ions". In: *Phys. Rev. Lett.* 74 (May 1995), pp. 4091–4094. DOI: 10.1103/PhysRevLett.74.4091.

[De 21] G. De Luca. "A Survey of NISQ Era Hybrid Quantum-Classical Machine Learning Research". In: *Journal of Artificial Intelligence and Technology* 2 (Dec. 2021), pp. 9–15. DOI: 10.37965/jait.2021.12002.

[Deu85] D. Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proc. R. Soc. Lond. A* 400 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070.

[Dic54] R. H. Dicke. "Coherence in Spontaneous Radiation Processes". In: *Phys. Rev.* 93 (Jan. 1954), pp. 99–110. DOI: 10.1103/PhysRev.93.99.

[Dij59] E. W. Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1 (Dec. 1959), pp. 269–271. DOI: 10.1007/BF01386390.

[DiV00] D. P. DiVincenzo. "The physical implementation of quantum computation". In: *Fortschritte der Physik* 48 (2000), pp. 771–783. DOI: https://doi.org/10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E.

[DJ92] D. Deutsch and R. Jozsa. "Rapid solution of problems by quantum computation". In: *Proc. R. Soc. Lond. A* 439 (Dec. 1992), pp. 553–558. ISSN: 2053-9177. DOI: 10.1098/rspa.1992.0167.

[DMN13] S. J. Devitt, W. J. Munro, and K. Nemoto. "Quantum error correction for beginners". In: *Rep. Prog. Phys.* 76 (2013), p. 076001. DOI: 10.1088/0034-4885/76/7/076001.

[DN06] C. M. Dawson and M. A. Nielsen. "The Solovay-Kitaev algorithm". In: *Quantum Info. Comput.* 6 (Jan. 2006), pp. 81–95. ISSN: 1533-7146.

[Dum+02] R. Dumke, M. Volk, T. Müther, F. B. J. Buchkremer, G. Birkl, and W. Ertmer. "Micro-optical Realization of Arrays of Selectively Addressable Dipole Traps: A Scalable Configuration for Quantum Computation with Atomic Qubits". In: *Phys. Rev. Lett.* 89 (Aug. 2002), p. 097903. DOI: 10.1103/PhysRevLett.89.097903.

[DVC00] W. Dür, G. Vidal, and J. I. Cirac. "Three qubits can be entangled in two inequivalent ways". In: *Phys. Rev. A* 62 (Nov. 2000), p. 062314. DOI: 10.1103/PhysRevA.62.062314.

[Ebl+23] D. Ebler, M. Horodecki, M. Marciniak, T. Młynik, M. T. Quintino, and M. Studziński. "Optimal Universal Quantum Circuits for Unitary Complex Conjugation". In: *IEEE Transactions on Information Theory* 69 (2023), pp. 5069–5082. DOI: 10.1109/TIT.2023.3263771.

[Eis+00] J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio. "Optimal local implementation of nonlocal quantum gates". In: *Phys. Rev. A* 62 (Oct. 2000), p. 052317. DOI: 10.1103/PhysRevA.62.052317.

[Eli57] P. Elias. "List Decoding for Noisy Channels". In: *IRE WESCON Convention Record, 1957*. Vol. 2. 1957, pp. 94–104.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Phys. Rev.* 47 (May 1935), pp. 777–780. DOI: 10.1103/PhysRev.47.777.

[ER59] P. Erdös and A. Rényi. "On random graphs I". In: *Publ. math. debrecen* 6 (1959), p. 18. DOI: 10.5486%2FPMD.1959.6.3-4.12.

[Eur24] EuroHPC JU. *Leading the Way in European Supercomputing*. 2024. URL: https://eurohpc-ju.europa.eu/index_en (visited on 08/27/2024).

[Fan+24] J.-K. Fang, Y.-F. Lin, J.-H. Huang, Y. Chen, G.-M. Fan, Y. Sun, G. Feng, C. Guo, T. Meng, Y. Zhang, X. Xu, J. Xiang, and Y. Li. "Divide-and-Conquer Quantum Algorithm for Hybrid *denovo* Genome Assembly of Short and Long Reads". In: *PRX Life* 2 (Apr. 2024), p. 023006. DOI: 10.1103/PRXLife.2.023006.

[Fey82] R. P. Feynman. "Simulating physics with computers". In: *Int J Theor Phys* 21 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179.

[FGG14] E. Farhi, J. Goldstone, and S. Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph].

[FH61] R. M. Fano and D. Hawkins. "Transmission of information: A statistical theory of communications". In: *Am. J. Phys.* 29 (1961), pp. 793–794. DOI: 10.1119/1.1937609.

[Fou88] J. B. J. Fourier. *Théorie analytique de la chaleur*. Vol. 1. Gauthier-Villars, 1888. DOI: 10.1017/CBO9780511693229.

[Gar+23] G. García-Pérez, O. Kerppo, M. A. C. Rossi, and S. Maniscalco. "Experimentally accessible nonseparability criteria for multipartite-entanglement-structure detection". In: *Phys. Rev. Res.* 5 (Mar. 2023), p. 013226. DOI: 10.1103/PhysRevResearch.5.013226.

[GC99] D. Gottesman and I. L. Chuang. "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations". In: *Nature* 402 (1999), pp. 390–393. DOI: 10.1038/46503.

[GE21] C. Gidney and M. Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". In: *Quantum* 5 (Apr. 2021), p. 433. ISSN: 2521-327X. DOI: 10.22331/q-2021-04-15-433.

[GL89] O. Goldreich and L. A. Levin. "A hard-core predicate for all one-way functions". In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. STOC '89. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 25–32. ISBN: 0897913078. DOI: 10.1145/73007.73010.

[Goo23] Google Quantum AI. "Suppressing quantum errors by scaling a surface code logical qubit". In: *Nature* 614 (2023), pp. 676–681. DOI: 10.1038/s41586-022-05434-1.

[Got98] D. Gottesman. *The Heisenberg Representation of Quantum Computers*. 1998. arXiv: quant-ph/9807006 [quant-ph].

[Gow+25] W. T. Gowers, B. Green, F. Manners, and T. Tao. *On a conjecture of Marton*. To appear in Annals of Mathematics. 2025. arXiv: 2311.05762 [math.NT].

[Gow01] W. Gowers. "A new proof of Szemerédi's theorem". In: *GAFA, Geom. funct. anal.* 11 (Aug. 2001), pp. 465–588. DOI: 10.1007/s00039-001-0332-9.

[Gow98] W. T. Gowers. "A New Proof of Szemerédi's Theorem for Arithmetic Progressions of Length Four". In: *GAFA, Geom. funct. anal.* 8 (July 1998), pp. 529–551. DOI: 10.1007/s000390050065.

[Gre+02] F. Green, S. Homer, C. Moore, and C. Pollett. "Counting, Fanout and the Complexity of Quantum *ACC*". In: *Quantum Info. Comput.* 2 (Dec. 2002), pp. 35–65. ISSN: 1533-7146.

[Gre07] B. Green. "Montréal notes on quadratic Fourier analysis". In: *Additive Combinatorics*. Vol. 43. 2007, pp. 69–102.

[Gro96] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866.

[GRS22] V. Guruswami, A. Rudra, and M. Sudan. *Essential Coding Theory*. 2022. URL: https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/.

[GS20] D. Grier and L. Schaeffer. "Interactive Shallow Clifford Circuits: Quantum Advantage against NC^1 and Beyond". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC'20)*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 875–888. ISBN: 9781450369794. DOI: 10.1145/3357713.3384332.

[GT08] B. Green and T. Tao. "An inverse theorem for the Gowers $U^3(G)$ norm". In: *Proceedings of the Edinburgh Mathematical Society* 51 (Feb. 2008), pp. 73–153. ISSN: 1464-3839. DOI: 10.1017/s0013091505000325.

[GT09] B. Green and T. Tao. "New bounds for Szemerédi's theorem, I: progressions of length 4 in finite field geometries". In: *Proceedings of the London Mathematical Society* 98 (2009), pp. 365–392. DOI: https://doi.org/10.1112/plms/pdn030.

[GTZ12] B. Green, T. Tao, and T. Ziegler. "An inverse theorem for the Gowers $U^{s+1}[N]$-norm". In: *Annals of Mathematics* 176 (Sept. 2012), pp. 1231–1372. DOI: 10.4007/annals.2012.176.2.11.

[GV10] V. Guruswami and S. Vadhan. "A Lower Bound on List Size for List Decoding". In: *IEEE Transactions on Information Theory* 56 (2010), pp. 5681–5688. DOI: 10.1109/TIT.2010.2070170.

[GW11] W. T. Gowers and J. Wolf. "Linear forms and higher-degree uniformity for functions on $\mathbb{F}_p^n$". In: *Geom. Funct. Anal.* 21 (2011), pp. 36–69. ISSN: 1016-443X. DOI: 10.1007/s00039-010-0106-3.

[Had+19] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas. "From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz". In: *Algorithms* 12 (Feb. 2019), p. 34. DOI: 10.3390/a12020034.

[Häf+05] H. Häffner, W. Hänsel, C. F. Roos, J. Benhelm, D. Chek-al-kar, M. Chwalla, T. Körber, U. D. Rapol, M. Riebe, P. O. Schmidt, C. Becher, O. Gühne, W. Dür, and R. Blatt. "Scalable multiparticle entanglement of trapped ions". In: *Nature* 438 (Dec. 2005), pp. 643–646. DOI: 10.1038/nature04279.

[Ham50] R. W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29 (1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[Hen+15] B. Hensen et al. "Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres". In: *Nature* 526 (Oct. 2015), pp. 682–686. ISSN: 1476-4687. DOI: 10.1038/nature15759.

[HHL09] A. W. Harrow, A. Hassidim, and S. Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Phys. Rev. Lett.* 103 (Oct. 2009), p. 150502. DOI: 10.1103/PhysRevLett.103.150502.

[HHL19] H. Hatami, P. Hatami, and S. Lovett. "Higher-order Fourier analysis and applications". In: *Found. Trends Theor. Comput. Sci.* 13 (2019), pp. 247–448. ISSN: 1551-305X. DOI: 10.1561/0400000064.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[Hoe63] W. Hoeffding. "Probability Inequalities for Sums of Bounded Random Variables". In: *Journal of the American Statistical Association* 58 (1963), pp. 13–30. DOI: 10.1080/01621459.1963.10500830.

[Hou+09] A. A. Houck, J. Koch, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. "Life after charge noise: recent results with transmon qubits". In: *Quantum Inf Process* 8 (2009), pp. 105–115. DOI: 10.1007/s11128-009-0100-6.

[HR90] T. Hagerup and C. Rüb. "A guided tour of Chernoff bounds". In: *Information Processing Letters* 33 (1990), pp. 305–308. ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(90)90214-I.

[HŠ05] P. Høyer and R. Špalek. "Quantum Fan-out is Powerful". In: *Theory of Computing* 1 (2005), pp. 81–103. DOI: 10.4086/toc.2005.v001a005.

[IBM25] IBM Quantum Computing. *IBM Quantum*. 2025. URL: https://quantum.ibm.com/ (visited on 01/16/2025).

[IDM18] IDM. *Types of Cloud Services*. Aug. 9, 2018. URL: https://medium.com/@IDMdatasecurity/types-of-cloud-services-b54e5b574f6 (visited on 12/06/2021).

[Ima+99] A. Imamog¯lu, D. D. Awschalom, G. Burkard, D. P. DiVincenzo, D. Loss, M. Sherwin, and A. Small. "Quantum Information Processing Using Quantum Dot Spins and Cavity QED". In: *Phys. Rev. Lett.* 83 (Nov. 1999), pp. 4204–4207. DOI: 10.1103/PhysRevLett.83.4204.

[Joh90] D. S. Johnson. "A Catalog of Complexity Classes". In: *Algorithms and Complexity*. Elsevier, 1990, pp. 67–161. DOI: 10.1016/b978-0-444-88071-0.50007-2.

[Joz06] R. Jozsa. "An introduction to measurement based quantum computation". In: *NATO Science Series, III: Computer and Systems Sciences. Quantum Information Processing-From Theory to Experiment* 199 (2006), pp. 137–158.

[KA22] J. S. Kottmann and A. Aspuru-Guzik. "Optimized low-depth quantum circuits for molecular electronic structure using a separable-pair

approximation". In: *Phys. Rev. A* 105 (Mar. 2022), p. 032449. DOI: 10.1103/PhysRevA.105.032449.

[Kal16] G. Kalai. "The Quantum Computer Puzzle". In: *Notices of the American Mathematical Society* 63 (May 2016), pp. 508–516. ISSN: 1088-9477. DOI: 10.1090/noti1380.

[Kal20] G. Kalai. "The argument against quantum computers". In: *Quantum, probability, logic: The work and influence of Itamar Pitowsky* (2020), pp. 399–422.

[Kan+19] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta. "Error mitigation extends the computational reach of a noisy quantum processor". In: *Nature* 567 (2019), pp. 491–495. DOI: 10.1038/s41586-019-1040-7.

[Kit97] A. Y. Kitaev. "Quantum computations: algorithms and error correction". In: *Russian Mathematical Surveys* 52 (Dec. 1997), p. 1191. DOI: 10.1070/RM1997v052n06ABEH002155.

[Kja+20] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver. "Superconducting Qubits: Current State of Play". In: *Annual Review of Condensed Matter Physics* 11 (2020), pp. 369–395. DOI: 10.1146/annurev-conmatphys-031119-050605.

[KL08] T. Kaufman and S. Lovett. "Worst Case to Average Case Reductions for Polynomials". In: *2008 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*. 2008, pp. 166–175. DOI: 10.1109/FOCS.2008.17.

[KLM01] E. Knill, R. Laflamme, and G. J. Milburn. "A scheme for efficient quantum computation with linear optics". In: *Nature* 409 (Jan. 2001), pp. 46–52. ISSN: 1476-4687. DOI: 10.1038/35051009.

[KLO04] W. M. Kaminsky, S. Lloyd, and T. P. Orlando. *Scalable Superconducting Architecture for Adiabatic Quantum Computation*. 2004. arXiv: quant-ph/0403090 [quant-ph].

[KLT23] D. Kim, A. Li, and J. Tidor. "Cubic Goldreich-Levin". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, Jan. 2023, pp. 4846–4892. DOI: 10.1137/1.9781611977554.ch178.

[KLZ98] E. Knill, R. Laflamme, and W. H. Zurek. "Resilient quantum computation: error models and thresholds". In: *Proc. R. Soc. Lond. A.* 454 (Jan. 1998), pp. 365–384. ISSN: 1471-2946. DOI: 10.1098/rspa.1998.0166.

[KN98] T. Kadowaki and H. Nishimori. "Quantum annealing in the transverse Ising model". In: *Phys. Rev. E* 58 (Nov. 1998), pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355.

[KSK19] M. Kaiiali, S. Sezer, and A. Khalid. "Cloud computing in the quantum era". In: *2019 IEEE Conference on Communications and Net-*

work Security (CNS). 2019, pp. 1–4. DOI: 10.1109/CNS44998.2019.8952589.

[KY10] A. Kawachi and T. Yamakami. "Quantum Hardcore Functions by Complexity-Theoretical Quantum List Decoding". In: *SIAM Journal on Computing* 39 (2010), pp. 2941–2969. DOI: 10.1137/080716840.

[Lan12] G. P. Lansbergen. "Transistors arrive at the atomic limit". In: *Nature Nanotech* 7 (Feb. 2012), pp. 209–210. ISSN: 1748-3395. DOI: 10.1038/nnano.2012.23.

[Le 19] F. Le Gall. "Average-Case Quantum Advantage with Shallow Circuits". In: *34th Computational Complexity Conference (CCC 2019)*. Ed. by A. Shpilka. Vol. 137. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 21:1–21:20. ISBN: 978-3-95977-116-0. DOI: 10.4230/LIPIcs.CCC.2019.21.

[Li+17] J. Li, X. Yang, X. Peng, and C.-P. Sun. "Hybrid Quantum-Classical Approach to Quantum Optimal Control". In: *Phys. Rev. Lett.* 118 (Apr. 2017), p. 150503. DOI: 10.1103/PhysRevLett.118.150503.

[LL24] J. Luo and L. Li. *Circuit Complexity of Sparse Quantum State Preparation*. 2024. arXiv: 2406.16142 [quant-ph].

[Lon01] G. L. Long. "Grover algorithm with zero theoretical failure rate". In: *Phys. Rev. A* 64 (July 2001), p. 022307. DOI: 10.1103/PhysRevA.64.022307.

[Lov19] S. Lovett. "The analytic rank of tensors and its applications". In: *Discrete Anal.* (2019), Paper No. 7, 10. DOI: 10.19086/da.8654.

[LU05] M. Lanzagorta and J. K. Uhlmann. "Hybrid quantum-classical computing with applications to computer graphics". In: *ACM SIGGRAPH 2005 Courses*. SIGGRAPH '05. Los Angeles, California: Association for Computing Machinery, 2005, 2–es. ISBN: 9781450378338. DOI: 10.1145/1198555.1198723.

[Lub+23] T. Lubinski, S. Johri, P. Varosy, J. Coleman, L. Zhao, J. Necaise, C. H. Baldwin, K. Mayer, and T. Proctor. "Application-Oriented Performance Benchmarks for Quantum Computing". In: *IEEE Transactions on Quantum Engineering* 4 (2023), pp. 1–32. DOI: 10.1109/TQE.2023.3253761.

[LV08] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer New York, 2008. DOI: 10.1007/978-0-387-49820-1.

[LV17] C. H. Lee and E. Viola. "Some Limitations of the Sum of Small-Bias Distributions". In: *Theory of Computing* 13 (2017), pp. 1–23. DOI: 10.4086/toc.2017.v013a016.

[MAA21] S. Martiel, T. Ayral, and C. Allouche. "Benchmarking Quantum Coprocessors in an Application-Centric, Hardware-Agnostic, and

Scalable Way". In: *IEEE Transactions on Quantum Engineering* 2 (2021), pp. 1–11. ISSN: 2689-1808. DOI: 10.1109/tqe.2021.3090207.

[Mad+22] L. S. Madsen et al. "Quantum computational advantage with a programmable photonic processor". In: *Nature* 606 (June 2022), pp. 75–81. ISSN: 1476-4687. DOI: 10.1038/s41586-022-04725-x.

[MAM22] K. Mesman, Z. Al-Ars, and M. Möller. *QPack: Quantum Approximate Optimization Algorithms as universal benchmark for quantum computers*. 2022. arXiv: 2103.17193 [cs.ET].

[McC+20] A. J. McCaskey, D. I. Lyakh, E. F. Dumitrescu, S. S. Powers, and T. S. Humble. "XACC: a system-level software infrastructure for heterogeneous quantum–classical computing". In: *Quantum Science and Technology* 5 (Feb. 2020), p. 024002. DOI: 10.1088/2058-9565/ab6bf6.

[Med+12] J. Medford, Ł. Cywi ński, C. Barthel, C. M. Marcus, M. P. Hanson, and A. C. Gossard. "Scaling of Dynamical Decoupling for Spin Qubits". In: *Phys. Rev. Lett.* 108 (Feb. 2012), p. 086802. DOI: 10.1103/PhysRevLett.108.086802.

[Mes+24] K. J. Mesman, W. van der Schoot, M. Möller, and N. M. P. Neumann. "QuAS: Quantum Application Score for Benchmarking the Utility of Quantum Computers". In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. 2024, pp. 921–929. DOI: 10.1109/QCE60285.2024.00111.

[MGE11] E. Magesan, J. M. Gambetta, and J. Emerson. "Scalable and Robust Randomized Benchmarking of Quantum Processes". In: *Phys. Rev. Lett.* 106 (May 2011), p. 180504. DOI: 10.1103/PhysRevLett.106.180504.

[MN01] C. Moore and M. Nilsson. "Parallel Quantum Computation and Quantum Codes". In: *SIAM Journal on Computing* 31 (2001), pp. 799–815. DOI: 10.1137/S0097539799355053.

[Mon12] A. Montanaro. "The quantum query complexity of learning multilinear polynomials". In: *Information Processing Letters* 112 (2012), pp. 438–442. ISSN: 0020-0190. DOI: https://doi.org/10.1016/j.ipl.2012.03.002.

[Moo65] G. Moore. "Cramming More Components Onto Integrated Circuits". In: *Proceedings of the IEEE* (Apr. 1965). DOI: 10.1109/JPROC.1998.658762.

[Moo99] C. Moore. *Quantum Circuits: Fanout, Parity, and Counting*. 1999. arXiv: quant-ph/9903046 [quant-ph].

[MR18] D. Maslov and M. Roetteler. "Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations". In: *IEEE Transactions on Information Theory* 64 (2018), pp. 4729–4738. DOI: 10.1109/TIT.2018.2825602.

[MTS24] R. Mao, G. Tian, and X. Sun. "Toward optimal circuit size for sparse quantum state preparation". In: *Phys. Rev. A* 110 (Sept. 2024), p. 032439. DOI: 10.1103/PhysRevA.110.032439.

[Mul54] D. E. Muller. "Application of Boolean algebra to switching circuit design and to error detection". In: *Transactions of the I.R.E. Professional Group on Electronic Computers* EC-3 (1954), pp. 6–12. DOI: 10.1109/IREPGELC.1954.6499441.

[MY23] T. Metger and H. Yuen. "stateQIP = statePSPACE". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 1349–1356. DOI: 10.1109/FOCS57990.2023.00082.

[MZ22] G. Moshkovitz and D. G. Zhu. *Quasi-linear relation between partition and analytic rank*. arXiv:2211.05780. 2022. DOI: 10.48550/ARXIV.2211.05780.

[Nas20] E. Naslund. "The partition rank of a tensor and k-right corners in $\mathbb{F}_q^n$". In: *J. Combin. Theory Ser. A* 174 (2020), pp. 105190, 25. ISSN: 0097-3165. DOI: 10.1016/j.jcta.2019.105190.

[Nay+08] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma. "Non-Abelian anyons and topological quantum computation". In: *Rev. Mod. Phys.* 80 (Sept. 2008), pp. 1083–1159. DOI: 10.1103/RevModPhys.80.1083.

[NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[Nee+10] M. Neeley, R. C. Bialczak, M. Lenander, E. Lucero, M. Mariantoni, A. D. O'Connell, D. Sank, H. Wang, M. Weides, J. Wenner, Y. Yin, T. Yamamoto, A. N. Cleland, and J. M. Martinis. "Generation of three-qubit entangled states using superconducting phase qubits". In: *Nature* 467 (Sept. 2010), pp. 570–573. DOI: 10.1038/nature09418.

[Neu45] J. von Neumann. "First draft of a report on the EDVAC". In: *IEEE Annals of the History of Computing* 15 (1945), pp. 27–75. DOI: 10.1109/85.238389.

[New78] I. Newton. *Philosophiae Naturalis Principia Mathematica*. Londini: Jussu Societatis Regiæ ac Typis Joseph Streater, 1678.

[Nie+21] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout. "Gate Set Tomography". In: *Quantum* 5 (Oct. 2021), p. 557. ISSN: 2521-327X. DOI: 10.22331/q-2021-10-05-557.

[NM14] Y. Nakata and M. Murao. "Diagonal quantum circuits: Their computational power and applications". In: *The European Physical Journal Plus* 129 (July 2014). DOI: 10.1140/epjp/i2014-14152-9.

[NSS23] N. M. P. Neumann, W. van der Schoot, and T. Sijpesteijn. "Quantum Cloud Computing from a User Perspective". In: *Innovations for Community Services*. Ed. by U. R. Krieger, G. Eichler, C. Erfurth, and G. Fahrnberger. Cham: Springer Nature Switzerland, 2023, pp. 236–249. ISBN: 978-3-031-40852-6. DOI: 10.1007/978-3-031-40852-6_13.

[NW22] N. M. P. Neumann and R. S. Wezeman. "Distributed Quantum Machine Learning". In: *Innovations for Community Services*. Ed. by F. Phillipson, G. Eichler, C. Erfurth, and G. Fahrnberger. Cham: Springer International Publishing, 2022, pp. 281–293. ISBN: 978-3-031-06668-9. DOI: 10.1007/978-3-031-06668-9_20.

[ÖSI07] S. K. Özdemir, J. Shimamura, and N. Imoto. "A necessary and sufficient condition to play games in quantum mechanical settings". In: *New J. Phys.* 9 (Feb. 2007), pp. 43–43. DOI: 10.1088/1367-2630/9/2/043.

[Ouy14] Y. Ouyang. "Permutation-invariant quantum codes". In: *Phys. Rev. A* 90 (Dec. 2014), p. 062317. DOI: 10.1103/PhysRevA.90.062317.

[Par06] M.-A. Parseval. "Mémoire sur les séries et sur l'intégration complète d'une équation aux différences partielles linéaires du second ordre, à coefficients constants". In: *Mém. prés. par divers savants, Acad. des Sciences, Paris,(1)* 1 (1806), p. 42.

[PB10] M. Plesch and V. Bužek. "Efficient compression of quantum information". In: *Phys. Rev. A* 81 (Mar. 2010), p. 032317. DOI: 10.1103/PhysRevA.81.032317.

[PCZ22] F. Pan, K. Chen, and P. Zhang. "Solving the Sampling Problem of the Sycamore Quantum Circuits". In: *Phys. Rev. Lett.* 129 (Aug. 2022), p. 090502. DOI: 10.1103/PhysRevLett.129.090502.

[Ped+19] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, and R. Wisnieff. *Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits*. 2019. arXiv: 1910.09534 [quant-ph].

[Per+14] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature Communications* 5 (July 2014), p. 4213. ISSN: 2041-1723. DOI: 10.1038/ncomms5213.

[Pér+23] A. Pérez-Salinas, R. Dra ški ć, J. Tura, and V. Dunjko. "Shallow quantum circuits for deeper problems". In: *Phys. Rev. A* 108 (Dec. 2023), p. 062423. DOI: 10.1103/PhysRevA.108.062423.

[Pia+21] M. Piattini, M. Serrano, R. Perez-Castillo, G. Petersen, and J. L. Hevia. "Toward a Quantum Software Engineering". In: *IT Professional* 23 (2021), pp. 62–66. DOI: 10.1109/MITP.2020.3019522.

[PNW23] F. Phillipson, N. Neumann, and R. Wezeman. "Classification of Hybrid Quantum-Classical Computing". In: *Computational Science –*

ICCS 2023. Ed. by J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot. Cham: Springer Nature Switzerland, 2023, pp. 18–33. ISBN: 978-3-031-36030-5. DOI: 10.1007/978-3-031-36030-5_2.

[Pok+18] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar. "Demonstration of Fidelity Improvement Using Dynamical Decoupling with Superconducting Qubits". In: *Phys. Rev. Lett.* 121 (Nov. 2018), p. 220502. DOI: 10.1103/PhysRevLett.121.220502.

[PP21] C. Psaroudaki and C. Panagopoulos. "Skyrmion Qubits: A New Class of Quantum Logic Elements Based on Nanoscale Magnetization". In: *Phys. Rev. Lett.* 127 (Aug. 2021), p. 067201. DOI: 10.1103/PhysRevLett.127.067201.

[Pre+09] R. Prevedel, G. Cronenberg, M. S. Tame, M. Paternostro, P. Walther, M. S. Kim, and A. Zeilinger. "Experimental Realization of Dicke States of up to Six Qubits for Multiparty Quantum Networking". In: *Phys. Rev. Lett.* 103 (July 2009), p. 020503. DOI: 10.1103/PhysRevLett.103.020503.

[Pre18] J. Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79.

[PRK01] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge Nonlinear Science Series. Cambridge University Press, 2001. DOI: 10.1017/CBO9780511755743.

[Pro+21] T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout. "Measuring the capabilities of quantum computers". In: *Nat. Phys.* 18 (Dec. 2021), pp. 75–79. DOI: 10.1038/s41567-021-01409-7.

[PS13] P. Pham and K. M. Svore. "A 2D nearest-neighbor quantum architecture for factoring in polylogarithmic depth". In: *Quantum Info. Comput.* 13 (Nov. 2013), pp. 937–962. ISSN: 1533-7146.

[PSC21] L. Piroli, G. Styliaris, and J. I. Cirac. "Quantum Circuits Assisted by Local Operations and Classical Communication: Transformations and Phases of Matter". In: *Phys. Rev. Lett.* 127 (Nov. 2021), p. 220503. DOI: 10.1103/PhysRevLett.127.220503.

[Raz87] A. A. Razborov. "Lower bounds on the size of bounded depth circuits over a complete basis with logical addition". In: *Mathematical Notes of the Academy of Sciences of the USSR* 41 (Apr. 1987), pp. 333–338. DOI: 10.1007/bf01137685.

[RB01] R. Raussendorf and H. J. Briegel. "A One-Way Quantum Computer". In: *Phys. Rev. Lett.* 86 (May 2001), pp. 5188–5191. DOI: 10.1103/PhysRevLett.86.5188.

[Ree54] I. Reed. "A class of multiple-error-correcting codes and the decoding scheme". In: *Transactions of the IRE Professional Group on Information Theory* 4 (1954), pp. 38–49. DOI: 10.1109/TIT.1954.1057465.

[Rei+17] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer. "Elucidating reaction mechanisms on quantum computers". In: *Proc. Natl. Acad. Sci. U.S.A.* 114 (2017), pp. 7555–7560. DOI: 10.1073/pnas.1619152114.

[RG17] L. Ruiz-Perez and J. C. Garcia-Escartin. "Quantum arithmetic with the quantum Fourier transform". In: *Quantum Inf Process* 16 (Apr. 2017). ISSN: 1573-1332. DOI: 10.1007/s11128-017-1603-1.

[RLR24] D. Ramacciotti, A. I. Lefterovici, and A. F. Rotundo. "Simple quantum algorithm to efficiently prepare sparse states". In: *Phys. Rev. A* 110 (Sept. 2024), p. 032609. DOI: 10.1103/PhysRevA.110.032609.

[Rob49] J. B. Robinson. *On the Hamiltonian game (a traveling salesman problem)*. Rand Corporation, 1949.

[Röt09] M. Rötteler. "Quantum Algorithms to Solve the Hidden Shift Problem for Quadratics and for Functions of Large Gowers Norm". In: *Mathematical Foundations of Computer Science 2009*. Springer Berlin Heidelberg, 2009, pp. 663–674. DOI: 10.1007/978-3-642-03816-7_56.

[Rot53] K. F. Roth. "On Certain Sets of Integers". In: *Journal of the London Mathematical Society* 28 (1953), pp. 104–109. DOI: https://doi.org/10.1112/jlms/s1-28.1.104.

[RS60] I. S. Reed and G. Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of the Society for Industrial and Applied Mathematics* 8 (1960), pp. 300–304. DOI: 10.1137/0108018.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: *Commun. ACM* 21 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342.

[RU10] A. Rudra and S. Uurtamo. "Two Theorems on List Decoding". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Ed. by M. Serna, R. Shaltiel, K. Jansen, and J. Rolim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 696–709. ISBN: 978-3-642-15369-3.

[RY22] G. Rosenthal and H. S. Yuen. "Interactive Proofs for Synthesizing Quantum States and Unitaries". In: *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Ed. by M. Braverman. Vol. 215. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 112:1–112:4. ISBN: 978-3-95977-217-4. DOI: 10.4230/LIPIcs.ITCS.2022.112.

[Sam07] A. Samorodnitsky. "Low-Degree Tests at Large Distances". In: *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*. STOC '07. San Diego, California, USA: Association for Computing Machinery, 2007, pp. 506–515. ISBN: 9781595936318. DOI: 10.1145/1250790.1250864.

[SB09] D. J. Shepherd and M. J. Bremner. "Temporally unstructured quantum computation". In: *Proc. R. Soc. A.* 465 (2009), pp. 1413–1439. DOI: 10.1098/rspa.2008.0443.

[Sch+22] W. v. d. Schoot, D. Leermakers, R. Wezeman, N. Neumann, and F. Phillipson. "Evaluating the Q-score of Quantum Annealers". In: *2022 IEEE International Conference on Quantum Software (QSW)*. 2022, pp. 9–16. DOI: 10.1109/QSW55613.2022.00017.

[Sch+23] W. van der Schoot, R. Wezeman, P. T. Eendebak, N. M. Neumann, and F. Phillipson. "Evaluating three levels of quantum metrics on quantum-inspire hardware". In: *Quantum Inf Process* 22 (2023), p. 451. DOI: 10.1007/s11128-023-04184-x.

[Sch+24] W. van der Schoot, R. Wezeman, N. Neumann, F. Phillipson, and R. Kooij. "Extending the Q-Score to an Application-Level Quantum Metric Framework". In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. 2024, pp. 941–951. DOI: 10.1109/QCE60285.2024.00113.

[Sch14] T. Schoen. "New bounds in Balog-Szemerédi-Gowers theorem". In: *Combinatorica* 35 (Oct. 2014), pp. 695–701. DOI: 10.1007/s00493-014-3077-4.

[Sha48] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[Sho94] P. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[Sho95] P. W. Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Phys. Rev. A* 52 (Oct. 1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493.

[Sho97] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.

[Sil+24] M. P. da Silva et al. *Demonstration of logical qubits and repeated error correction with better-than-physical error rates*. 2024. arXiv: 2404.02280 [quant-ph].

[Sim97] D. R. Simon. "On the Power of Quantum Computation". In: *SIAM Journal on Computing* 26 (1997), pp. 1474–1483. DOI: 10.1137/S0097539796298637.

[Siu+93] K.-Y. Siu, J. Bruck, T. Kailath, and T. Hofmeister. "Depth efficient neural networks for division and related problems". In: *IEEE Transactions on Information Theory* 39 (1993), pp. 946–956. DOI: 10.1109/18.256501.

[Smo87] R. Smolensky. "Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity". In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC '87. New York, New York, USA: Association for Computing Machinery, 1987, pp. 77–82. ISBN: 0897912217. DOI: 10.1145/28395.28404.

[Spe18] E. Sperling. *Quantum Effects At 7/5nm And Beyond*. 2018. URL: https://semiengineering.com/quantum-effects-at-7-5nm/ (visited on 09/18/2024).

[SS14] H. Singh and A. Sachdev. "The Quantum way of Cloud Computing". In: *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*. 2014, pp. 397–400. DOI: 10.1109/ICROIT.2014.6798362.

[SSW23] A. A. Sherstov, A. A. Storozhenko, and P. Wu. "An Optimal Separation of Randomized and Quantum Query Complexity". In: *SIAM Journal on Computing* 52 (2023), pp. 525–567. DOI: 10.1137/22M1468943.

[STV99] M. Sudan, L. Trevisan, and S. Vadhan. "Pseudorandom generators without the XOR lemma". In: *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*. 1999, pp. 4–. DOI: 10.1109/CCC.1999.766253.

[Sun+23] X. Sun, G. Tian, S. Yang, P. Yuan, and S. Zhang. "Asymptotically Optimal Circuit Depth for Quantum State Preparation and General Unitary Synthesis". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42 (2023), pp. 3301–3314. DOI: 10.1109/TCAD.2023.3244885.

[Sus18] L. Susskind. *Three Lectures on Complexity and Black Holes*. 2018. arXiv: 1810.11563 [hep-th].

[Tan+24] N. Tantivasadakarn, R. Thorngren, A. Vishwanath, and R. Verresen. "Long-Range Entanglement from Measuring Symmetry-Protected Topological Phases". In: *Phys. Rev. X* 14 (June 2024), p. 021040. DOI: 10.1103/PhysRevX.14.021040.

[TBG17] K. Temme, S. Bravyi, and J. M. Gambetta. "Error Mitigation for Short-Depth Quantum Circuits". In: *Phys. Rev. Lett.* 119 (Nov. 2017), p. 180509. DOI: 10.1103/PhysRevLett.119.180509.

[TDL10] O. Tsyplyatyev, J. von Delft, and D. Loss. "Simplified derivation of the Bethe-ansatz equations for the Dicke model". In: *Phys. Rev. B* 82 (Sept. 2010), p. 092203. DOI: 10.1103/PhysRevB.82.092203.

[Tom+23] T. Tomesh, Z. H. Saleem, M. A. Perlin, P. Gokhale, M. Suchara, and M. Martonosi. "Divide and Conquer for Combinatorial Optimization and Distributed Quantum Computation". In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 1. 2023, pp. 1–12. DOI: 10.1109/QCE57702.2023.00009.

[Tót12] G. Tóth. "Multipartite entanglement and high-precision metrology". In: *Phys. Rev. A* 85 (Feb. 2012), p. 022322. DOI: 10.1103/PhysRevA.85.022322.

[Tre04] L. Trevisan. *Some Applications of Coding Theory in Computational Complexity*. 2004. arXiv: cs/0409044 [cs.CC].

[TT13] Y. Takahashi and S. Tani. "Collapse of the Hierarchy of Constant-Depth Exact Quantum Circuits". In: *2013 IEEE Conference on Computational Complexity*. 2013, pp. 168–178. DOI: 10.1109/CCC.2013.25.

[Tur37] A. M. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society* s2-42 (1937), pp. 230–265. DOI: https://doi.org/10.1112/plms/s2-42.1.230.

[TV06] T. Tao and V. H. Vu. *Additive Combinatorics*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006. DOI: 10.1017/CBO9780511755149.

[TV07] L. Trevisan and S. Vadhan. "Pseudorandomness and average-case complexity via uniform reductions". In: *comput. complex.* 16 (2007), pp. 331–364. DOI: 10.1007/s00037-007-0233-x.

[TVV23] N. Tantivasadakarn, R. Verresen, and A. Vishwanath. "Shortest Route to Non-Abelian Topological Order on a Quantum Processor". In: *Phys. Rev. Lett.* 131 (Aug. 2023), p. 060405. DOI: 10.1103/PhysRevLett.131.060405.

[TW14] M. Tulsiani and J. Wolf. "Quadratic Goldreich–Levin Theorems". In: *SIAM Journal on Computing* 43 (Jan. 2014), pp. 730–766. DOI: 10.1137/12086827x.

[TZ11] T. Tao and T. Ziegler. "The Inverse Conjecture for the Gowers Norm over Finite Fields in Low Characteristic". In: *Annals of Combinatorics* 16 (Dec. 2011), pp. 121–188. ISSN: 0219-3094. DOI: 10.1007/s00026-011-0124-3.

[Van+21] J. S. Van Dyke, G. S. Barron, N. J. Mayhall, E. Barnes, and S. E. Economou. "Preparing Bethe ansatz eigenstates on a quantum computer". In: *PRX Quantum* 2 (2021), p. 040329. DOI: 10.1103/PRXQuantum.2.040329.

[Vio06] E. Viola. "The Complexity of Hardness Amplification and Derandomization". AAI3217914. PhD thesis. USA: Harvard University, 2006. ISBN: 9780542694301.

[Vio09] E. Viola. "The sum of d small-bias generators fools polynomials of degree d". In: *comput. complex.* 18 (2009). Preliminary version in CCC'08, pp. 209–217. ISSN: 1016-3328. DOI: 10.1007/s00037-009-0273-5.

[VKL99] L. Viola, E. Knill, and S. Lloyd. "Dynamical Decoupling of Open Quantum Systems". In: *Phys. Rev. Lett.* 82 (Mar. 1999), pp. 2417–2421. DOI: 10.1103/PhysRevLett.82.2417.

[VL98] L. Viola and S. Lloyd. "Dynamical suppression of decoherence in two-state quantum systems". In: *Phys. Rev. A* 58 (Oct. 1998), pp. 2733–2744. DOI: 10.1103/PhysRevA.58.2733.

[VMS04] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa. "Efficient Decomposition of Quantum Gates". In: *Phys. Rev. Lett.* 92 (Apr. 2004), p. 177902. DOI: 10.1103/PhysRevLett.92.177902.

[Vol99] H. Vollmer. *Introduction to circuit complexity: a uniform approach.* Springer, 1999. DOI: 10.1007/978-3-662-03927-4.

[Wac+21] A. Wack, H. Paik, A. Javadi-Abhari, P. Jurcevic, I. Faro, J. M. Gambetta, and B. R. Johnson. *Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers.* 2021. arXiv: 2110.14108 [quant-ph].

[Wan+12a] Y. Wang, Z. Lü, F. Glover, and J.-K. Hao. "A Multilevel Algorithm for Large Unconstrained Binary Quadratic Optimization". In: *Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimzation Problems.* Springer. 2012, pp. 395–408. DOI: 10.1007/978-3-642-29828-8_26.

[Wan+12b] Y. Wang, Z. Lü, F. Glover, and J.-K. Hao. "Path relinking for unconstrained binary quadratic programming". In: *European Journal of Operational Research* 223 (2012), pp. 595–604. DOI: 10.1016/j.ejor.2012.07.012.

[Wan+20] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais. "Qudits and High-Dimensional Quantum Computing". In: *Frontiers in Physics* 8 (Nov. 2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2020.589504.

[WE16] J. J. Wallman and J. Emerson. "Noise tailoring for scalable quantum computation via randomized compiling". In: *Phys. Rev. A* 94 (Nov. 2016), p. 052325. DOI: 10.1103/PhysRevA.94.052325.

[Wed+21] B. Weder, J. Barzen, F. Leymann, and M. Zimmermann. "Hybrid Quantum Applications Need Two Orchestrations in Superposition: A Software Architecture Perspective". In: *2021 IEEE International Conference on Web Services (ICWS).* 2021, pp. 1–13. DOI: 10.1109/ICWS53863.2021.00015.

[Wig19] A. Wigderson. *Mathematics and computation: A theory revolutionizing technology and science*. Princeton University Press, 2019. DOI: 10.2307/j.ctvckq7xb.

[Woz58] J. M. Wozencraft. "List decoding". In: *Quarterly Progress Report* 48 (1958), pp. 90–95.

[Yam16] T. Yamakami. "Quantum List Decoding of Classical Block Codes of Polynomially Small Rate from Quantumly Corrupted Codewords". In: *Baltic J. Modern Computing* 4 (2016), pp. 753–788. DOI: 10.22364/bjmc.2016.4.4.12.

[Yao77] A. C.-C. Yao. "Probabilistic computations: Toward a unified measure of complexity". In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. 1977, pp. 222–227. DOI: 10.1109/SFCS.1977.24.

[YL04] A. Yimsiriwattana and J. Lomonaco. "Generalized GHZ States and Distributed Quantum Computing". In: *Coding Theory and Quantum Computing* 381 (Mar. 2004). DOI: 10.48550/arXiv.quant-ph/0402148.

[You20] R. Youssef. "Measuring and Simulating T1 and T2 for Qubits". In: *Conference at Fermi National Accelerator Lab.* Aug. 2020. DOI: 10.2172/1656632.

[Zha+22] P. Zhao, K. Linghu, Z. Li, P. Xu, R. Wang, G. Xue, Y. Jin, and H. Yu. "Quantum Crosstalk Analysis for Simultaneous Gate Operations on Superconducting Qubits". In: *PRX Quantum* 3 (Apr. 2022), p. 020301. DOI: 10.1103/PRXQuantum.3.020301.

[Zha23] Y. Zhao. *Graph theory and additive combinatorics—exploring structure and randomness*. Cambridge University Press, Cambridge, 2023, pp. xvii+316. ISBN: 978-1-009-31094-9. DOI: 10.1017/9781009310956.

[Zho+20] H.-S. Zhong et al. "Quantum computational advantage using photons". In: *Science* 370 (Dec. 2020), pp. 1460–1463. ISSN: 1095-9203. DOI: 10.1126/science.abe8770.

[Zom+24] M. Zomorodi, H. Amini, M. Abbaszadeh, J. Sohrabi, V. Salari, and P. Plawiak. *Optimal Quantum Circuit Design via Unitary Neural Networks*. 2024. arXiv: 2408.13211 [quant-ph].

Titles in the ILLC Dissertation Series:

ILLC DS-2020-12: **Bastiaan van der Weij**
Experienced listeners: Modeling the influence of long-term musical exposure on rhythm perception

ILLC DS-2020-13: **Thom van Gessel**
Questions in Context

ILLC DS-2020-14: **Gianluca Grilletti**
Questions & Quantification: A study of first order inquisitive logic

ILLC DS-2020-15: **Tom Schoonen**
Tales of Similarity and Imagination. A modest epistemology of possibility

ILLC DS-2020-16: **Ilaria Canavotto**
Where Responsibility Takes You: Logics of Agency, Counterfactuals and Norms

ILLC DS-2020-17: **Francesca Zaffora Blando**
Patterns and Probabilities: A Study in Algorithmic Randomness and Computable Learning

ILLC DS-2021-01: **Yfke Dulek**
Delegated and Distributed Quantum Computation

ILLC DS-2021-02: **Elbert J. Booij**
The Things Before Us: On What it Is to Be an Object

ILLC DS-2021-03: **Seyyed Hadi Hashemi**
Modeling Users Interacting with Smart Devices

ILLC DS-2021-04: **Sophie Arnoult**
Adjunction in Hierarchical Phrase-Based Translation

ILLC DS-2021-05: **Cian Guilfoyle Chartier**
A Pragmatic Defense of Logical Pluralism

ILLC DS-2021-06: **Zoi Terzopoulou**
Collective Decisions with Incomplete Individual Opinions

ILLC DS-2021-07: **Anthia Solaki**
Logical Models for Bounded Reasoners

ILLC DS-2021-08: **Michael Sejr Schlichtkrull**
Incorporating Structure into Neural Models for Language Processing

ILLC DS-2021-09: **Taichi Uemura**
Abstract and Concrete Type Theories

ILLC DS-2021-10: **Levin Hornischer**
Dynamical Systems via Domains: Toward a Unified Foundation of Symbolic and Non-symbolic Computation

ILLC DS-2021-11: **Sirin Botan**
Strategyproof Social Choice for Restricted Domains

ILLC DS-2021-12: **Michael Cohen**
Dynamic Introspection

ILLC DS-2021-13: **Dazhu Li**
Formal Threads in the Social Fabric: Studies in the Logical Dynamics of Multi-Agent Interaction

ILLC DS-2021-14: **Álvaro Piedrafita**
On Span Programs and Quantum Algorithms

ILLC DS-2022-01: **Anna Bellomo**
Sums, Numbers and Infinity: Collections in Bolzano's Mathematics and Philosophy

ILLC DS-2022-02: **Jan Czajkowski**
Post-Quantum Security of Hash Functions

ILLC DS-2022-03: **Sonia Ramotowska**
Quantifying quantifier representations: Experimental studies, computational modeling, and individual differences

ILLC DS-2022-04: **Ruben Brokkelkamp**
How Close Does It Get?: From Near-Optimal Network Algorithms to Suboptimal Equilibrium Outcomes

ILLC DS-2022-05: **Lwenn Bussière-Carae**
No means No! Speech Acts in Conflict

ILLC DS-2022-06: **Emma Mojet**
Observing Disciplines: Data Practices In and Between Disciplines in the 19th and Early 20th Centuries

ILLC DS-2022-07: **Freek Gerrit Witteveen**
Quantum information theory and many-body physics

ILLC DS-2023-01: **Subhasree Patro**
Quantum Fine-Grained Complexity

ILLC DS-2023-02: **Arjan Cornelissen**
Quantum multivariate estimation and span program algorithms

ILLC DS-2023-03: **Robert Paßmann**
Logical Structure of Constructive Set Theories

ILLC DS-2023-04: **Samira Abnar**
Inductive Biases for Learning Natural Language

ILLC DS-2023-05: **Dean McHugh**
Causation and Modality: Models and Meanings

ILLC DS-2023-06: **Jialiang Yan**
Monotonicity in Intensional Contexts: Weakening and: Pragmatic Effects under Modals and Attitudes

ILLC DS-2023-07: **Yiyan Wang**
Collective Agency: From Philosophical and Logical Perspectives

ILLC DS-2023-08: **Lei Li**
Games, Boards and Play: A Logical Perspective

ILLC DS-2023-09: **Simon Rey**
Variations on Participatory Budgeting

ILLC DS-2023-10: **Mario Giulianelli**
Neural Models of Language Use: Studies of Language Comprehension and Production in Context

ILLC DS-2023-11: **Guillermo Menéndez Turata**
Cyclic Proof Systems for Modal Fixpoint Logics

ILLC DS-2023-12: **Ned J.H. Wontner**
Views From a Peak: Generalisations and Descriptive Set Theory

ILLC DS-2024-01: **Jan Rooduijn**
Fragments and Frame Classes: Towards a Uniform Proof Theory for Modal Fixed Point Logics

ILLC DS-2024-02: **Bas Cornelissen**
Measuring musics: Notes on modes, motifs, and melodies

ILLC DS-2024-03: **Nicola De Cao**
Entity Centric Neural Models for Natural Language Processing

ILLC DS-2024-04: **Ece Takmaz**
Visual and Linguistic Processes in Deep Neural Networks: A Cognitive Perspective

ILLC DS-2024-05: **Fatemeh Seifan**
Coalgebraic fixpoint logic Expressivity and completeness result

ILLC DS-2024-06: **Jana Sotáková**
Isogenies and Cryptography

ILLC DS-2024-07: **Marco Degano**
Indefinites and their values

ILLC DS-2024-08: **Philip Verduyn Lunel**
Quantum Position Verification: Loss-tolerant Protocols and Fundamental Limits

ILLC DS-2024-09: **Rene Allerstorfer**
Position-based Quantum Cryptography: From Theory towards Practice

ILLC DS-2024-10: **Willem Feijen**
Fast, Right, or Best? Algorithms for Practical Optimization Problems

ILLC DS-2024-11: **Daira Pinto Prieto**
Combining Uncertain Evidence: Logic and Complexity

ILLC DS-2024-12: **Yanlin Chen**
On Quantum Algorithms and Limitations for Convex Optimization and Lattice Problems

ILLC DS-2024-13: **Jaap Jumelet**
Finding Structure in Language Models

ILLC DS-2025-01: **Julian Chingoma**
On Proportionality in Complex Domains

ILLC DS-2025-02: **Dmitry Grinko**
Mixed Schur-Weyl duality in quantum information

ILLC DS-2025-03: **Rochelle Choenni**
Multilinguality and Multiculturalism: Towards more Effective and Inclusive Neural Language Models

ILLC DS-2025-04: **Aleksi Anttila**
Not Nothing: Nonemptiness in Team Semantics

ILLC DS-2025-05: **Niels M. P. Neumann**
Adaptive Quantum Computers: decoding and state preparation