# Empowering Quantum Computation with: Measurements, Catalysts, and Guiding States

Marten Folkertsma

# Empowering Quantum Computation with: Measurements, Catalysts, and Guiding States

Marten J. Folkertsma

# Empowering Quantum Computation with: Measurements, Catalysts, and Guiding States

ILLC Dissertation Series DS-2026-02

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/`

Cover design by Jasmijn Roeland and Marten Folkertma
Printed and bound ProefschriftMaken.

Empowering Quantum Computation with: Measurements, Catalysts, and
Guiding States

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 maandag 19 januari 2026, te 13.00 uur

door Marten Joran Folkertsma

geboren te Amsterdam

***Promotiecommissie***

| *Promotores:* | prof. dr. H.M. Buhrman | Universiteit van Amsterdam |
| | prof. dr. C.J.M. Schoutens | Universiteit van Amsterdam |
| | | |
| *Overige leden:* | dr. F. Speelman | Universiteit van Amsterdam |
| | prof. dr. F.E. Schreck | Universiteit van Amsterdam |
| | prof. dr. C. Schaffner | Universiteit van Amsterdam |
| | dr. J. Helsen | CWI |
| | prof. dr. M. Walter | Ruhr-Universität Bochum |

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

# Contents

# Part One: Measurements and fast intermediate classical calculations

# Part Two: Catalysis of space

## Part Three: Guiding states

# Chapter 1

# Introduction

Computation has been central to human progress for thousands of years. Ancient civilizations developed tools to perform arithmetic and solve practical problems. One of the earliest examples is the abacus, believed to have originated in Mesopotamia, which enabled efficient calculations such as addition and multiplication [Ifr00]. The invention of the modern computer marked a turning point, enabling computations to be carried out automatically—without human intervention or mistakes. A key insight in constructing these machines is that by combining many elementary steps, much more complex problems can be solved. The construction of these complex *procedures* is captured by the concept of an *algorithm*: a finite set of precise rules that, when followed step-by-step, transforms a given input into a desired output. The invention of the automatic execution of algorithms has led to remarkable advances in nearly every aspect of society, from the simulation of physical systems to worldwide communication.

The process of developing modern computers has progressed along multiple fronts, including hardware engineering, software development, and theoretical computer science. For example, the hardware that executes these elementary steps has been developed at an exponential rate, following Moore's law with a doubling of the number of transistors every two years [Moo+65]. In parallel, *computational science* focuses on developing more efficient algorithms and analyzing the resources required to solve specific problems.

The development of algorithms and hardware must often progress in tandem. A clear example of this is the rise of artificial intelligence, which has significantly influenced the design of modern computational devices. Most machine learning algorithms consist of many relatively simple operations—particularly large-scale matrix multiplications—that can be executed in parallel [Goo+16]. While CPUs (central processing units) are optimized for sequential and complex logic, GPUs (graphics processing units) excel at performing many simple tasks simultaneously, making them far more efficient for matrix-based computations. As a result, the recent success of machine learning has driven a shift in the types of resources

1

needed for efficient computation. For many years, the CPU was considered the most important component of a computer. Today, however, interest is growing in more powerful GPUs [CLS25] and in new architectures such as TPUs (tensor processing units), developed by Google specifically for machine learning tasks [Jou+17].

With the development of *quantum computers* on the horizon, the question of which *resources* are most important has become more relevant than ever. A quantum computer operates in a *fundamentally different* way from a classical one. Classical computers rely on the principles of classical physics and perform logical operations on binary strings. In contrast, *quantum computers* process information using microscopic physical systems, such as electrons or heavy ions, that behave according to the laws of *quantum mechanics.* This enables them to exploit uniquely quantum phenomena such as *superposition* and *entanglement*, which, when harnessed effectively, can lead to more efficient algorithms.
Feynman and Benioff were among the first to propose a computational model based on the rules of quantum physics for simulating quantum systems. They recognized the exponential overhead required to perform such simulations using traditional computational means [Fey82; Ben80].
However, a quantum computer is not simply a *faster* classical computer. In fact, quantum devices are often much slower on a *per-step* basis. Nevertheless, the use of superposition and entanglement allows quantum algorithms to solve certain problems in significantly fewer steps than any known classical algorithm.

The first example of a problem with a provable quantum advantage was given by Deutsch and Jozsa in 1992. They constructed a problem for which the best deterministic classical algorithm requires a number of steps that scales with the input size. In contrast, they developed a quantum algorithm that solves the problem in just five steps [Deu85]. A key ingredient of their quantum algorithm is the ability to ask a question to an oracle[1] in superposition. However, it should be noted, that this problem is somewhat contrived, and if one allows for randomized classical computation with a small probability of error, then the advantage gained by the quantum algorithm disappears.
The discovery of Deutsch and Jozsa sparked significant interest in the development of quantum algorithms. It was soon followed by more sophisticated problems and accompanying algorithms [BV93; Sim97], establishing the existence of exponential speed-ups by quantum algorithms, even compared to randomized classical computation[2]. Among the most notable results are the discoveries by Peter Shor and Lov Grover. Shor devised an efficient quantum algorithm for factoring large numbers—a problem for which no efficient classical algorithm is known—which

---

[1]An oracle is a black-box function, often encoding a specific problem instance. In many query-complexity settings (e.g., Deutsch–Jozsa, Simon), it maps a bit string $x$ to a bit $f(x) \in \{0, 1\}$ and can be queried on superpositions of inputs by a quantum algorithm.
[2]These problems, which allow for provable exponential speed-ups, are all defined with respect to an oracle.

forms the basis of modern cryptographic protocols such as RSA [Sho97]. Grover developed a quantum algorithm that achieves a quadratic speed-up for unstructured search, an algorithm with numerous applications [Gro96].

These discoveries prompted researchers to search for practical physical implementations of quantum computation. In classical computation, the *transistor*, an electronic implementation of an on/off switch, serves as the fundamental building block for storing the 1s and 0s (bits). However, it remains an open question which type of physical system best realizes the *qubit*, the fundamental unit of quantum information analogous to the classical bit. There are numerous physical systems that allow for storing this fundamental unit of quantum information. Some examples are trapped ions, neutral atoms, and superconducting circuits[Fos+24; Win+23; Kra+19]. Companies and universities alike are pursuing different physical realizations, aiming to build the largest quantum computational devices.

Even though the number of qubits has been steadily growing for many of these different realizations, they all face one fundamental problem. Quantum information is inherently *fragile*. Qubits suffer from *decoherence*, meaning that the information stored in a qubit is easily lost due to interactions with the environment around it [Alm+17]. One possible solution to this problem is to isolate the qubit in an environment with minimal noise, at cryogenic temperatures. However, even at such low temperatures, when running deeper circuits—such as those required to generate entanglement—qubits tend to accumulate too much noise and cause the system to decohere.

An algorithmic solution to this problem was originally suggested by Shor and Steane (who discovered it separately) and is the implementation of *quantum error correction*[Sho95; Ste96]. In quantum error correction one combines multiple *physical* qubits into one *logical* qubit in such a way that the errors in the logical qubit are exponentially suppressed. However, this requires the physical error rate to be below a certain threshold value $\eta_c$, which depends on the type of error-correcting code that is used [AB99]. This year (2025) we have seen the first results of small-scale quantum systems that can operate below this threshold value and thereby are able to run an error-correcting code for multiple cycles [Ach+25; Dag+25]. These small-scale experiments are a milestone on the road to *fault-tolerant* (error resilient) quantum computation. However, for running more sophisticated processes, one needs to scale these results to much larger systems. Estimates for the number of physical qubits required to form one single logical qubit tend to vary depending on the type of error correction code that is involved. For instance, in [Gid25] the authors estimate that to run Shor's algorithm they require combining 430 physical qubits to form one logical qubit using a surface code[3], the error correcting code used in [Ach+25]. This means that even the

---

[3]In the paper the author splits required logical qubits into two categories called hot and cold patches. These different patches have different requirements. The hot patches require significantly more physical qubits to create one logical qubit, in total 1352. However, the vast majority of logical qubits used are part of the cold patches, therefore we give the estimate for

largest devices can support at most a handful of logical qubits in the current state. Therefore, it will be necessary to scale the recent architectures to a sufficiently large size, before fully *fault-tolerant* quantum computation becomes available. A natural question to ask is:

> Are there other *resources* that could aid in the development of quantum computational devices?

In this thesis, we try to tackle this problem from a computational theoretical perspective.

## 1.1   Three computational resources

We identify three different points in the development of quantum computational devices and suggest three different resources that could aid in those settings.

### 1.1.1   Intermediate measurements

The first setting we investigate is that of pre-error correction quantum computation. In this situation we do not make use of error correction and instead try to perform computations with the physically available qubits. This imposes a severe limit on the achievable *depth* of quantum computation (i.e., how long a quantum computer can run) due to the build-up of errors accumulated through extended runs. Therefore, we consider only *constant-depth* quantum computation to be available. To combat the depth restriction, we suggest to make use of interspersed *measurements* and *fast* classical computations as an additional *resource* to facilitate quantum computation.

### 1.1.2   Catalysts

The second setting is that of *early fault-tolerant* quantum computation. This setting is identified by the access to a restricted number of *stable* logical qubits. Therefore, in this setting we focus on space-bounded rather than time-bounded computations. To combat these space constraints, we draw upon a concept developed in classical computational theory called *catalytic computation*. As an additional resource, a classical space-bounded computational device is given access to an additional hard drive, which at the start of the computation contains arbitrary data. The space-bounded machine can make use of this hard drive in whichever way it wants, but with the added caveat that it has to return the additional hard drive to its original state. Thus, the space-bounded machine can use the hard drive as a *catalyst*. Access to such a resource gives a surprising amount

---

the cold patches here.

of power to a classical space-bounded computational model. In this thesis, we begin the investigation of *quantum catalytic space*, asking whether similar results hold for the quantum setting.

### 1.1.3 Guiding states

The third setting is the application of quantum computers to the simulation of quantum chemistry systems. Many consider this one of the most promising applications of quantum computation [Aar09; Bau+20]. Whilst perhaps the original vision of the early pioneers of quantum computing was to simulate the time-dynamics of quantum systems [Fey82; Ben80], for many applications one is interested in stationary properties. One particularly noteworthy quantity is the *ground state energy* (which corresponds to the smallest eigenvalue) of a *local Hamiltonian* describing a quantum mechanical system of interest, say a small molecule or segment of material. It is well known that the problem of estimating this value up to some additive error is hard, even for fault-tolerant quantum computers. Therefore, it is generally believed that, without any additional help or structure, quantum computers are not able to accurately estimate the smallest eigenvalues of general local Hamiltonians, and there is some evidence that this hardness carries over to those Hamiltonians relevant to chemistry and materials science [OGo+22]. In the quantum chemistry community, it is often suggested that this extra help could come from a classical heuristic that first finds some form of *guiding state*: a classical description of a quantum state that can be used as an input to a quantum algorithm to compute the ground state energy accurately [Liu+22]. In this thesis, we investigate how such an additional resource, a *guiding state*, could help in the application of quantum computers to quantum chemistry.

The application of these resources is inspired by the three different stages of the development of quantum computational devices. However, all three can be applicable beyond these stages. Our goal is to analyze these three resources and draw a conclusion about their use with respect to quantum computational devices. We analyze them through the lens of *computational complexity theory*.

## 1.2 Complexity theory

In computational complexity theory one studies the relative power of computational models with respect to specific computational problems. The goal of complexity theory is to classify problems into sets, we call those sets *complexity classes*. Giving a concrete description of a complexity class requires us to first specify three things. First, we need to specify what type of computational problems, or type of functions, we try to characterize. Second, we need to define the computational model that attempts to solve these problems, specifying the type

of resources available for the computation. Last, we need to specify what measure to which we compare these models.

There are two specifications of computational problems used in this thesis. The first, and most used in the thesis, are what we call *decision problems*. To solve a decision problem, a computational device has to "decide" if a given input has a specific *property* or not. An example of such a problem would be to ask if a given number is odd or even (More details on decision problems found in Section 2.3 of the Preliminaries). The second type of problems we will care about, are quantum state preparation problems. In these types of problems, one asks what type of states a quantum computational model can prepare. These types of problems will be discussed more thoroughly in Chapter 3.

There are two main computational models that will be discussed in this thesis, both of which will be extended by giving them access to additional resources. The first is the well-known *Turing machine*, arguably the defining computational model in theoretical computer science. Introduced by Alan Turing in 1936 [Tur+36], a Turing machine is an abstract mathematical computation model capable of performing a sequence of elementary steps to solve complex logical problems. Turing showed that any computation carried out by a physical computational device—such as a modern-day computer—can also be performed by a Turing machine. This model can be made more powerful by providing it with additional resources, such as *randomness* and *nondeterminism*. (A more detailed discussion of Turing machines can be found in Section 2.3.1 of the preliminaries.) The definition of a Turing machine can also be extended to include access to *quantum operations*, resulting in a *quantum Turing machine*, which we will discuss briefly in Chapter 6. However, most of quantum complexity theory is not studied through this model, but rather through the *quantum circuit model*. A quantum circuit is a sequence of quantum gates, chosen from a finite set known as the gate set, applied to qubits in order to perform a computation (more precise definition of a quantum circuit in Section 2.2.6 of the preliminaries). This model more closely resembles actual quantum hardware implementations.

Finally, we will compare these models in two measures. The first measure is the amount of time required to perform a computation. For a Turing machine this is measured by counting the number of "elementary steps" that the Turing machine takes. In the quantum circuit model, this is the number of gates in the circuit. The second measure is the amount of space required to perform a computation. For a Turing machine, this is measured by the size of *work tape*, or the number of bits of working memory the Turing needs during computation. For the quantum circuit model this is measured by the number of qubits required for a computation. Sometimes we will not be interested in the actual total number of gates of a quantum circuit, but instead in its depth. Any sequential gates acting on a non-overlapping subset of the qubits are considered to be applied in parallel, on the same layer. The number of layers in a circuit is the circuit depth (more specifics on these measures can be found in Section 2.2.6 of the preliminaries).

With these three things specified, we can finally give an example of a complexity class: the complexity class $\mathsf{P}$. The complexity class $\mathsf{P}$ contains all *decision problems* for which there exists a *Turing machine* that can decide them in *polynomial time*. Polynomial time means that the number of steps the Turing machine has to take to solve the problem, scales at most polynomially with the input size (a more formal definition can be found in Section 2.3.2 of the preliminaries). In this thesis, we will make use of this machinery in the following way.

Above, we described three distinct computational settings where we propose the introduction of additional resources to aid computation. These settings correspond to different constraints on the underlying computational model. For instance, the first setting—pre-error-corrected quantum computation—adheres to the quantum circuit model with restricted depth, corresponding to the complexity class $\mathsf{QNC}^0$. We augment this model with an additional resource: measurements combined with fast, intermediate classical computations. We then compare the resulting complexity class to its unaugmented counterpart. We will demonstrate that in all three cases, the addition of the specified resource increases the power of the computational model.

## 1.3  Organization of the thesis

The organization of this thesis is as follows. In Chapter 2, we start out by giving the appropriate background for the rest of the thesis. This includes a description of the notation and terminology used, an introduction into the quantum computational model, and an introduction into complexity theory including the definition of the most important complexity theoretic models.

In Part One, we start by discussing the paradigm of short depth quantum computational circuits interspersed with measurements and fast classical computation. We define a new computational model that captures the power of this additional resource in constant-depth circuits. We call this model *Local Alternating Quantum Classical Computations*, or $\mathsf{LAQCC}$ for short. In Chapter 3 we start by giving a concrete definition of this model and compare it to known computational models with similar ideas behind it. We give a list of operations based on results from literature contained in this computational model. Then we discuss its relation with respect to other complexity theoretic classes in the context of what we call *state complexity*. Finally, we give explicit novel constant depth $\mathsf{LAQCC}$ circuits for four classes of states: the uniform superposition of size $q$ (with $q$ not a power of two), the $W$ state, the $(n, k)$-Dicke state, and a specific type of many-body scar states.

In Part Two, we dive into the field of space-bounded computation, and more specifically try to understand how having access to a full memory can aid in computation. We start by giving a concrete introduction of the classical *catalytic computational model* in Chapter 4. In Chapter 5, we deviate slightly from our

original path, discussing the classical model of catalytic computation in the presence of *errors*. We give a full characterization of the additional power that is gained by the catalytic model, when it is allowed to make a small number of errors on reset.

In Chapter 6, we initiate the study of the model of *quantum catalytic computation*. First, we give a robust definition of quantum catalytic computation, and focus on a specific instantiation of this model called *quantum catalytic log-space* (QCL for short). We try to compare the quantum catalytic model to its classical counterpart, however we find that this comparison is much less straightforward than expected. It turns out that the reset requirement in the quantum setting is significantly more stringent than in the classical setting, this allows us to give a polynomial time bound for QCL machines. Something that is not known for the classical setting, where in a similar model, catalytic log space, only an expected polynomial-time bound is known. Therefore, we cannot conclude anything yet about the relative power of the quantum and classical catalytic model. We show that certain techniques from the classical setting, are also accessible in the quantum setting. This allows us to show that $\mathsf{TC}_1 \subseteq \mathsf{QCL}$ (logarithmic depth threshold circuits), suggesting that a quantum catalyst gives additional power to the model. We end this chapter by giving a comparison between the *one-clean qubit* model and both quantum and catalytic log-space. We show that in a specific setting, where the quantum model is not allowed to make intermediate measurements $\mathsf{Q_UCL}$ is contained in the one-clean qubit model. Furthermore, we show that CL is contained in the one-clean qubit model.

In Part Three, we discuss a problem central to quantum chemistry, estimating the smallest eigenvalue of a *Local Hamiltonian*. The suggested strategy for finding these eigenvalues consists of the following two steps: First, a *classical* heuristic is used to find a *guiding state* $|\psi\rangle$, a state that has reasonable overlap with the ground-space of the Hamiltonian. Second, the guiding state $|\psi\rangle$ is used as input to Quantum Phase Estimation (definition of this protocol in Lemma 7.8.1) to efficiently and accurately calculate the ground state energy. The second step of this process is characterized by the *Guided Local Hamiltonian* problem. In this problem one is given the description of both a Hamiltonian and a corresponding *semi-classical guiding state* as an input and must find the ground state energy of this Hamiltonian. This problem was first introduced in [GL22] and shown to be BQP-hard for a set of parameters. In Chapter 7 we extend their results showing that the problem remains BQP-hard for a broader set of parameters. Additionally, we extend this problem to the setting of estimating excited state eigenvalues, instead of the ground-state energy.

We build upon this problem in Chapter 8. There we study *Guidable local Hamiltonian problems*, where instead of receiving a classical description of the guiding state as an *input*, it is only *promised* to exist. We show that this problem is QCMA-hard for a broad range of parameters. This hardness result allows us to give complexity theoretic evidence that using a *classical heuristic* to find a

guiding state is just as powerful as using a *quantum heuristic*. Furthermore, we complement our quantum hardness result with a classical containment result (in NP) for a specific range of parameters. This classical containment result allows us to give a restriction, analogous to [AG19], on possible gap amplification procedures required for proving the quantum PCP (probabilistically checkable proofs) conjecture (more details on this in Chapter 8).

## 1.4   List of publications

This thesis is based on the following papers (in chronological order); all authors contributed equally:

[Cad+23a] **Improved Hardness Results for the Guided Local Hamiltonian Problem**
Chris Cade, Marten Folkertsma, Sevag Gharibian, Ryu Hayakawa, François Le Gall, Tomoyuki Morimae, Jordi Weggemans
Contributed talk at International Colloquium on Automata, Languages, and Programming (ICALP), 2023.
Contributed talk at Quantum Information Processing (QIP), 2023.
Chapter 7 is based on this paper.

[WFC23] **Guidable Local Hamiltonian Problems with Implications to Heuristic Ansatz State Preparation and the Quantum PCP Conjecture**
Jordi Weggemans, Marten Folkertsma, Chris Cade
Contributed talk at Theory of Quantum Computation (TQC), 2024. Chapter 8 is based on this paper.

[Buh+24b] **State Preparation by Shallow Circuits Using Feed Forward**
Harry Buhrman, Marten Folkertsma, Bruno Loff, and Niels Neumann
Quantum, vol. 8, page 1552, December 2024.
Chapter 3 is based on this paper.

[Fol+25] **Fully Characterizing Lossy Catalytic Computation**
Marten Folkertsma, Ian Mertz, Florian Speelman, Quinten Tupker
Contributed talk at Innovations in Theoretical Computer Science (ITCS), 2025. Chapter 5 is based on this paper.

[Buh+24a] **Quantum Catalytic Computing**
Harry Buhrman, Marten Folkertsma, Ian Mertz, Florian Speelman, Sergii Strelchuk, Sathya Subramanian, Quinten Tupker
Accepted for a talk at Theory of Quantum Computation (TQC), 2025.
Chapter 6 is based on this paper.

The author additionally co-authored the following papers that are not included in this thesis.

[Cad+23b] **Quantifying Grover Speed-ups Beyond Asymptotic Analysis**
Chris Cade, Marten Folkertsma, Ido Niesen, Jordi Weggemans
Quantum, vol. 7, page 1133, October 2023.
Contributed talk at Quantum Computing Theory in Practice (QCTIP), 2022.

[Cad+24] **Quantum Algorithms for Community Detection and their Empirical Run-times**
Chris Cade, Marten Folkertsma, Ido Niesen, Jordi Weggemans
Quantum Information & Computation, Vol.24, No.5 & 6, pages 361-410, April 2024.

# Chapter 2

# Preliminaries

## 2.1 Notation and terminology

We start by defining some notation that will be used in the thesis. We will denote $\mathbb{N}$ as the set of natural numbers, $\mathbb{R}$ as the set of real numbers and $\mathbb{C}$ as the set of complex numbers. Furthermore we write $[n] := \{0, \ldots n - 1\}$ as the set of numbers 0 up to $n - 1$. We denote $\{0, 1\}^n$ as the set of all n-bit strings. For $x \in \{0, 1\}^n$ we denote $x_i$ as the $i$th bit in $x$ and we write $|x|$ to signify the length of the bit string $x$. For two bit strings $x, y \in \{0, 1\}^n$ we write $x \oplus y$ as the bitwise sum of the two strings, also known as the bitwise XOR between $x$ and $y$.

Often we are interested in the asymptotic behavior of functions. Let $f, g$ be two functions from $\mathbb{N}$ to $\mathbb{N}$, to analyze their respective asymptotic behavior we make use of the big-$\mathcal{O}$ notation which is defined as follows:

$$f(n) \in \mathcal{O}(g(n)) \Leftrightarrow \exists c > 0, N \geq 0 \text{ such that } \forall n > N : f(n) \leq cg(n)$$

Sometimes we require a stronger notion of an upper bound, which is given by little-$o$ notation:

$$f(n) \in o(g(n)) \Leftrightarrow \forall c > 0, \exists N \geq 0 \text{ such that } \forall n > N : f(n) \leq cg(n)$$

Similarly we define $\Omega$ notation as, used for lower bounds:

$$f(n) \in \Omega(g(n)) \Leftrightarrow \exists c, N \geq 0 \text{ such that } \forall n > N : f(n) \geq cg(n)$$

or for stronger lower bounds we can use $\omega$ notation,

$$f(n) \in \omega(g(n)) \Leftrightarrow \forall c > 0, \exists N \geq 0 \text{ such that } \forall n > N : f(n) \geq cg(n).$$

Big-$\mathcal{O}$ and $\Omega$ notation are related as follows:

$$f(n) \in \mathcal{O}(g(n)) \Leftrightarrow g(n) \in \Omega(f(n)).$$

Finally, we denote $f(n) \in \Theta(g(n))$ if both hold, more precisely $f(n) \in \mathcal{O}(g(n))$ and $f(n) \in \Omega(g(n))$. As a variant of $\mathcal{O}$ notation we also use the following notation:

$$f(x) \in poly(x) \Leftrightarrow \exists k \in \mathbb{N}, \text{ such that, } f(x) \in \mathcal{O}(x^k)$$

and

$$f(x) \in polylog(x) \Leftrightarrow \exists k \in \mathbb{N}, \text{ such that, } f(x) \in \mathcal{O}(\log^k(x))$$

### 2.1.1 Dirac notation

In this thesis we make liberal use of bra-ket notation, also known as Dirac notation, which we will shortly describe here. We use what is called a ket to write vectors, for instance we write $|0\rangle$ which should be understood as $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, similarly we write $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. When taking conjugate transpose of these vectors we get: $\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $\langle 1| = \begin{pmatrix} 0 & 1 \end{pmatrix}$ (also known as bra's). In this notation we write the inner product as: $\langle i| \, |j\rangle$ or even simpler $\langle i|j\rangle = \delta_{i,j}$. Where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise (also known as the Kronecker delta). Composition of two vectors is done by taking the tensor product, which is often omitted when writing bra-ket notation:

$$|0\rangle \otimes |0\rangle = |0\rangle |0\rangle = |00\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Given a bit-string $x \in \{0,1\}^n$ we write $|x\rangle$ to signify the state $|x\rangle = \bigotimes_{i=0}^{n-1} |x_i\rangle$. Notation is sometimes slightly abused where for a number $i \in \mathbb{Z}_{\geq 0}$ we can also write $|i\rangle$, which is identified by the computational basis vector indexed by the binary expansion of $i$. It should be clear from the context if the entry of a ket is a number or a bit string. Furthermore, the dimensionality of these vectors should be clear from the context.

The vectors we have written so far, consisted of one entry being 1 and the rest of the entries being 0. We call these types of states *computational basis states*. In the next section we will expand on this notation, introducing its use for writing quantum states.

### 2.1.2 Hermitian matrices

For any linear operator $A \in \mathbb{C}^{m \times n}$, we can make use of Dirac notation and write it as:

$$A = \sum_{i=0, j=0}^{i=m-1, j=n-1} \alpha_{i,j} |i\rangle \langle j| ,$$

where $\alpha_{i,j} \in \mathbb{C}$. For a matrix $A$, we write $A^\dagger$ for its conjugate transpose:

$$A^\dagger = \sum_{i=0,j=0}^{i=m-1,j=n-1} \bar{\alpha}_{ij} \, |j\rangle \, \langle i| \, ,$$

where $\bar{\alpha}_{i,j}$ is the complex conjugate of $\alpha_{i,j}$. A matrix $A$ is called *Hermitian* if $A = A^\dagger$. We write $\lambda_i(A)$ to denote the $i$th eigenvalue of a Hermitian matrix $A$, ordered in non-decreasing order, with $\lambda_0(A)$ denoting the smallest eigenvalue (ground energy). We denote $\mathrm{eig}(A) = \{\lambda_0(A), \ldots, \lambda_{\dim(A)-1}(A)\}$ for the (ordered) set of all eigenvalues of $A$.

### 2.1.3  Norm

Let $d \in \mathbb{N}$, for a vector $v = (v_0, \ldots v_{d-1})^T$, with $v_i \in \mathbb{C}$, we write $\|v\|$ to denote the $l_2$ norm of $v$:

$$\|v\| := \sqrt{\sum_{i=0}^{d-1} |v_i|^2}.$$

For a matrix $M \in \mathbb{C}^{d \times d}$, and $0 < p \leq \infty$ the *Schatten $p$-norm* of $M$ is:

$$\|M\|_p := (\mathrm{Tr}[(\sqrt{M^\dagger M})^p])^{1/p}.$$

There are two particular instances of this norm used in this thesis:

- The Trace-norm (Schatten 1-norm): $\|M\|_1 = \mathrm{Tr}[\sqrt{M^\dagger M}]$

- Operator norm (Schatten $\infty$-norm): $\|M\|_\infty = \sup_{v \in \mathbb{C}^d} \|Mv\|_2$, which is equal to the largest singular value of $M$. This will sometimes be written as $\|M\|$, when the context is clear.

## 2.2  Quantum computation

In this section we give a short overview of the fundamental concepts behind quantum computation; for a more comprehensive explanation we refer the reader to [Wol19], or [NC10].

### 2.2.1  Quantum states

The mathematical description of a quantum state that we use in this thesis is that of a state living in a Hilbert space $\mathcal{H}(N)$. The Hilbert space is a vector space of dimension $N$ which is equipped with an inner product, and for this thesis can be understood as isomorphic to $\mathbb{C}^N$. What we call *pure* quantum states, are vectors in the Hilbert space written as, $|\psi\rangle \in \mathcal{H}(N)$ with norm 1, $\langle\psi|\psi\rangle = 1$.

When $N = 2$ we call these states quantum bits or qubits. A qubit is the fundamental unit of information in quantum computing, and the general state of the qubit can be written as:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle,$$

with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. To create more complex quantum states we can compose different Hilbert spaces, which is done by taking the tensor product. For instance, if we combine two qubits the resulting Hilbert space is $\mathcal{H}(2) \otimes \mathcal{H}(2) = \mathcal{H}(4)$. A general $n$-qubit state therefore lives in the Hilbert space $\mathcal{H}(2)^{\otimes n} = \mathcal{H}(2^n)$, which is isomorphic to $\mathbb{C}^{2^n}$.

Interestingly, by adopting the tensor product as composition of quantum states we can find a fundamental property of quantum states known as *entanglement*. A pure quantum state, $|\psi\rangle \in \mathcal{H}(4)$ is known to be entangled if we cannot write it as a tensor product of local states, i.e. $\nexists |\phi_1\rangle, \nexists |\phi_2\rangle \in \mathcal{H}(2)$ s.t. $|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$. An example of such a state is the well-known EPR (Einstein, Podolsky, Rosen) state,

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

As stated before, general states can be written as $|\psi\rangle \in \mathcal{H}(N)$. We can now give a description of an $n$-qubit quantum state $|\psi\rangle$:

$$|\psi\rangle = \sum_{i \in \{0,1\}^n} \alpha_i |i\rangle,$$

where $\alpha_i \in \mathbb{C}$ are called the amplitudes and they satisfy the constrained $\sum_i |\alpha_i|^2 = 1$. To simplify notation we will sometimes refer to the Hilbert space $\mathcal{H}$ without referring to its dimension, which should be clear from context.

### 2.2.2   Density matrices

In a more general setting, in quantum information, quantum states can be described by *density matrices* $\rho$, which are positive semi-definite operators acting on the Hilbert space $\mathcal{H}$ with trace $\mathrm{Tr}(\rho) = 1$. Density matrices describe *mixed states* which, beyond pure quantum states, can also capture classical uncertainty. In other words, they correspond to classical mixtures of pure quantum states. The density matrix of a pure state is $\rho = |\psi\rangle \langle \psi|$. Given an ensemble of states $\{|\psi_i\rangle\}$ and corresponding probabilities $\{p_i\}$, with $p_i \geq 0$ and $\sum_i p_i = 1$, it can be represented by a mixed state of the form $\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|$. We will denote the set of mixed states a Hilbert space $\mathcal{H}$ by $D(\mathcal{H})$.

### 2.2.3   Quantum gates and unitary matrices

Any $n$-qubit quantum state $|\psi\rangle$ can be transformed into another $n$-qubit $|\phi\rangle$ via *unitary* operations on $\mathcal{H}$. Let $\mathcal{U}(N)$ denote the space of unitary linear operators on $\mathcal{H}(N)$. For any state $|\psi\rangle \in \mathcal{H}(N)$ and unitary matrix $U \in \mathcal{U}(N)$ we have that:

1. $U^\dagger U = I$ where $U^\dagger$ denotes the conjugate transpose of $U$ and $I$ denotes the identity matrix.

2. Let $|\phi\rangle = U |\psi\rangle$ then $||\,|\phi\rangle\,|| = ||U\,|\psi\rangle\,|| = 1$.

3. $U$ is linear, therefore $U |\psi\rangle = \sum_{i=0}^{2^d-1} \alpha_i U |i\rangle$.

Unitary matrices (unitaries for short) acting on 1, 2, or sometimes 3 qubit states are commonly referred to as *quantum gates*, which are the building blocks for quantum circuits (more on quantum circuits in Section 2.2.6). We will start by summarizing a set of important *elementary* gates that will often appear in the thesis. We begin with the *Pauli matrices*. There are four Pauli matrices which all act on a single qubit:

$$X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Their actions on a one-qubit state can be identified by their actions on the computational basis states:

$$X |b\rangle = |b \oplus 1\rangle, \quad Y |b\rangle = -i(-1)^b |b \oplus 1\rangle, \quad Z |b\rangle = (-1)^b |b\rangle, \quad I |b\rangle = |b\rangle,$$

with $b \in \{0,1\}$, these actions extend to all one-qubit quantum states by linearity. Together, the Pauli matrices adhere to a group structure, squaring any of them forms the identity, and multiplying two different Pauli gates creates the third one up to a phase. More concretely:

**2.2.1.** DEFINITION. The Pauli group $P_1$ consists of the three Pauli matrices and the identity, $\{X, Y, Z, I\}$, and additional phase factors $\pm 1$ or $\pm i$. The $n$-qubit Pauli-group $P_n$ is the set of all $4^{n+1}$ possible tensors of length $n$ of matrices from $P_1$, together with a global phase of $\pm 1$ or $\pm i$.

Three other important single-qubit gates are the *Hadamard* gate, the $S$ gate, and $T$ gate:

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad S := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \qquad T := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

The $S$ and the $T$ gates can be understood as taking roots of the $Z$ gate: $S = \sqrt{Z}$ and $T = \sqrt{S}$. The Hadamard gate can be applied to $n$-qubits initialized in the $|0^n\rangle$ state, creates the uniform superposition over all $n$-qubits:

$$H^{\otimes n} |0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle.$$

An important 2-qubit gate is the *Controlled*-NOT gate:

$$\mathsf{CNOT} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

This 2-qubit gate allows interactions between different qubits and can be used to create (or destroy) entanglement. For instance, when combined with the Hadamard gate, it can construct the EPR state:

$$\mathsf{CNOT}(H \otimes I) \ket{00} = \mathsf{CNOT} \frac{1}{\sqrt{2}}(\ket{00} + \ket{10}) = \frac{1}{\sqrt{2}}(\ket{00} + \ket{11}).$$

Another useful 2-qubit gate is the SWAP gate, which exchanges two states between two different registers:

$$\mathsf{SWAP} \ket{\phi} \ket{\psi} = \ket{\psi} \ket{\phi}.$$

Finally, an important 3-qubit gate is the Toffoli gate, which is the reversible quantum equivalent to the AND gate:

$$\mathsf{Toffoli} \ket{x} \ket{y} \ket{b} = \ket{x} \ket{y} \ket{b \oplus (x \cdot y)},$$

where $x, y, b \in \{0, 1\}$.

### 2.2.4 Quantum measurement

Given a quantum state $\ket{\psi} = \sum_{i \in \{0,1\}^n} \alpha_i \ket{i}$, prepared through a sequence of unitary operations, one might wish to extract information from it. However, we cannot directly observe or "see" a quantum state. Instead, we can only obtain classical information by performing a *quantum measurement*.

Quantum measurements are governed by the *Born's rule*, which states that the probability of obtaining a particular outcome is equal to the square of the amplitude associated with that outcome. The most common type of measurement is a measurement in the computational basis. When we perform a computational basis measurement on $\ket{\psi}$, the probability of observing outcome $i$ is given by $|\alpha_i|^2$.

After the measurement, the quantum state *collapses* to a state consistent with the observed outcome. Specifically, if we measure and obtain outcome $i$, the post-measurement state becomes $\ket{i}$. This process allows us to extract $n$ bits of classical information from $\ket{\psi}$, but it comes at the cost of irreversibly destroying the original superposition.

More generally, we can perform a *projective measurement*. A projective measurement is described by a set of mutually orthogonal projectors $\{P_0, \dots P_m\}$,

which sum up to identity $\sum_{i=0}^{m} P_i = I$. Given a state $|\psi\rangle$, the probability of observing outcome $i$ is then given by:

$$||P_i |\psi\rangle ||^2 = \langle\psi| P_i |\psi\rangle$$

after which the state collapses to

$$\frac{P_i |\psi\rangle}{||P_i |\psi\rangle ||}.$$

Note that all the possible post-measurement states are mutually orthogonal, due to the projectors being mutually orthogonal.

In principle, it is not necessary to measure an entire $n$-qubit state. Instead one can opt for measuring only a subset of the qubits. For instance given a quantum state

$$\alpha |0\rangle |\phi_0\rangle + \beta |1\rangle |\phi_1\rangle ,$$

where $\langle\phi_0|\phi_1\rangle = 0$, one can only measure the first qubit. In the language of projective measurements, this can be understood as the projectors: $\{|0\rangle \langle 0| \otimes I, |1\rangle \langle 1| \otimes I\}$. Now by following the rules described above, this measurement gives outcome 0 with probability $|\alpha|^2$ and if the outcome is 0, the state collapses to

$$|0\rangle |\phi_0\rangle ,$$

thereby projecting the latter $n-1$ qubits into the state $|\phi_0\rangle$. Using measurements for projecting onto certain quantum states can be a very powerful tool, which will be extensively studied in Chapter 3.

### 2.2.5 Quantum channels

More general quantum state transformations that combine unitaries and measurements are *quantum channels*.

**2.2.2. Definition** (Quantum channels). A *quantum channel* is a linear operator that maps density matrices to density matrices, $\Phi : D(\mathcal{H}_1) \to D(\mathcal{H}_2)$ (also known as super operators or completely positive trace preserving (CPTP) maps). It is also required to have two additional properties: 1) it must be completely positive; and 2) it must be trace preserving. We denote the set of channels from $D(\mathcal{H})$ to itself by $\mathcal{C}(D(\mathcal{H}))$.

### 2.2.6 Quantum circuits

Similarly to classical circuits – which are created from AND, OR and NOT gates acting on input bits – we can combine elementary quantum gates, as described above, to create more complex quantum unitaries. We call these combinations of elementary gates *quantum circuits*. The set of elementary gates we use to construct a quantum circuit is called the gate set.

**2.2.3.** Definition. A *quantum circuit* is a sequence of gates from a fixed gate set $\mathcal{G}$. The *size* of the quantum circuit is the total number of gates. Sequential quantum gates acting on different qubits are considered to be applied in parallel on the same layer, the total number of layers of a circuit is the circuits *depth*, the total number of qubits on which the circuit acts is its *width*.

Unlike in the classical setting where any Boolean function can be computed by a circuit consisting of AND and OR gates, a similar type of universality is not possible in the quantum setting. This has to do with the fact that the unitary group is a continuous group, therefore, it is not possible to construct all unitaries using a finite gate set. Some gate sets allow for approximating all unitaries up to arbitrary precision, we call such a gate sets *universal*. A famous example of such a gate set is:

$$\{H, T, \mathsf{CNOT}\}.$$

By the Solovay-Kitaev theorem, the overhead of approximating any 1-qubit unitary using such a finite gate set is at most $O(\log^c(1/\epsilon))$ for some constant $c$[DN06].

In chapter 3 we study a computational model with severe restriction on the depth of a quantum circuit. This requires a more complete gate set to be used; we refer in this chapter to a continuous gate set, which is:

$$\mathcal{G}_{\mathsf{PC}} = \{X(\theta), Z(\theta'), \mathsf{CNOT} | \theta, \theta' \in \mathsf{PC} \cap [0, 2\pi)\} \tag{2.1}$$

where $X(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$, $Z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$, and $\mathsf{PC}$ is the set of polynomial time computable numbers [NO09]. Note that this gate set is not actually continuous due to the requirement that the precision of $\theta$ and $\theta'$ are bounded by being polynomially time computable.

An example of a famous gate set that is not universal is a set of gates from the *Clifford group*. Circuits constructed from the Clifford group form a well-known group of quantum circuits that *stabilize* the Pauli group.

**2.2.4.** Definition. The Clifford group $C_n$ consists of all $n$-qubit unitaries that leave the Pauli group $P_n$ invariant under conjugation. That is, let $c \in C_n$ be any Clifford circuit, then for any $P \in P_n$, there exists a $P' \in P_n$, such that $cP = P'c$.

The Clifford group is generated by quantum circuits constructed from the gate set:

$$\{\mathsf{CNOT}, H, S\}$$

Any circuit constructed using only these three gates is called a Clifford circuit.

Unsurprisingly, Clifford circuits only cover a small part of the possible quantum circuits. Moreover, on a linear nearest-neighbor architecture, $\mathcal{O}(n)$ deep Clifford circuits suffice to simulate any Clifford unitary of size $2^n \times 2^n$ [MR18]. The actions of Clifford circuits can be efficiently [Got98] simulated by classical

computers. Universal quantum computations require additional gates, although almost any quantum gate suffices. For example, adding the single qubit $T$-gate, $T : |x\rangle \mapsto e^{i\pi x/4} |x\rangle$, to the Clifford group gives a universal gate set.

## 2.3 Computational complexity theory

In computational complexity theory one is interested in categorizing the difficulty of solving a problem based on a specific measure. Most often this measure is time, typically measured in number of steps that is required to solve a problem. The type of problems that one discusses are traditionally decision problems: A computational device is given an input, in the form of a bit string $x \in \{0,1\}^*$, and has to decide if the bit string has a certain property or not. This property can mathematically be specified as a boolean function $f : \{0,1\}^* \to \{0,1\}$, which assigns a value 1 or 0 to every bit string $x$ signifying that $x$ has, or does not have this property. The set of strings for which $f(x) = 1$ is called a *language*: $L := \{x \in \{0,1\}^* | f(x) = 1\}$. The task of the computational machine is to output $f(x)$ given input $x$.

Sometimes we are interested in what is known as a *promise problem*. We call a promise problem $A = (A_{yes}, A_{no})$ a tuple of non-intersecting sets, where $A_{yes}, A_{no} \subseteq \{0,1\}^*$ ($A_{yes} \cap A_{no} = \emptyset$). Given an $x \in \{0,1\}^*$ the machine has to output YES if $x \in A_{yes}$ and NO if $x \in A_{no}$. If $x$ is not in $A_{yes}$ or in $A_{no}$ the output of the machine is not specified and can be anything. Note that this reduces to the regular definition of deciding on languages if $A_{yes} \cup A_{no} = \{0,1\}^*$. Any class defined on languages can also be defined using promise problems. Note that classes defined on languages do not have to be equivalent to the same class defined on promise problems. In Chapters 7 and 8 we will make use of classes defined on promise problems instead of languages.

### 2.3.1 Turing machine

The abstract computational model that is often used to analyze the complexity of decision problems is called a *Turing machine*. Here we will give an informal description of a Turing machine; however, we would like to refer the more interested reader to [AB09] for a more concrete and in-depth definition. A Turing machine is a small finite state machine that is equipped with an *input tape* containing the input bit string, with the restriction that it can only be read from, a write only *output tape* where the machine can write the final answer of a computation, and a read-write *work tape*, which the machine can use for intermediate calculations. Each tape is accessed by the machine through a *tape head*, which can read and/or write one bit at the time. The machine itself has a finite set of *states*, denoted by $Q$. The machine contains an internal "register" that can hold one element of $Q$, determining the current state of the machine. This state of the machine deter-

mines the action that the machine will take in the next step of the computation. A step consists of, (1) reading the bit at every location of the read tape heads, (2) writing a bit at the location of a write tape head, (3) update the internal register determining the state of the machine and (4) move the tape heads left, or right. There are two special states in $Q$, the start state, in which the machine is initialized for computation, and the halt state, which signifies the end of the computation.

There are two measures of complexity of Turing machines relevant to this thesis. The first is *time*, which is the number of steps that the Turing machine takes before it halts. The second is *space*, which is the maximum number of entries of the work tape in use by the Turing machine at any one time-step during the computation. When deciding if a bit string is contained in a language, we say that a Turing machine *accepts* the input if at the end of the computation the Turing machine wrote a 1 on the output tape, and it *rejects* the string if it wrote a 0 on the output tape. Furthermore, we say that a Turing machine $M$ runs in polynomial time, if there exist constants $\exists c > 0, \exists k \in \mathbb{N}$, such that $\forall x \in \{0,1\}^*$, $M$ halts within time $c|x|^k + c$ (where $|x|$ denotes the length of bit string $x$). Similarly, a Turing machine $M$ is said to *run in logarithmic space* if there exists a constant $\exists c > 0$, such that for every input $x \in \{0,1\}^*$, at every step of the computation $M$ uses at most $c \log |x| + c$ cells on its work tape.

## 2.3.2 Classical complexity classes

The goal within complexity theory is to order the difficulty of problems into sets, we call these sets *complexity classes*. The most famous of those classes is $\mathsf{P}$, problems decidable in polynomial time, also known as problems that have an efficient solution. The definition of $\mathsf{P}$ is as follows:

**2.3.1.** DEFINITION ($\mathsf{P}$). A language $L \subseteq \{0,1\}^*$ is in $\mathsf{P}$ (polynomial time) if and only if there exists a deterministic polynomial-time Turing machine $M$ such that for every input $x \in \{0,1\}^*$,

- if $x \in L$ then $M$ accepts $x$.

- if $x \notin L$ then $M$ rejects $x$.

One way to extend a complexity class is to equip the Turing machine with an additional resource. A natural choice for such a resource is *randomness*, resulting in what is known as a *probabilistic* Turing machine. This modification also necessitates a slight change to the acceptance criterion: The machine must accept with sufficiently high probability.

**2.3.2.** DEFINITION ($\mathsf{BPP}$). A language $L \subseteq \{0,1\}^*$ is in $\mathsf{BPP}$ (bounded probabilistic polynomial time) if and only if there exists a *probabilistic* polynomial-time Turing machine $M$, such that for every input $x \in \{0,1\}^*$,

- if $x \in L$ then $M$ accepts $x$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then $M$ accepts $x$ with probability at most $\frac{1}{3}$.

The choice of the acceptance probability, such as $\frac{2}{3}$, is somewhat arbitrary. Changing this threshold does not significantly affect the power of the model, as long as it remains at least polynomially separated from a $\frac{1}{2}$. This is because BPP supports *error reduction*: by running the same computation multiple times and taking the majority vote of the outcomes, one can amplify the success probability.

Another important resource that can be given to a Turing machine is *non-determinism*. A non-deterministic Turing machine is granted access to an additional tape containing a *witness*—a proposed solution to help determine whether an input string $x$ belongs to a language. If $x$ is in the language, there exists at least one witness that causes the machine to accept. However, if $x$ is not in the language, then the machine must reject regardless of the witness provided.
This captures the idea of problems whose solutions are *efficiently verifiable*: given a proposed solution, one can efficiently check if it is correct. An intuitive example of a problem in this complexity class is determining whether a given Sudoku puzzle has a solution. While verifying a provided solution is easy, actually finding one may be much harder.

**2.3.3.** Definition (NP). A language $L \subseteq \{0,1\}^*$ is in NP (non-deterministic polynomial time) if and only if there exists a deterministic polynomial-time Turing machine $M$ and a polynomial $p$, where $M$ takes as input a string $x \in \{0,1\}^*$ and a $p(|x|)$-bit witness $y$, such that for every $x \in \{0,1\}^*$,

- if $x \in L$ then there exists a witness $y \in \{0,1\}^{p(|x|)}$ such that $M$ accepts $(x,y)$.

- if $x \notin L$ then for every witness $y \in \{0,1\}^{p(|x|)}$ we have that $M$ rejects $(x,y)$.

### 2.3.3 Quantum complexity classes

Instead of relying on quantum Turing machines, which serve as the natural quantum analogue of classical Turing machines, we define quantum complexity classes in terms of the quantum circuit model. Defining complexity classes with respect to the circuit model instead of the quantum Turing machine model does not change the power of the class as it has been shown that these two models are equivalent [NO09]. Using the circuit model requires a specific *uniformity* condition, we will use the uniformity condition as provided in [Wol19].

A family of quantum circuits is a set of quantum circuits $\{V_n\}$, one for each $n$. Each quantum circuit has one dedicated output qubit, which on measurement in the computational basis produces the outcome of a calculation. A circuit family recognizes a language $L \subseteq \{0,1\}^*$, if for every $n$ and every input $x \in \{0,1\}^n$

on input $x$ the circuit $V_n$ outputs 1 if $x \in L$ and 0 if $x \notin L$. If $V_n$ outputs 1, we say that $V_n$ accepts input $x$, and if $V_n$ outputs 0, we say it rejects input $x$. Note that this acceptance criterion can also be modified to output 1 with a certain probability. Furthermore we call a circuit family *polynomial-time uniform* if there exists a deterministic polynomial time Turing machine that on input $n$ outputs a description of $V_n$.

The most famous quantum complexity class is BQP, *bounded quantum polynomial time* computations:

**2.3.4.** DEFINITION (BQP). A language $L \subseteq \{0,1\}^*$ is in BQP (bounded quantum polynomial time) if and only if there exists a polynomial-time uniform family $\{V_n\}$ of polynomial size quantum circuits, such that for every $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $V_n$ accepts $x$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then $V_n$ accepts $x$ with probability at most $\frac{1}{3}$.

There also exists a quantum complexity class more equivalent to P, which requires perfect acceptance and rejection probability. This class is called *exact quantum polynomial time* EQP, however, this class is not very natural and widely used because its definition relies on the choice of gate set. As discussed in Section 2.2.6, given a fixed finite gate set it is not possible to construct an exact quantum circuit for every unitary operator. In fact, most unitaries can only be implemented approximately. Such approximations introduce a small error with respect to the target unitary. However, any circuit that decides a language in EQP must produce the correct outcome with probability exactly 1, which rules out the use of approximations in EQP.

**2.3.5.** DEFINITION (EQP). Given a gate set G, a language $L \subseteq \{0,1\}^*$ is in EQP (exact quantum polynomial time) if there exists a polynomial-time uniform family $\{V_n\}$ of polynomial size quantum circuits (constructed from G), such that for every $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $V_n$ accepts $x$ with probability 1.

- if $x \notin L$ then $V_n$ rejects $x$ with probability 1.

Similar to the classical case, quantum circuits can be enhanced by adding additional power such as non-determinism. In the quantum setting there are two distinct forms of non-determinism that we are interested in. We can have witnesses more equivalent to the classical setting, where a witness consists of some polynomially sized bit string. We can also have quantum witnesses, where a witness consists of a quantum state on a polynomial number of qubits. This distinction gives two separate complexity classes called *Quantum Merlin Arthur* and *Quantum Classical Merlin Arthur*:

**2.3.6.** DEFINITION (QMA). A language $L \subseteq \{0,1\}^*$ is in QMA (Quantum Merlin Arthur) if and only if there exists a polynomial-time uniform family $\{V_n\}$ of polynomial size quantum circuits and a polynomial $p$, where $V_n$ takes as input a string $x \in \{0,1\}^*$ with $|x| = n$, and a $p(n)$-qubit witness quantum state, such that for every $x \in \{0,1\}^*$,

- if $x \in L$ then there exists a witness state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes p(n)}$ such that $V_n$ accepts $(x, |\psi\rangle)$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then for every witness state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes p(n)}$, $V_n$ accepts $(x, |\psi\rangle)$ with probability at most $\frac{1}{3}$.

**2.3.7.** DEFINITION (QCMA). A language $L \subseteq \{0,1\}^*$ is in QCMA (Quantum Classical Merlin Arthur) if and only if there exists a polynomial-time uniform family $\{V_n\}$ of polynomial size quantum circuits and a polynomial $p$, where $V_n$ takes as input a string $x \in \{0,1\}^*$ with $|x| = n$, and a $p(n)$-bit classical witness $y \in \{0,1\}^{p(n)}$ written as $|y\rangle$, such that for every $x \in \{0,1\}^*$,

- if $x \in L$ then there exists a witness $y \in \{0,1\}^{p(n)}$ such that on input $V_n$ accepts $(x, |y\rangle)$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then for every witness $y \in \{0,1\}^{p(n)}$, $V_n$ accepts $(x, |y\rangle)$ with probability at most $\frac{1}{3}$.

Note that in these proof systems, we can again allow the verifier to accept with probability other than $\frac{2}{3}$ and $\frac{1}{3}$. We sometimes refer to these classes as QMA$[c, s]$ and QCMA$[c, s]$, where $c$ and $s$ mean the following: We say that in the YES case $(x \in L)$ the acceptance probability is at least $c$, where $c$ is called the *completeness*. Similarly, in the NO case $(x \notin L)$ the acceptance probability is at most $s$, where $s$ is called the *soundness*. For any class with completeness $c$ and soundness $s$ such that $c - s \geq 1/\mathrm{poly}(n)$, standard error reduction allows us to amplify the gap and reduce to the canonical values $c = \frac{2}{3}$ and $s = \frac{1}{3}$.

It is still an open question whether QMA and QCMA are equal, or if QMA is strictly larger than QCMA. There are oracle separations, which suggest that QMA might be strictly larger than QCMA[AK07; Zha24].

### 2.3.4 Complexity of space

Additionally, we care about space bounded complexity, here a Turing machine is not bounded by time, but instead in the space that it is allowed to use. A natural space bounded class is *log-space*:

**2.3.8.** DEFINITION (L). A language $L \subseteq \{0,1\}^*$ is in L (log-space) if there exists a Turing machine $M$ using at most $r = O(\log(n))$ space such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $M$ accepts $x$.

- if $x \notin L$ then $M$ rejects $x$.

Analogous to the time-bounded setting, there exists a quantum version of the log-space complexity class known as *bounded-error quantum log-space*. We define this class using the quantum circuit model; however, doing so requires careful consideration of the uniformity condition.

In quantum log-space, we consider quantum circuits that act on $O(\log n)$ qubits. This poses a challenge, as there is not enough space to explicitly store the input string $x \in \{0,1\}^n$ within the circuit. One way to circumvent this is to allow read-only access to an input register containing $x$. However, in our approach, we resolve this by modifying the uniformity condition.

We say that a family of bounded-width quantum circuits $\{V_x\}$ is *log-space uniform* if there exists a deterministic log-space Turing machine that, on input $x$, outputs a description of the circuit $V_x$[1].

**2.3.9.** DEFINITION (BQL). A language $L \subseteq \{0,1\}^*$ is in BQL (bounded quantum log-space) if there exists a log-space uniform family of quantum circuits $\{V_x\}$ of width $w = O(\log(n))$ such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $V_x$ accepts $x$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then $V_x$ accepts $x$ with probability at most $\frac{1}{3}$.

### 2.3.5 Circuit classes

Lastly, we will require the definition of low-depth classical and quantum circuit classes. We will first recall the definition of a classical circuit from [Wol19].

A *Boolean circuit* is a finite directed acyclic graph whose internal nodes are AND, OR and NOT gates. It has $n$-input nodes, which contain the $n$-bits of the input and there are one or more output nodes. Each internal node applies its gate to the values on its incoming edges. At the end of the computation the output nodes assume some value. We say that a circuit computes a Boolean function $f : \{0,1\}^n \to \{0,1\}^m$, if the output nodes contain the correct value $f(x)$ for every input $x$. The *size* of the circuit is the number of gates in the circuit, and its *depth* is the maximal length of a path from an input node to an output node. The number of inputs to a node, also called fan-in, can be bounded or unbounded.

---

[1]Our definition of log-space (Definition 2.3.8) is defined with respect to decision problems. Therefore, the output of a log-space calculation, as defined, is only one bit and not a full description of a circuit. One can easily extent this by equipping the log-space Turing machine with a write only output tape, on which the full description can be written.

**2.3.10.** DEFINITION ($\mathsf{NC}^k$). For every $k$, a language $L \subseteq \{0,1\}^*$ is in $\mathsf{NC}^k$ if and only if there exists a family of circuits $\{C_n\}$ where $C_n$ has $poly(n)$ size, depth $\mathcal{O}((\log n)^k)$, and consists of **bounded**-fan-in AND- and OR-gates, such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $C_n$ accepts $x$.

- if $x \notin L$ then $C_n$ rejects $x$.

**2.3.11.** DEFINITION ($\mathsf{AC}^k$). For every $k$, a language $L \subseteq \{0,1\}^*$ is in $\mathsf{AC}^k$ if and only if there exists a family of circuits $\{C_n\}$ where $C_n$ has $poly(n)$ size, depth $\mathcal{O}((\log n)^k)$, and consists of **unbounded**-fan-in AND- and OR-gates, such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $C_n$ accepts $x$.

- if $x \notin L$ then $C_n$ rejects $x$.

**2.3.12.** DEFINITION ($\mathsf{TC}^k$). For every $k$, a language $L \subseteq \{0,1\}^*$ is in $\mathsf{TC}^k$ if and only if there exists a family of circuits $\{C_n\}$ where $C_n$ has $poly(n)$ size, depth $\mathcal{O}((\log n)^k)$, and consisting of unbounded-fan-in AND-, OR- and Threshold$_t$-gates, where a Threshold$_t$-gate evaluates to one if and only if the sum of the inputs is at least $t$, such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $C_n$ accepts $x$.

- if $x \notin L$ then $C_n$ rejects $x$.

These classes also have a quantum equivalent class.

**2.3.13.** DEFINITION. For every $k$, a language $L \subseteq \{0,1\}^*$ is in $\mathsf{QNC}^k$ if and only if there exists a family of quantum circuits $\{V_n\}$ where $V_n$ has $poly(n)$ size, $\mathcal{O}((\log n)^k)$ depth, and consists of single- and two-qubit quantum gates, such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $V_n$ accepts $x$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then $V_n$ accepts $x$ with probability at most $\frac{1}{3}$.

Definitions for the quantum versions of $\mathsf{AC}^k$ and $\mathsf{TC}^k$ also exist (which will not be required for this thesis). However, when equipping the class $\mathsf{QNC}^k$ with unbounded-fan-in parity gates, all three classes intersect [Gre+02; Moo99; TT13]. The final class is the class of polynomial depth circuits:

**2.3.14.** DEFINITION ($\mathsf{Qpoly}$). A language $L \subseteq \{0,1\}^*$ is in $\mathsf{Qpoly}$ (Quantum polynomial time) if and only if there exists a family of quantum circuits $\{V_n\}$, where $V_n$ has $poly(n)$ depth, $poly(n)$ size, and consists of single- and two-qubit quantum gates from $\mathcal{G}_{\mathbf{PC}}$, such that for every input $x \in \{0,1\}^*$, with $|x| = n$,

- if $x \in L$ then $V_n$ accepts $x$ with probability at least $\frac{2}{3}$.

- if $x \notin L$ then $V_n$ accepts $x$ with probability at most $\frac{1}{3}$.

# Part One
# Measurements and fast intermediate classical calculations

# Chapter 3
## Local Alternating Quantum-Classical Computation

A major problem in the use of current quantum hardware is that they are unable to carry out universal quantum computations due to the 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 [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.

In this chapter we will introduce the first resource that can help boost the power of near-term quantum computations. We take inspiration from the suggested four step process of fault-tolerant quantum computing, not to carry out error corrected calculations, but to directly enhance short depth quantum calculations. We show that this can be used to drastically reduce the depth of certain quantum circuits. To formalize this study, we introduce a new computational model: *Local Alternating Quantum-Classical Computations*, or LAQCC for short.

## 3.1 The four step process

For fault-tolerant quantum computing, we repeat four steps:

1. We apply a number of single and two-qubit quantum gates, in parallel whenever possible;

2. We perform a syndrome measurement on a subset of the qubits;

3. We perform fast classical computations to determine which errors have occurred and how to correct them;

4. We apply correction terms based on the classical computations.

We then repeat these four steps with a next sequence of gates. These four steps are essential to fault-tolerant quantum computing.

The starting point of this chapter 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-classical 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 classical 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, reducing the actual clock time of the calculation.

## 3.2   The LAQCC model

Our first contribution is to formalize the computational power of low-depth quantum computations leveraging the four-step process. Therefore, we introduce a new computational model, called *Local Alternating Quantum-Classical Computations* (LAQCC). In this model we alternate between running quantum circuits (constrained by locality), ending in the measurement of a subset of qubits (step 2), and fast classical computations based on the measurement results (step 3). The outcome of the classical computations is then used to control future quantum circuits (step 4). We allow for flexibility in this model by giving different constraints to the power of both the quantum circuits and the classical circuits as well as the

number of alternations between them. Most attention will be given to LAQCC containing quantum circuits of constant depth, classical 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 ourselves to logarithmic depth classical computations, as this is the first natural and non-trivial extension beyond constant-depth classical computations, for measurement-based models. Constant-depth classical computations do however also have an equivalent constant-depth quantum implementation.

We define the computational model *Local Alternating Quantum-Classical Computations* (LAQCC) as follows:

**3.2.1.** DEFINITION (Local Alternating Quantum-Classical Computations). Let 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;

- After every quantum circuit $Q$ a subset of the qubits is measured;

- Every classical layer implements a classical circuit $C \in \mathcal{C}$;

- The classical circuit receives input from the measurement outcomes of previous quantum layers;

- The classical circuit can control quantum operations in future layers;

- There are $d$ alternating layers of quantum and classical circuits.

The allowed gates in the quantum and classical layers are given by $\mathcal{Q}$ and $\mathcal{C}$ respectively. Furthermore, we require a circuit in LAQCC($\mathcal{Q}, \mathcal{C}, d$) to deterministically prepare a pure state on the all-zero initial state.

The grid topology imposed on the quantum operations implies that qubits can only interact with their direct neighbors on the grid. A circuit in LAQCC($\mathcal{Q}, \mathcal{C}, d$) can use the results of the classical intermediate layers and control quantum operations in future layers. In a sense, information is fed forward in the circuit. Note, as classical computations are in general significantly faster than the quantum operations, we only count quantum operations towards the depth of the circuit, unless specified otherwise. A graphic illustration of a general LAQCC circuit can be found in Figure 3.1.

**3.2.2.** REMARK. There exists ambiguity in the choices for $\mathcal{Q}$, $\mathcal{C}$ and $d$. For example, we have LAQCC(QPoly($n$), P, $\mathcal{O}(1)$) = LAQCC(QNC$^0$, P, $\mathcal{O}(\text{poly}(n))$). This follows as any P-circuit is in QPoly($n$), and we can concatenate poly($n$) constant-depth quantum circuits with trivial intermediate classical computations.

First classical circuit
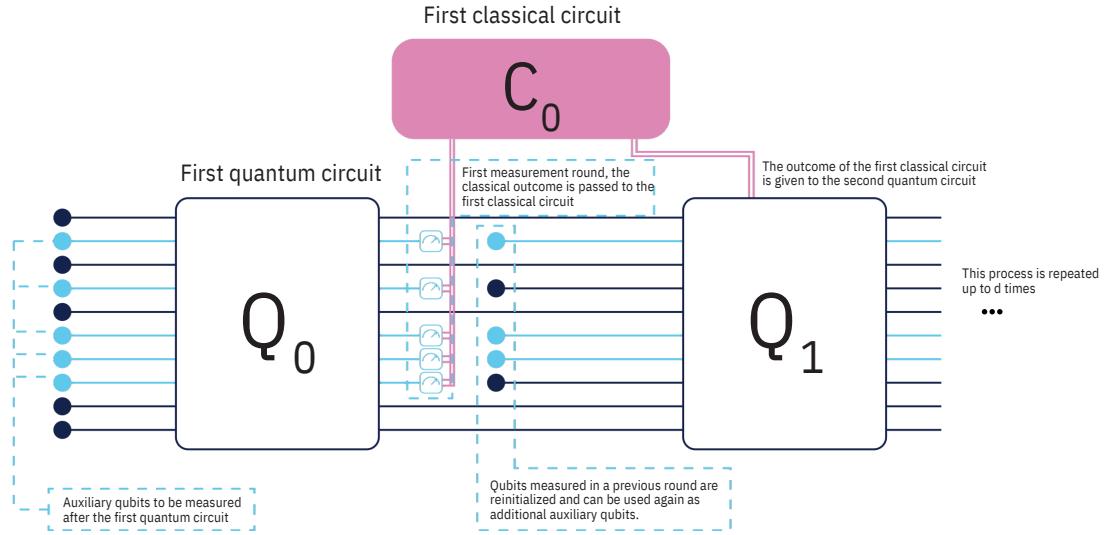


Figure 3.1: Illustration of a general LAQCC circuit.

This ambiguity can be non-trivial: consider for instance

$$\mathsf{LAQCC}(\mathsf{QNC}^1, \mathsf{NC}^1, \mathcal{O}(1)) \subseteq \mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(\log(n))).$$

The inclusion from left to right follows immediately by the same argument as above. It is however not obvious if the logarithmic number of measurement rounds, allowed in the right hand side, can be simulated by a $\mathsf{QNC}^1$ circuit. Even stronger, in Section 3.3.2 we will show that threshold gates are available in $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(1))$. From this fact it follows immediately that any $\mathsf{TC}^1$-circuit is contained in $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(\log(n)))$ . It is unclear these circuits are also contained in $\mathsf{LAQCC}(\mathsf{QNC}^1, \mathsf{NC}^1, \mathcal{O}(1))$[1].

In the remainder of this work, we consider a specific instantiation of $\mathsf{LAQCC}(\mathcal{Q}, \mathcal{C}, d)$.

**3.2.3.** NOTATION. *We let* LAQCC *refer to the instance* $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(1))$, *together with a grid nearest-neighbor topology and the quantum gate-set* $\mathcal{G}_{\mathbf{PC}}$ *from Equation 2.1. The classical computations are bounded to polynomial size, logarithmic depth, and of bounded-fan-in.*

The class $\mathsf{NC}^1$ is a natural non-trivial class beyond constant-depth complexity classes. As the depth of these circuits remains low, they can be implemented quickly. In this work, we assume qubits do not decohere during the classical computation. Incorporating the errors throughout the whole LAQCC-computation,

---

[1]This is not super surprising, if you completely forget about the quantum operations you can see that the model that allows for $d = \mathcal{O}(\log(n))$ repetitions contains classical circuits of $\mathcal{O}((\log n)^2)$ depth, which can clearly simulate $\mathsf{TC}^1$ circuits. However, the model with a constant number of repetitions does not.

including the classical intermediate computations, proves an interesting direction for future research.

### 3.2.1 Similar models

There have been many results in the last 25 years using the four-step process in a similar way as in the LAQCC model. The first and most famous of them can be found in the paradigm of measurement-based quantum computing [GC99; RB01; Joz06; CJL08]: A universal form of quantum computing where a quantum state is prepared and operations are performed by measuring qubits in different bases, depending on previous measurements and intermediate measurements.

Pham and Svore were the first to formalize the four-step protocol 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, including specifying costs in the width, number of qubits, depth, number of concurrent time steps, size, and total number of non-Identity operations. As a result, they find 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 fan-out gate, can also be replaced by additional log-depth classical 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 circuits [PSC21]. Similarly to the four-step scheme we just described, they allow for a short depth circuit to be run on the qubits, 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 transitional 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 [TVV23a; Tan+24]. The definition of LAQCC sharpens the original definition of Pham and Svore by adding constraints to the intermediate classical computations.

The main result of Piroli, Styliaris, and Cirac, that 1D translational invariant MPS with fixed bond dimension can be prepared by LOCC-assisted circuits, relies on local symmetries of the MPS. These symmetries allow them to prepare local states (on a constant number of qubits) and glue them together by doing one round of the appropriate entangling measurement and corrections, after which they run a round of local unitaries to get the desired result. This general scheme for preparing states that exhibit an MPS description with the appropriate local symmetries requires only geometrically local unitaries and one round of measurement and corrections and therefore is accessible in LAQCC. The search for measurement-based constant depth circuits for states such as Symmetry Protected Topological (SPT) phases is a broad ongoing field of research [TVV23a;

Tan+24; Smi+23]. All these schemes have a LAQCC implementation. Note however that Piroli, Styliaris, and Cirac also suggest a circuit for the $W$-state. This circuit uses sequentially and dependent measurement-based corrections of the auxiliary qubits. These dependent measurements translate to sequential alternations between the quantum and classical circuits and therefore increase the total depth to linear depth, exceeding the constant-depth constraints imposed by LAQCC-circuits.

## 3.3 LAQCC subroutines

Our first investigation of the LAQCC model is to figure out a list of useful gates and subroutines available to LAQCC circuits. The main workhorse behind most of these routines is the ability to parallelize Clifford circuits using quantum teleportation. We will start out giving an explanation of this in the following section, after which we will give a list of useful gates and subroutines.

### 3.3.1 Clifford circuits

The concept of intermediate measurements with subsequent computations is closely related to measurement-based quantum computing. A famous result from measurement-based quantum computing is that all Clifford circuits can be parallelized using measurements. In this section we borrow techniques from this result to show that any Clifford circuit has an 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, or ebit, state, up to local Pauli gates. This projection combined with an ebit state teleports a quantum state between qubits. After teleportation, one needs to correct the local Pauli gate created by the Bell measurement. A similar process can be used to apply quantum gates. However, the Pauli gates that arise during teleportation have to be corrected before the calculations can proceed, which necessitates subsequent adaptive operations.

With Clifford circuits, these subsequent operations can be omitted. Clifford circuits stabilize the Pauli group, allowing for simultaneous measurements and hence parallelization of the entire Clifford circuit [Joz06]. Consider a simple example of teleporting a single-qubit quantum state. 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.

By using one Bell-basis measurement, we can apply two sequential Clifford

gates $U_1$ and $U_2$ on a quantum state $|\psi\rangle$, which gives:

$$\sum_{i,j\in\{0,1\}} \left[ (\langle ii| \, (P^{a,b} \otimes I) \otimes I \right] U_1 \otimes I \otimes U_2 \, |\psi\rangle \, |jj\rangle = \sum_{i,j\in\{0,1\}} \langle i| \, P^{a,b} U_1 \, |\psi\rangle \, \langle i|j\rangle \, U_2 \, |j\rangle$$

$$= \sum_{i\in\{0,1\}} U_2 \, |i\rangle \, \langle i| \, P^{a,b} U_1 \, |\psi\rangle$$

$$= U_2 P^{a,b} U_1 \, |\psi\rangle \, .$$

Note that besides projecting on a Bell state, an initial entangled Bell-state is required. $U_2$ is a Clifford gate, hence there exists a $P^{\hat{a},\hat{b}}$ such that $U_2 P^{a,b} U_1 \, |\psi\rangle = P^{\hat{a},\hat{b}} U_2 U_1 \, |\psi\rangle$, allowing the correction term to be pushed to the end of the circuit. Repeating the same argument for multiple Clifford unitaries gives the quantum state $\ldots P_2^{a_2,b_2} U_2 P_1^{a_1,b_1} U_1 \, |\psi\rangle$. Due to the conjugation relation of the Clifford and Pauli gates, all correction terms can be postponed to the end of the computation.

**Clifford-ladder circuit**

A similar argument holds when looking at *Clifford-ladder circuits.*

**3.3.1.** DEFINITION (Clifford-ladder circuit). Let $\{U^i\}_{i=0}^{n-2}$ be a collection of $n-1$ 2-qubit Clifford unitaries. There are two possible types of Clifford-ladder circuits, a right and a left Clifford-ladder: A right Clifford-ladder circuit $C_{ladder}$ is a circuit of depth $\mathcal{O}(n)$ and width $\mathcal{O}(n)$ of the following form:

$$C_{ladder} = \prod_{i=0}^{n-2} U_{i,i+1}^{(i)}$$

where $U_{i,i+1}^{(i)}$ denotes that unitary $U^{(i)}$ is applied on qubits $i$ and $i+1$. Similarly a left Clifford-ladder circuit is a circuit of the form:

$$C_{ladder} = \prod_{i=0}^{n-2} U_{n-(i+2),n-(i+1)}^{(i)}$$

Both right and left Clifford are considered a Clifford ladder circuit

**3.3.2.** REMARK. Note that a ladder circuit does not have to start at either the first or the last qubit, as the 2-qubit gate $I \otimes I$ is also a Clifford gate, furthermore, note that each 2-qubit Clifford unitary $U^{(i)}$ itself is of constant-depth.

The next lemma shows that any Clifford-ladder circuit has an equivalent LAQCC circuit. Figure 3.2 shows this mapping graphically. Each two-qubit unitary is parallelized using gate teleportation and with the Clifford commutation relations, the Pauli correction terms are pushed to the end of the computation.

**3.3.3.** LEMMA. *Any Clifford-ladder circuit has an* LAQCC *implementation of depth $\mathcal{O}(1)$ and width $\mathcal{O}(n)$.*
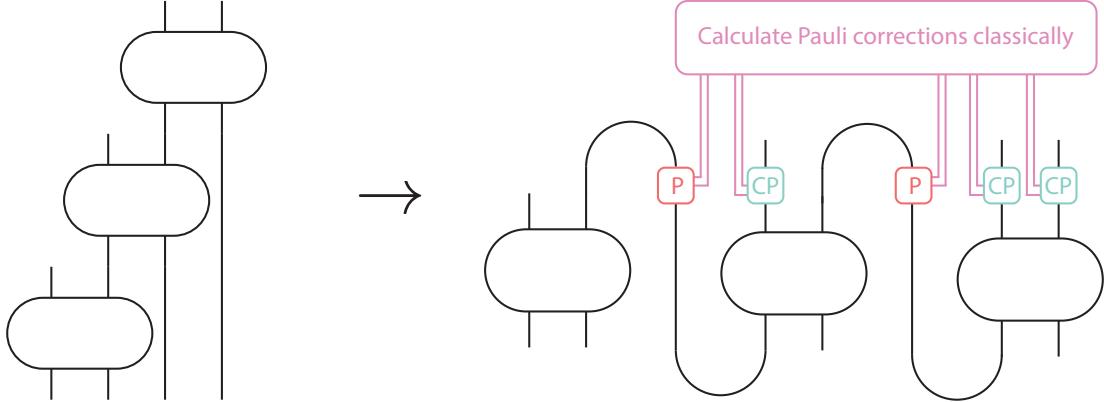
Figure 3.2: Graphical representation of Clifford-ladder circuit parallelization. Time flows upward and lines represent qubits and boxes quantum gates. A half circle represents either a Bell-state creation (ends pointing upwards) or a Bell-state measurement (ends pointing downwards). The Bell-state measurements can produce Pauli errors $P = Z^a X^b$, which are corrected by the boxes $CP$ (corrective Pauli). The computations to determine how errors propagate are performed classically before the computations.

**Proof:**
Figure 3.2 shows the construction of a $\mathsf{LAQCC}$ circuit of width $\mathcal{O}(n)$ and depth $\mathcal{O}(1)$ implementing a Clifford-ladder circuit. The caps and cups denote Bell-state measurements and Bell-state creation, respectively. What remains to show is that an $\mathsf{NC}^1$ circuit computes the Pauli-correction terms.

The $i$-th Bell measurement results in Pauli error $P_i = Z^{a_i} X^{b_i}$. A Clifford-ladder circuit of size $n$ hence has an error vector $\begin{pmatrix} a\,b \end{pmatrix}$ of length $2n$. The correction terms that have to be applied have the same form: we can label every corrective Pauli by an index $j$, such that $\hat{P}_j = Z^{\hat{a}_j} X^{\hat{b}_j}$. This gives a correction vector $\begin{pmatrix} \hat{a}\,\hat{b} \end{pmatrix}$. Note that Pauli matrices anti-commute, hence reordering them will only incur a global phase. This implies a binary linear map $M : \begin{pmatrix} a\,b \end{pmatrix} \mapsto \begin{pmatrix} \hat{a}\,\hat{b} \end{pmatrix}$. As matrix vector multiplication is in $\mathsf{NC}^1$, this error calculation is in $\mathsf{NC}^1$ and Clifford-ladder circuits have an $\mathsf{LAQCC}$ implementation. $\qquad\square$

**3.3.4.** REMARK. Constructing the binary linear map $M$ is not in $\mathsf{NC}^1$, but it does follow directly from the quantum circuit. Instead, an $\mathsf{L}$ (logspace) precomputation gives the matrix associated to $M$.

This result directly implies that in $\mathsf{LAQCC}$ we can apply two-qubit gates on any two non-adjacent qubits.

**3.3.5.** COROLLARY. *Any SWAP circuit needed to do an operation between non-adjacent qubits is a Clifford-ladder circuit and hence in $\mathsf{LAQCC}$.*
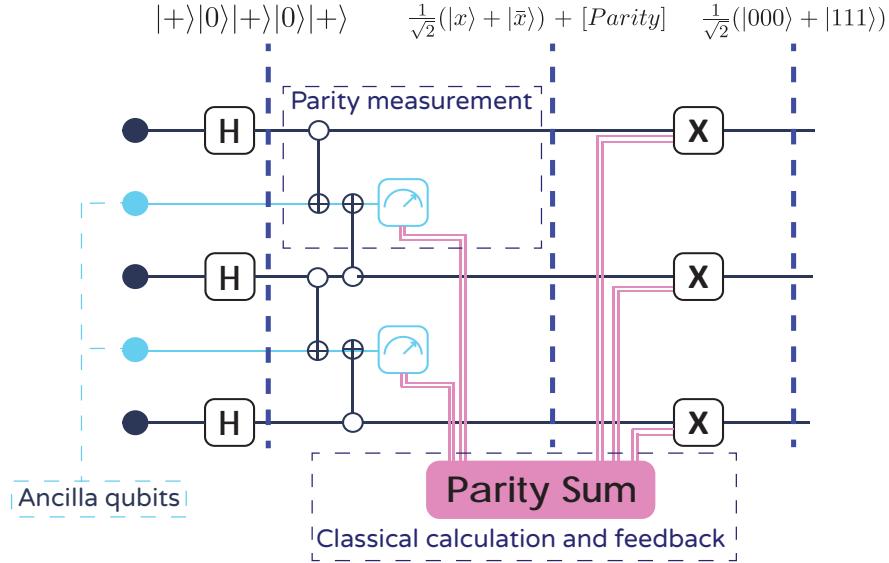
Figure 3.3: The quantum circuit to prepare the 3-qubit GHZ state. Double lines indicate classical information and dotted lines the quantum state at various points.

*This effectively removes the locality constraint in LAQCC for applying a single 2-qubit gate on non-adjacent qubits.*

An example of a Clifford-ladder circuit is the creation of a GHZ state. We can parallelize this directly, for instance following the poor man's cat state approach of [Wat+19]. Figure 3.3 shows a LAQCC circuit using $2n - 1$ qubits placed on a line that prepares an $n$-qubit GHZ state.

A multi-qubit gate that will be very important is the Fanout$_n$ gate, which is a CNOT gate with one control qubit and $n - 1$ output qubits.

$$\text{Fanout}_4 \,|b_0\rangle \,|b_1\rangle \,|b_2\rangle \,|b_3\rangle = |b_0\rangle \,|b_1 \oplus b_0\rangle \,|b_2 \oplus b_0\rangle \,|b_3 \oplus b_0\rangle$$

Hoyer and Spalek have shown that this operation is very powerful when parallelizing unitaries [HŠ05], which will be further discussed in Section 3.3.2. The Fanout$_n$ gate can be implemented by two Clifford ladder circuits:

**3.3.6.** LEMMA. *There is an implementation of the Fanout$_n$ gate by two successive Clifford-ladder circuits.*

**Proof:**
The circuit of two successive Clifford-ladder circuits for the Fanout$_n$ is as follows: First, a right Clifford-ladder circuit with the $U^{(i)}$ consist of the successive CNOT and SWAP gate:

$$C_1 = \prod_{i=0}^{n-2} \text{SWAP}_{i,i+1} \text{CNOT}_{i,i+1},$$

where both the CNOT and the SWAP gate are applied to qubits $i$ and $i+1$. The first application of this unitary applies the CNOT from the control qubit to the first output qubit, and then swaps these two qubits, putting the control qubit next to the second output qubit. This action is repeated until the control qubit is moved past all output qubits. This results in a state with the correct application of the Fanout$_n$ gate, up to a reordering of the qubits, as the control qubit has moved past all output qubits. This ordering is reverted by a left Clifford-ladder circuit where all the $U^{(i)}$ are the SWAP gate:

$$C_2 = \prod_{i=0}^{n-2} \mathsf{SWAP}_{n-(i+2),n-(i+1)}.$$

$C_1$ and $C_2$ together apply the Fanout$_n$ gate. $\qquad\qquad\qquad\qquad\square$

**3.3.7.** COROLLARY. *Fanout$_n$ is accessible in* LAQCC.

**Clifford-grid circuit**

Any Clifford unitary can be mapped to a linear-depth circuit given a linear nearest-neighbor architecture [MR18]. The most general representation of these circuits are so-called Clifford-grid circuits.

**3.3.8.** DEFINITION (Clifford-grid circuit). Let $n$ be the number of qubits. A Clifford-grid circuit of depth $d$ is a circuit of the form

$$C_{grid} = \prod_{i=0}^{d} \bigotimes_{j=0}^{\frac{n}{2}} U_{i,j},$$

for Clifford unitaries $U_{i,j}$, such that gate $U_{i,j}$ acts on qubits $2j$ and $2j+1$ if $i$ is even, and $2j+1$ and $2j+2$ if $i$ is odd.

The next lemma shows that Clifford-grid circuits also have an efficient LAQCC implementation.

**3.3.9.** LEMMA. *Any Clifford-grid circuit of depth $\mathcal{O}(n)$ has an* LAQCC *implementation of depth $\mathcal{O}(1)$ and width $\mathcal{O}(n^2)$.*

**Proof:**
Similar to the Clifford-ladder circuits, gate teleportation allows parallelization to obtain a LAQCC circuit. With a total of $\mathcal{O}(n^2)$ Clifford gates, this also requires $\mathcal{O}(n^2)$ qubits. Figure 3.4 illustrates the transformation.

Any Bell measurement in the circuit can incur a Pauli error, which has to be dealt with at the end of the circuit. The number of Pauli gates now scales with $\mathcal{O}(n^2)$. Similar to the Clifford-ladder circuits, there now is a vector $(ab)$ of length

$\mathcal{O}(n^2)$ containing the information of the Pauli errors. The vector of correction terms, the vector $(\hat{a}\hat{b})$, has length $\mathcal{O}(n)$.

As these Pauli errors anti-commute, there again is a binary linear map $M : (ab) \mapsto (\hat{a}\hat{b})$. The corresponding matrix is rectangular and the error-correction calculations are in $\mathsf{NC}^1$. □
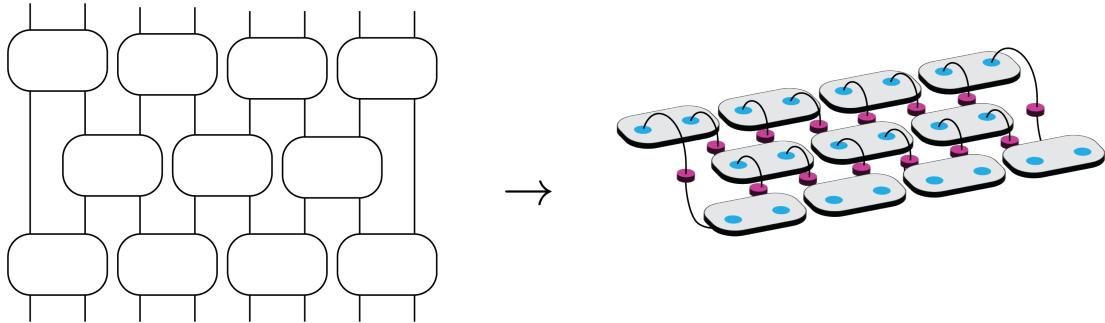


Figure 3.4: Graphical representation of Clifford-grid circuit parallelization. Every blue dot represents a qubit and all Clifford gates (boxes) are applied in parallel. The lines again represent Bell-state creations and Bell-state measurements, indicated by the pink boxes. The propagating Pauli errors can be corrected using the Bell-state measurement results.

Finding the matrix $M$ for correcting a Clifford-grid circuit is more complex than for a Clifford-ladder circuit. An error occurring in the grid can have multiple paths contributing to a single output wire. For the final correction, the parity of each contributing error-path is needed. This computation is in $\oplus\mathsf{L} \subseteq \mathsf{NC}^2$ [2]. A precomputation again gives the matrix corresponding to the bilinear map $M$.

## 3.3.2 Useful gates and subroutines with an LAQCC implementation

This section groups useful multi-qubit gates with an LAQCC implementation. The construction of $W$-states and Dicke states uses these gates, but their use might be of broader interest.

Before we can give a list of accessible gates and subroutines, we need to review two algorithms that we will use to construct our gates. The first concerns Grover's algorithm with zero failure probability [Lon01]. The second concerns parallelization of commuting gates using quantum fan-out gates [HŠ05].

Grover's search algorithm gives a quadratic speed-up for unstructured search [Gro96]. After sufficient iterations, a measurement returns a target state

---

[2]This is not too surprising as simulating Clifford circuits classically is a complete problem for $\oplus\mathsf{L}$

with high probability. Surprisingly, if the exact number of target states is known, a slight modification of the Grover iterates allows for returning a target state with certainty, assuming noiseless computations. Lemma 3.3.13 uses the next lemma to prepare quantum states instead of to find a target state.

**3.3.10.** LEMMA ([Lon01]). *Let $L$ be a set of items and $T \subseteq L$ a set of targets, with $N = |L|$ and $m = |T|$ both known. Let $g : L \to \{0, 1\}$ label the items in $L$ and define the oracle $O_g : |x\rangle |b\rangle \mapsto |x\rangle |b \oplus g(x)\rangle$.*

*Then, there exists a quantum amplitude amplification algorithm that makes $\mathcal{O}(\sqrt{N/m})$ queries to $O_g$ and prepares the quantum state $\frac{1}{\sqrt{|T|}} \sum_{x \in T} |x\rangle$.*

For the other result, we use the quantum fan-out gate, which we showed that is accessible in LAQCC in Lemma 3.3.6. Hoyer and Spalek [HŠ05] introduced this gate and analyzed its properties. The state preparation protocols given in Section 3.4 use the property that the quantum fan-out gate allows parallelization of commuting quantum gates [HŠ05].

**3.3.11.** LEMMA. ([HŠ05, Theorem 3.2])  *Let $\{U_i\}_{i=1}^n$ be a pairwise commuting set of gates on $k$ qubits. Let $U_i^{x_i}$ be the gate $U_i$ controlled by qubit $|x_i\rangle$. Let $T$ be the unitary that mutually diagonalizes all $U_i$. Then there exists a quantum circuit, using quantum fan-out gates, computing $U = \prod_{i=1}^n U_i^{x_i}$ with depth $\max_{i=1}^n \operatorname{depth}(U_i) + 4 \cdot \operatorname{depth}(T) + 2$ and size $\sum_{i=1}^n \operatorname{size}(U_i) + (2n+2) \cdot \operatorname{size}(T) + 2n$, using $(n-1)k$ auxiliary qubits.*

### Gates accessible in LAQCC

Here we give several tables containing useful multi qubit operations accessible in LAQCC. The tables give the action of the gates on computational basis states and the number of qubits required to perform them in LAQCC. Their effect on arbitrary states follows by linearity.

The first two gates directly follow from the Clifford-parallelization results described in Section 3.3.1.

| Gate | Operation on basis states | Width |
|---|---|---|
| Fanout$_n$ | $|x\rangle |y_1\rangle \dots |y_n\rangle \mapsto |x\rangle |y_1 \oplus x\rangle \dots |y_n \oplus x\rangle$ | $\mathcal{O}(n)$ |
| Permutation$(\pi)_n$ | $|y_1\rangle \dots |y_n\rangle \mapsto \left|y_{\pi(1)}\right\rangle \dots \left|y_{\pi(n)}\right\rangle$ | $\mathcal{O}(n^2)$ |

Table 3.1: Operations contained in LAQCC based on Clifford parallelization. Here $\pi \in S_n$ denotes a permutation of $n$ elements.

In the previous chapter, we saw an implementation of the Fanout$_n$ gate as Clifford-ladder circuit 3.3.6. Prior works extensively studied the Fanout$_n$ gate, for instance to construct a constant-depth OR$_n$ function with one-sided error [HŠ05] and with an exact implementation [TT13, Theorem 1], both assuming the Fanout$_n$

gate to be a native gate. The $\mathsf{OR}_n$ gate also implies two other gates, as the following table shows. The implementation of $\mathsf{OR}_n$ gate based on [TT13] can be

| Gate | Operation on basis states | Width |
|------|---------------------------|-------|
| $\mathsf{OR}_n$ | $\lvert y_1\rangle \dots \lvert y_n\rangle \lvert x\rangle \mapsto \lvert y_1\rangle \dots \lvert y_n\rangle \lvert \mathsf{OR}_n(y) \oplus x\rangle$ | $\mathcal{O}(n\log(n))$ |
| $\mathsf{AND}_n$ | $\lvert y_1\rangle \dots \lvert y_n\rangle \lvert x\rangle \mapsto \lvert y_1\rangle \dots \lvert y_n\rangle \lvert \mathsf{AND}_n(y) \oplus x\rangle$ | $\mathcal{O}(n\log(n))$ |
| $\mathrm{Equal}_i$ | $\lvert j\rangle \lvert b\rangle \mapsto \begin{cases} \lvert j\rangle \lvert 1 \oplus b\rangle & \text{if } \lvert j\rangle = \lvert i\rangle \\ \lvert j\rangle \lvert b\rangle & \text{else} \end{cases}$ | $\mathcal{O}(n\log(n))$ |

Table 3.2: Operations contained in LAQCC based on $\mathrm{Fanout}_n$ and local 1-qubit unitaries.

found in 3.B.1. The $\mathsf{AND}_n$ gate is constructed by negating all the inputs and the final output of the $\mathsf{OR}_n$ gate. Similarly $\mathrm{Equal}_i$ is implemented by mapping the bit string $i$ to the all 0 string using $X$ gates, than applying $\mathsf{OR}_n$ and negating the output of $\mathsf{OR}_n$.

With these unbounded-fan-in OR and AND gates, all $\mathsf{AC}^0$ circuits can be implemented. The next step is implementing LAQCC-type modular addition circuits, which gives circuits to check for equality and greater-than. These three gates take $n$-qubit quantum states as input. We introduce the indicator variable $\mathbb{1}_A$ for a Boolean expression $A$, which evaluates to 1 if $A$ is true. Similarly, $\lvert \mathbb{1}_A\rangle = \lvert 1\rangle$ if and only if $A$ is true.

| Gate | Operation on $n$-qubit integers $\lvert x\rangle$, $\lvert y\rangle$ | Width |
|------|----------------------------------------------------------------------|-------|
| $\mathrm{Add}_n$ | $\lvert x\rangle \lvert y\rangle \mapsto \lvert x\rangle \lvert y + x \bmod 2^n\rangle$ | $\mathcal{O}(n^2)$ |
| Equality | $\lvert x\rangle \lvert y\rangle \lvert 0\rangle \mapsto \lvert x\rangle \lvert y\rangle \lvert \mathbb{1}_{x=y}\rangle$ | $\mathcal{O}(n^2)$ |
| GreaterThan | $\lvert x\rangle \lvert y\rangle \lvert 0\rangle \mapsto \lvert x\rangle \lvert y\rangle \lvert \mathbb{1}_{x>y}\rangle$ | $\mathcal{O}(n^2)$ |

Table 3.3: Operations contained in LAQCC based on $\mathsf{AC}^0$ circuits.

$\mathrm{Add}_n$ can be implemented by an $\mathsf{AC}^0$ circuit, which by the previous table is accessible in LAQCC. The implementation of Equality and GreaterThan are constructed using $\mathrm{Add}_n$ and can be found in Appendix 3.B.2 and Appendix 3.B.3 respectively. Høyer and Špalek showed that fanout-gates imply efficient constant-depth implementations of for instance the quantum Fourier transform [HŠ05]. They use this constant-depth quantum Fourier transform to construct a constant-depth circuit for weighted counting. In particular, this circuit can be used to calculate the Hamming weight of an $n$-bit string, and to implement an "Exact $t$"-gate and a threshold gate. Appendix 3.B.5 also explains how to modify the threshold gate to a weighted threshold gate. The implementation of the QFT can be found in [HŠ05, Theorem 4.12] and the implementation of the Hammingweight gate in [TT13, Lemma 4]. The implementations of $\mathrm{Exact}_t$ and $\mathrm{Threshold}_t$ can be found in Appendix 3.B.4 and Appendix 3.B.5 respectively.

| Gate | Operation on $n$-qubit basis state $|x\rangle$ | Width |
|------|-----------------------------------------------|-------|
| QFT | $|x\rangle \mapsto \frac{1}{\sqrt{2^{n-1}}} \sum_{j=0}^{2^{n-1}} e^{i2\pi \frac{x \cdot j}{2^n}} |j\rangle$ | $\mathcal{O}(n^3 \log(n))$ |
| Hammingweight | $|x\rangle_n |0\rangle_{\log(n)} \mapsto |x\rangle_n \big||x|\big\rangle_{\log(n)}$ | $\mathcal{O}(n \log(n))$ |
| $\text{Exact}_t$ | $|x\rangle |0\rangle \mapsto |x\rangle \big|\mathbb{1}_{|x|=t}\big\rangle$ | $\mathcal{O}(n \log(n))$ |
| $\text{Threshold}_t$ | $|x\rangle |0\rangle \mapsto |x\rangle \big|\mathbb{1}_{\sum_i x_i \geq t}\big\rangle$ | $\mathcal{O}(tn \log(n))$ |

Table 3.4: Quantum subroutines in $\mathsf{LAQCC}$ based on Høyer and Špalek.

**3.3.12.** REMARK. As the $\text{Threshold}_t$ gate is in $\mathsf{LAQCC}$, any classical $\mathsf{TC}_0$ circuit is in $\mathsf{LAQCC}$.

**Subroutine accessible in $\mathsf{LAQCC}$**

This section concludes not with a gate, but with a tool used for preparing uniform superpositions. This lemma extends Lemma 3.3.10 to preparing states instead of finding marked items.

**3.3.13.** LEMMA. *Given an $n$-qubit unitary $U$, that is implementable by a constant-depth circuit, a basis $\mathcal{C}$ and a partition of $\mathcal{C}$ in $\mathcal{G}$ and $\mathcal{B}$ such that $\frac{|\mathcal{G}|}{|\mathcal{C}|}$ is a known constant $c$. Suppose that $U$ implements 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 $\mathsf{LAQCC}$ circuit that prepares the state $\frac{1}{\sqrt{|\mathcal{G}|}} \sum_{y \in \mathcal{G}} |y\rangle$ by using $U$ a constant number of times.*

**Proof:**
Define $|\mathcal{G}\rangle = \frac{1}{\sqrt{|\mathcal{G}|}} \sum_{y \in \mathcal{G}} |y\rangle$ and $|\mathcal{B}\rangle = \frac{1}{\sqrt{|\mathcal{B}|}} \sum_{y \in \mathcal{B}} |y\rangle$. As $\mathcal{B}$ and $\mathcal{G}$ partition $\mathcal{C}$, it follows that $\langle \mathcal{G}|\mathcal{B}\rangle = 0$. Lemma 3.3.10 implies the existence of a circuit that prepares the desired state. Below, we explicitly construct the circuit.

First, prepare a uniform superposition $\sum_{i=0}^{2^n-1} |i\rangle$. Then, iteratively reflect over the state $|\mathcal{B}\rangle$ using $U$, and reflect over the uniform superposition state $\sum_{i=0}^{2^n-1} |i\rangle$. Both reflections have a $\mathsf{LAQCC}$ implementation and we only need to apply them a constant number of iterations.

To reflect over the uniform superposition, we have to implement the operation $2|s\rangle \langle s| - I$, with $|s\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} |i\rangle$. To implement this operation, we first apply a layer of Hadamards, which implements a basis transformation mapping the uniform superposition state to the all zeroes state; Then apply the $\text{Exact}_0$-gate producing an output qubit that marks only the all zeroes-state and finally negate the output qubit and applies a $Z$-gate on it. Running this circuit in reverse, excluding the $Z$-gate, resets the output qubit and reverts the basis transformation.

The last step of Lemma 3.3.13 requires a reflection using an $R_Z$-gate (rotational $Z$-gate) instead of the $Z$-gate. As the $\text{Exact}_0$-gate has an LAQCC implementation (see Table 3.4), this second inversion operation has a LAQCC implementation.

The total number of iterations is $\mathcal{O}(\sqrt{N/m})$, where $N = |\mathcal{C}|$ and $m = |\mathcal{G}|$. As their fraction is the constant $c$, it follows that $\mathcal{O}(\sqrt{c}) = \mathcal{O}(1)$ iterations are needed. $\qquad\square$

### 3.3.3 Non-simulatability of LAQCC

Most of the power of LAQCC circuits seems to come from the classical intermediate calculations, which makes one wonder if these circuits are classically simulatable. Even if these circuits were indeed efficiently simulatable, they still have value as "fast" alternatives for state preparation. However, it is unlikely that all LAQCC circuits can be simulated efficiently by a classical simulator. Lemma 3.3.11 and the inclusion of the fan-out gate in LAQCC show that circuits consisting of commuting gates have an LAQCC implementation and in particular, the class of Instantaneous Quantum Polynomial-time (IQP) circuits, first introduced in [SB09], has equivalent LAQCC implementations.

**3.3.14.** DEFINITION (Definition 2 [NM14]). An IQP circuit on $n$ qubits is a quantum circuit with the following structure: 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.

**3.3.15.** LEMMA. *Any* IQP *circuit has an* LAQCC *implementation.*

**Proof:**
The following LAQCC circuit prepares the desired state: First prepare $|+\rangle^{\otimes n}$ by a single layer of Hadamard gates on all qubits. All the gates in the diagonal block of the IQP circuit commute, therefore we can apply Lemma 3.3.11. This allows us to parallelize all gates in the diagonal block using $poly(n)$ auxiliary qubits. Next, we can again apply a layer of Hadamard gates and finally measure the desired qubits. $\qquad\square$

Bremner, Jozsa, and Shepherd showed that efficient weak classical simulation of all possible IQP circuits up to small multiplicative error implies a collapse of the polynomial hierarchy [BJS10]. Note that a circuit family is weakly simulatable if given the description of the circuit family, its output distribution can be sampled by purely classical means in $poly(n)$ time.

**3.3.16.** LEMMA (Corollary 1 [BJS10]). *If the output probability distributions generated by uniform families of* IQP *circuits could be weakly classically simulated to within multiplicative error* $1 \leq c < \sqrt{2}$ *then the polynomial hierarchy would collapse to the third level, in particular,* PH $= \Delta_3^p$.

**3.3.17.** Corollary. *If the output probability distributions generated by uniform families of* LAQCC *circuits could be weakly classically simulated to within multiplicative error* $1 \leq c < \sqrt{2}$ *then the polynomial hierarchy would collapse to the third level, in particular,* $\mathsf{PH} = \Delta_3^p$.

## 3.3.4   Relationship LAQCC to QNC$^1$

Let $A$ be an LAQCC-circuit. We can write this circuit as a composition of unitary quantum layers $U_i$, measurements $M_i$ and classical calculation layers $C_i$:

$$A = M_k U_k C_k \ldots M_i U_i C_i \ldots M_1 U_1 C_1,$$

for some constant $k$. Any unitary $U_i$ is a QNC$^0$ circuit and any $C_i$ is 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]. Furthermore, in Lemma 3.A.1 we show that any NC$^1$ circuit can be replaced by a QNC$^1$ circuit, which performs the exact same calculation in superposition. This seems to imply that for any LAQCC circuit there exists a QNC$^1$ circuit that simulates it. However, when we try to prove this, we run into a problem which even seems to suggest that this might not be true at all.

The difficulty arises from the unitary applied after the measurement. When a measurement is performed, the quantum state collapses into a single branch of the superposition. Based on this measurement outcome, an NC$^1$ circuit determines which unitary to apply next. However, this unitary may vary significantly depending on the classical outcome.

If we replace the measurement with a CNOT, the state no longer collapses. Instead, it remains in a superposition, with each branch of the superposition corresponding to a different outcome of the original measurement. In this setting, a QNC$^1$ circuit replaces the NC$^1$ computation, coherently computing a superposition of all possible classical outputs. The subsequent unitary must now act coherently on this superposition, applying the appropriate correction to each branch.

This correction unitary can be significantly more complex than the one used after a classical measurement. For example, suppose the correction consists of a single CNOT gate applied between qubits $i$ and $j$, where $i$ and $j$ are determined by the output of the NC$^1$ circuit. Replacing the measurement with CNOTs and a QNC$^1$ circuit produces a superposition over all possible $i, j$ pairs. The corresponding correction must then consist of CNOT gates applied between each possible pair $i, j$, controlled on the output of the QNC$^1$ circuit. Without additional auxiliary qubits, implementing this unitary requires linear depth.

### 3.3.5 Complexity results for LAQCC$(\mathcal{Q}, \mathcal{C}, d)$

In its current definition, LAQCC$(\mathcal{Q}, \mathcal{C}, d)$, and hence also LAQCC, are classes of circuits. When considering the capabilities of LAQCC$(\mathcal{Q}, \mathcal{C}, d)$ in preparing states, it is helpful to define a related class that consists of states preparable by a circuit in LAQCC$(\mathcal{Q}, \mathcal{C}, d)$.

**3.3.18.** DEFINITION (StateClassX). Let $\mathcal{H}_n$ be a Hilbert space on $n$ qubits, then define

$$\mathsf{StateX}_{n,\varepsilon} = \{|\psi\rangle \in \mathcal{H}_n \mid \exists\, \mathsf{X}\text{-circuit } A : \langle\psi| A |0\rangle^{\otimes n} \geq 1 - \varepsilon\}.$$

This is the subset of $n$-qubit states $|\psi\rangle$ such that there exists a circuit corresponding to the class X that prepares a quantum state that has inner product at least $1 - \varepsilon$ with $|\psi\rangle$.

Define $\mathsf{StateX}_\varepsilon = \bigcup_{n \in \mathbb{N}} \mathsf{StateX}_{n,\varepsilon}$.

This definition extends already existing ideas and definitions of state-complexity [AAS20; RY22; Sus18]. Our definition is very similar to state complexity defined in [MY23], where we are interested in which states are contained in a class, however we drop the uniformity requirement and instead study the set of states that can be generated by a specific class of circuits. An example of a circuit class is $\mathsf{StateLAQCC}(\mathcal{Q}, \mathcal{C}, d)_{n,\varepsilon}$.

**3.3.19.** NOTATION. *The class* $\mathsf{StateLAQCC}(\mathcal{Q}, \mathcal{C}, d)_{n,\varepsilon}$ *consists of all $n$-qubit states $|\psi\rangle$ for which an* LAQCC$(\mathcal{Q}, \mathcal{C}, d)$ *circuit exists that prepares a state that has inner product at least $1 - \varepsilon$ with $|\psi\rangle$.*

Another example is the circuit class of PostQPoly.

**3.3.20.** DEFINITION (PostQPoly). The class PostQPoly consists of all polynomial-sized quantum circuits with one extra qubit, where the outcome state is considered conditional on the extra qubit being in the one state. If the extra qubit is in the zero state, the output state may be anything.

The class $\mathsf{StatePostQPoly}_{n,\varepsilon}$ consists of all $n$-qubit states $|\psi\rangle$ for which a polynomial-sized quantum circuit exists that prepares a state that, conditional on the extra qubit being one, has inner product at least $1 - \varepsilon$ with $|\psi\rangle$.

In the definition of LAQCC$(\mathcal{Q}, \mathcal{C}, d)$, we have freedom to choose $\mathcal{Q}$ and $\mathcal{C}$. If we give more power to both the quantum and the classical routines, we see that we can solve more complex problems and prepare a wider variety of quantum states. Yet, even with polynomial depth quantum circuits and unbounded classical computational power, limits exist.

**3.3.21.** NOTATION. *The class* LAQCC* *is the instantiation* LAQCC(QPoly($n$), ALL, poly($n$)): *The class of polynomially many alternating polynomial-sized quantum circuits and arbitrary powerful classical computations, together with feed-forward of the classical information to future quantum operations. The quantum computations are restricted to all single-qubit gates and the two-qubit CNOT gate.*

This directly gives us a definition of StateLAQCC$^*_\varepsilon$ for $\varepsilon > 0$.

**3.3.22.** REMARK. Note that for any non-zero $\varepsilon$, we can restrict ourselves to finite universal gate-sets. The Solovay-Kitaev theorem [Kit97; NC10] says that any multi-qubit unitary can be approximated to within precision $\delta$ by a quantum circuit with size depending on $\delta$. Therefore, with a finite universal gate-set, any LAQCC* circuit with a continuous gate-set can be approximated by an LAQCC* circuit with gates from the finite set.

Next, we prove StateLAQCC$^*_\varepsilon \subseteq$ StatePostQPoly$_\varepsilon$. Following the same decomposition as in the previous section, we find that any LAQCC* can be written as

$$\Pi_{i=0}^{\mathrm{poly}(n)} M_i U_i(y_i) C_i(x_i) |0\rangle^{\otimes \mathrm{poly}(n)}, \tag{3.1}$$

where again, $M_i$ denotes the $i$-th measurement layer, $U_i$ the $i$-th quantum layer and $C_i$ the $i$-th unbounded classical computation layer. The $x_i \in \{0,1\}^*$ denote the outcomes 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 $n$.

**3.3.23.** THEOREM. *It holds that* StateLAQCC$^*_\varepsilon \subseteq$ StatePostQPoly$_\varepsilon$.

**Proof:**
Fix $\varepsilon > 0$ and a positive integer $n$ and let $|\psi\rangle \in$ StateLAQCC$^*_\varepsilon$. By definition, there exists an LAQCC* circuit $A = \Pi_{i=0}^{\mathrm{poly}(n)} M_i U_i(y_i) C_i(x_i)$, which prepares a state $|\phi\rangle$ with inner product at least $1 - \varepsilon$ with $|\psi\rangle$.

Then consider the following PostQPoly-circuit: 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. The $\mathrm{Equal}_{x_i}$ gate replaces the measurement layer $M_i$, by checking if the subset of qubits that would be measured are in $|x_i\rangle$ computational basis state. It stores the output in an auxiliary qubit. As a last step, apply an $\mathrm{AND}_{\mathrm{poly}(n)}$-gate on the auxiliary qubits, which hold the outputs of the $\mathrm{Equal}_{x_i}$ gates, and store the result in an auxiliary qubit. Conditional on this last auxiliary qubit being one, the circuit prepares the state $|\phi\rangle$. □

Figure 3.5 gives a schematic overview of the proof and the translation of an LAQCC* circuit in a PostQPoly-circuit.
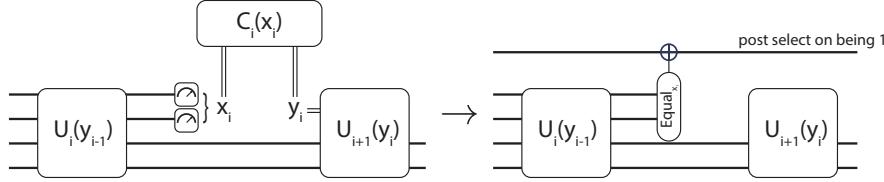
Figure 3.5: Schematic idea of transforming an LAQCC* circuit for generating $|\psi\rangle$ into a PostQPoly circuit.

# 3.4 State preparation in LAQCC

In this section we consider which quantum states we can prepare using an LAQCC circuit beyond the stabilizer states and Clifford circuits discussed in the previous section. Specifically, as noted in the introduction, we consider quantum states widely used in other quantum algorithms, for benchmarking, and in physics. First, we show how to create a uniform superposition of computational basis states for arbitrary $q$, where $q$ is not a power of 2, a state that is often used as initial state in other algorithms (including the following other state preparation protocols presented in this work). We then use this procedure to create $W$-states, the uniform superposition over all $n$-bit strings of Hamming weight 1, using a compress-uncompress method. This compress-uncompress method generalizes to preparing Dicke-$(n, k)$ states for $k = \mathcal{O}(\sqrt{n})$, uniform superpositions over all $n$-bitstrings of Hamming weight $k = \mathcal{O}(\sqrt{n})$. Dicke states find many applications, and especially the compress-uncompress approach might prove useful for entanglement distillation protocols. Preparing general Dicke-$(n, k)$ states requires a novel method to map between two integer representation systems, the factoradic representation and the combinatorial number representation. Finally, we present a state preparation protocol for quantum many-body scar states, states often used in physics, based on the Dicke-$(n, k)$ state preparation protocol for $k = \mathcal{O}(\sqrt{n})$.

## 3.4.1 Uniform superposition of size $q$

The uniform superposition is often used as an initial state in other quantum algorithms. A simple Hadamard gate applied to $n$ qubits prepares the uniform superposition $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$. Preparing the state $\frac{1}{\sqrt{q}} \sum_{i=0}^{q-1} |i\rangle$, the superposition up to size $q$, for arbitrary $q$, is less straightforward.

A simple probabilistic approach works as follows: 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 ancilla qubit; 3) measure this ancilla qubit. Based on the measurement result, the desired superposition is found, which happens with probability at least one half.

The next theorem modifies this probabilistic approach to a protocol that deterministically prepares the uniform superposition modulo $q$ in LAQCC.

**3.4.1.** THEOREM. *There is a deterministic* LAQCC *circuit that prepares the uniform superposition of size* $q$. *This circuit requires* $\mathcal{O}(\lceil \log_2(q)\rceil^2)$ *qubits.*

**Proof:**
Let $n = \lceil \log_2(q)\rceil$ and define $\mathcal{G} = \{i \mid 0 \le i < q\}$ and $\mathcal{B} = \{i \mid q \le i \le 2^n - 1\}$. Construct the unitary

$$U_q : |y\rangle\, |b\rangle \mapsto \begin{cases} |y\rangle\, |b \oplus 1\rangle & \text{if } y < q \\ |y\rangle\, |b\rangle & \text{if } y \ge q \end{cases}.$$

The Greaterthan-gate of Table 3.3 implements the operator $U_q$, note that this gate requires $\mathcal{O}(n^2)$ qubits.

As $|\mathcal{G}|/2^n \ge 1/2$ and known, applying Lemma 3.3.13 with the sets $\mathcal{G}$ and $\mathcal{B}$ and the constant-depth implementation of $U_q$, gives an LAQCC algorithm that boosts the amplitude of $|\mathcal{G}\rangle$ to 1. □

**3.4.2.** REMARK. Note that in Lemma 3.3.13 it was implicitly assumed that $|\mathcal{G}| + |\mathcal{B}|$ is a power of two (allowing for a simple reflection over the uniform superposition state). This LAQCC implementation of creating a uniform superposition modulo any $q$ removes this requirement.

## 3.4.2   $W$-state in LAQCC

In this section we consider the $W_n$-state and how to prepare this state in LAQCC. The $W_n$-state is a uniform superposition over all $n$-qubit states with a single qubit in the $|1\rangle$-state and all others in the $|0\rangle$-state:

$$|W_n\rangle = \frac{1}{\sqrt{n}} \sum_i |e_i\rangle,$$

where $|e_i\rangle$ is the state with a one on the $i$-th position and zeroes elsewhere.

A first observation is that the $W$-state can be seen as a one-hot encoding of a uniform superposition over $n$ elements. We can label the $n$ states with non-zero amplitude of the $W$-state with an index. More precisely, we want to design circuits that implement the following map:

$$|i\rangle\, |0\rangle \mapsto |0\rangle\, |e_i\rangle, \tag{3.2}$$

with $i$ an index and $e_i$ the one-hot encoding of $i$. This index – which equals the position of the 1 – compresses the representation from $n$ to $\log(n)$ bits. This compression naturally defines two operations:

$$\textbf{Uncompress:}\quad |i\rangle_{\log(n)}\, |0\rangle_n \mapsto |i\rangle_{\log(n)}\, |e_i\rangle_n, \tag{3.3}$$

$$\textbf{Compress:}\quad |i\rangle_{\log(n)}\, |e_i\rangle_n \mapsto |0\rangle_{\log(n)}\, |e_i\rangle_n. \tag{3.4}$$

Implementing both and combining them implements Mapping 3.2 giving an efficient $W$-state preparation protocol.

The **Compress** and **Uncompress** operations map between a one-hot and binary representation of an integer $i$. We call the registers containing the binary representation index registers, and the register containing the one-hot representation the system register. The index registers serve as ancilla qubits and the $W$-state is prepared in the system register.

**3.4.3.** LEMMA. *There exists an* LAQCC *circuit that, for any $n$, implements the* ***Uncompress*** *operation:*

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log(n)} |0\rangle_n \mapsto \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |e_i\rangle_n.$$

*This circuit uses $\mathcal{O}(n \log(n) \log(\log(n)))$ qubits placed in a grid pattern of size $n \times \log(n) \log(\log(n))$.*

**Proof:**
One column of the grid of length $n$ consists of system qubits placed in a line. Adjacent to this line are $\log(n) \log(\log(n))$ index qubits. The left grid in Figure 3.6 shows the initial layout. The same figure also shows the steps to prepare the $W$-state in the system qubits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log(n)} |0\rangle_{\log(n)}^{\otimes n-1} |0\rangle_n \xrightarrow{(1)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log(n)}^{\otimes n} |0\rangle_n$$

$$\xrightarrow{(2)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log(n)}^{\otimes n} |e_i\rangle_n$$

$$\xrightarrow{(3)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |0\rangle^{\otimes n-1} |e_i\rangle_n$$

Step (1) uses fanout-gates to create a fully entangled state between the different index registers. Step (2) applies Equal$_i$ gates in parallel from each index register to its corresponding system qubit to create the state $|e_i\rangle$ in the system register. Every Equal$_i$ gate requires additional an additional $\log \log n$ qubits to perform. Step (3) uses fanout-gates to disentangle and reset the index registers. Combined the **Uncompress** operation maps $\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log(n)} |0\rangle_{\log(n)}^{\otimes n-1} |0\rangle_n \mapsto \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |0\rangle^{\otimes n-1} |e_i\rangle_n$ as required. $\square$

**3.4.4.** LEMMA. *There exists an* LAQCC *circuit that, for any $n$, implements the* ***Compress*** *operation:*

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle_{\log(n)} |e_i\rangle_n \mapsto \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |0\rangle |e_i\rangle_n.$$

*This circuit uses $\mathcal{O}(n \log(n))$ qubits placed in a grid pattern of size $n \times \log(n)$.*
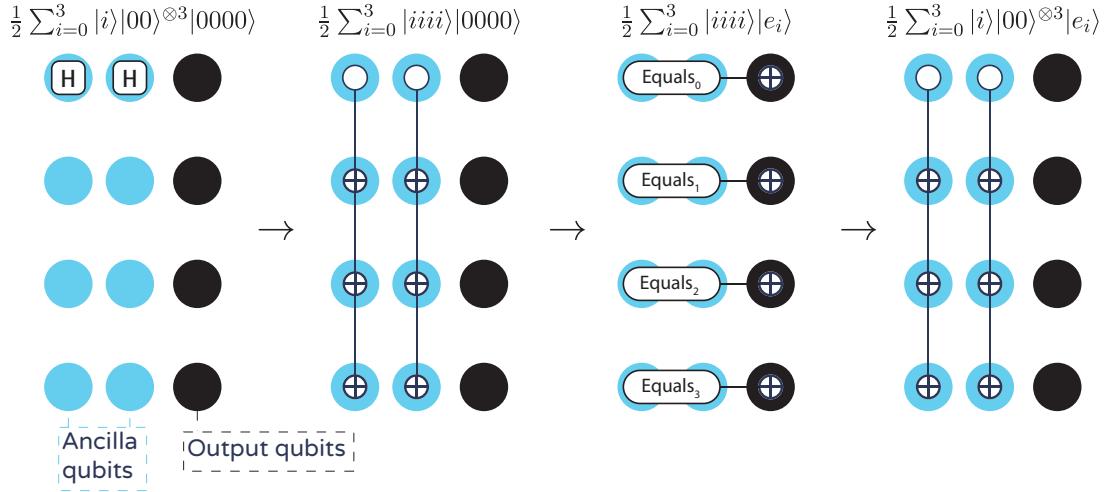
Figure 3.6: Circuit for the **Uncompress** operation for $n = 4$. Shown is a grid of 12 qubits: 8 blue index qubits, and 4 black system qubits. This schematic is simplified, it does not show the additional $\log \log n$ qubits required to perform the $\text{Equal}_i$ gates, nor some additional qubits to perform the fanout gate. Each of the four grids represents a single time slice in the **Uncompress** operation.

**Proof:**

To implement **Compress**, the index registers are uncomputed using parallel $CNOT$-operations, controlled by the system register. These controlled gates commute for different indices in the system register and hence by Lemma 3.3.11 a parallel circuit for the uncomputation exists. The **Compress** operation, also shown in Figure 3.7, consists of the operations:

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle_{\log(n)} |0\rangle_{\log(n)}^{\otimes n-1} |e_i\rangle_n \xrightarrow{(1)} \frac{1}{n} \sum_{i,j=0}^{n} (-1)^{i \cdot j} |j\rangle_{\log(n)} |0\rangle_{\log(n)}^{\otimes n-1} |e_i\rangle_n$$

$$\xrightarrow{(2)} \frac{1}{n} \sum_{i,j=0}^{n} (-1)^{i \cdot j} |j\rangle_{\log(n)}^{\otimes n} |e_i\rangle_n$$

$$\xrightarrow{(3)} \frac{1}{n} \sum_{i,j=0}^{n} |j\rangle_{\log(n)}^{\otimes n} |e_i\rangle_n$$

$$\xrightarrow{(4)} \frac{1}{n} \sum_{i,j=0}^{n} |j\rangle_{\log(n)} |0\rangle_{\log(n)}^{\otimes n-1} |e_i\rangle_n$$

$$\xrightarrow{(5)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n} |0\rangle_{\log(n)}^{\otimes n} |e_i\rangle_n$$

Step (1) applies Hadamard gates to the first index register, changing from the computational to the Hadamard basis, in which the $NOT$-operation is diagonal; Step (2) uses fanout-gates to create a fully entangled state in the index registers; Step (3) applies controlled-$Z$ gates, controlled by the system qubit $i$ and with

targets the qubits in the $i$-th index register corresponding to the ones in the binary representation of $i$; Step (4) disentangles the index registers using fanout-gates; and, Step (5) applies Hadamard gates to clean the index register.

The controlled-$Z$ gates in Step (3) apply phases that precisely cancel the phases already present, which disentangles the index registers from the system register. $\square$
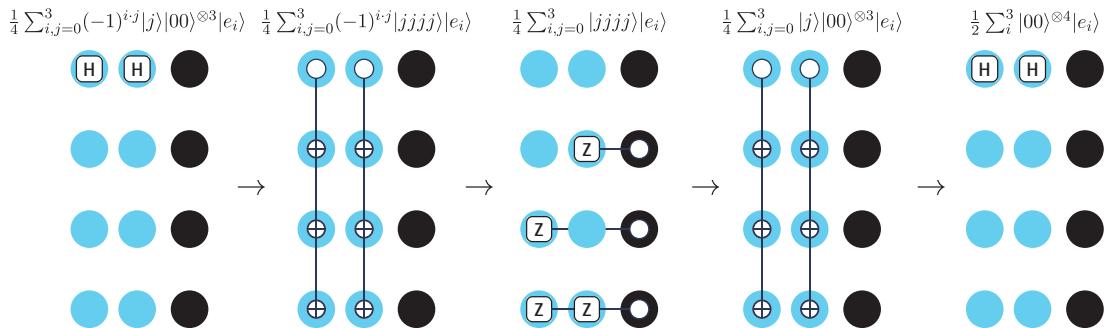


Figure 3.7: Circuit for the **Uncompress** operation for $n = 4$. Shown is a grid of 12 qubits: 8 blue index qubits, and 4 black system qubits. Each of the four grids represents a single time slice in the **Compress** circuit.

**3.4.5.** THEOREM. *There exists a circuit in* LAQCC *that prepares the $|W_n\rangle$ state. This circuit requires $\mathcal{O}(n \log(n) \log(\log(n)))$ qubits placed in a grid of size $n \times \log(n) \log(\log(n))$.*

**Proof:**
The circuit combines the circuits of Theorem 3.4.1, Lemma 3.4.3, and Lemma 3.4.4. It consists of three steps:

$$|0\rangle_{\log(n)}^{\otimes n} |0\rangle_n \xrightarrow{(1)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |0\rangle^{\otimes n-1} |0\rangle$$

$$\xrightarrow{(2)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle |0\rangle^{\otimes n-1} |e_i\rangle$$

$$\xrightarrow{(3)} \frac{1}{\sqrt{n}} \sum_{i=0}^{n} |0\rangle^{\otimes n} |e_i\rangle$$

Step (1) prepares the uniform superposition over indices. This can be done either by applying a layer of Hadamard gates, if $n$ is a power of 2, requiring $\mathcal{O}(log(n))$ qubits, or using Theorem 3.4.1 if $n$ is not a power of 2 requiring $\mathcal{O}(\lceil \log(n) \rceil^2)$ qubits; Step (2) is by Lemma 3.4.3 and requires $\mathcal{O}(n \log(n) \log(\log(n)))$ qubits; and, Step(3) is by Lemma 3.4.4 and requires $\mathcal{O}(n \log(n))$ qubits. $\square$

### 3.4.3 Dicke states for small $k$

In this section we generalize our method of preparing the $|W\rangle$-state in Theorem 3.4.5 to a more general set of states, Dicke states. A Dicke-$(n, k)$ state is the uniform superposition over bitstrings of Hamming weight $k$ and length $n$, which we again assume to be a power of 2 for simplicity:

$$|D_k^n\rangle = \binom{n}{k}^{-1/2} \sum_{x \in \{0,1\}^n : |x| = k} |x\rangle. \tag{3.5}$$

For $k = 1$, this state is precisely the $W$-state. There exists an efficient deterministic method to prepare a $|D_k^n\rangle$ state that requires a circuit of width $\mathcal{O}(n)$ and depth $\mathcal{O}(n)$, independent of $k$ [BE19]. This method starts from the $|1\rangle^{\otimes k} |0\rangle^{\otimes k-n}$ state and relies on a recursive formula for the Dicke state

$$|D_k^n\rangle = \alpha_{k,n} \left| D_k^{n-1} \right\rangle \otimes |0\rangle + \beta_{k,n} \left| D_{k-1}^{n-1} \right\rangle \otimes |1\rangle.$$

This relation implies a protocol that is inherently sequential, which is unsuited for an LAQCC implementation.

Instead, we present an LAQCC approach similar to the $W$-state preparation protocol. We apply the **Uncompress** operation of the $W$-state in parallel to put $k$ ones into the bitstring. This method allows for the preparation of Dicke states with $k = \mathcal{O}(\sqrt{n})$, using $\mathcal{O}(n^2 \log(n)^3)$ qubits. The bound on $k$ comes from the fact that using the **Uncompress** operation in parallel might cause overlaps to where the 1's are put into the system register. Having two ones in the same system qubit in effect negates the **Uncompress** operation. Following the lines of the birthday paradox, we find that overlaps between different indices happen not that often for $k = \mathcal{O}(\sqrt{n})$. Lemma 3.3.13 allows us to boost the amplitudes and make the protocol deterministic.

Again, consider two groups of qubits: Index registers with $\log(n)$ qubits each; and, system registers of $n$ qubits each. Contrary to the $W$-state, the Dicke state requires multiple system registers during the preparation. The state is prepared in only one system register. Denote the index registers with subscripts $i_1$ up to $i_k$ and the system registers with $s_1$ up to $s_n$. For clarity, these indices may be omitted if it is clear from the context.

The algorithm consists of four steps:

1. **Filling**: $|0\rangle_{i_1} \ldots |0\rangle_{i_k} |0\rangle_{s_1} \to \frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle_{i_1} \ldots |j_k\rangle_{i_k} |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle_{s_1}$

2. **Filtering**: $\to \sqrt{\frac{(n-k)!}{n!}} \sum_{j_1 \neq \ldots \neq j_k}^{n-1} |j_1\rangle \ldots |j_k\rangle |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle$

3. **Ordering**: $\to \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \cdots < j_k}^{n-1} |j_1\rangle \ldots |j_k\rangle |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle$

4. **Cleaning**: $\to \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \cdots < j_k}^{n-1} |0\rangle \ldots |0\rangle |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle$

Note that after **Filling** there is a multiplicity in states. First, **Filtering** removes those states in which different indices $j_l$ are the same, resulting in an incorrect state in the $s_1$ register. Second, **Ordering** removes the multiplicity from having multiple permutations of the index registers creating the same state in the $s_1$ register, by forcing a choice of ordering on the indices. These two steps give a unique pairing between index registers and system registers allowing the operation **Cleaning**.

We will now proof that these four steps are achievable in LAQCC and explicitly visualize the corresponding circuits for $n = 4$ and $k = 2$.

**3.4.6.** LEMMA. *An* LAQCC *circuit exists that implements* ***Filling****:*

$$|0\rangle_{i_1} \ldots |0\rangle_{i_k} |0\rangle_{s_1} \to \frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle_{i_1} \ldots |j_k\rangle_{i_k} |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle_{s_1}.$$

*This circuit uses* $\mathcal{O}(kn\log(n)\log(\log(n)))$ *qubits.*

**Proof:**
To achieve a circuit implementing **Filling** we use **Uncompress** from Lemma 3.4.3 $k$ times in parallel. Since two **Uncompress** operations commute, hence by Lemma 3.3.11 $k$ **Uncompress** operations can be implemented in parallel. Each of these parallel operations requires an index register, a system register and $\mathcal{O}(n\log(n)\log(\log(n)))$ extra ancilla qubits.

The corresponding circuit consists of five steps starting from the all zero state $|0\rangle_{i_1} \ldots |0\rangle_{i_k} |0\rangle_{s_1} \ldots |0\rangle_{s_k}$:

$$\xrightarrow{(1)} \frac{1}{n^{k/2}} \sum_{j_1\ldots j_k=0}^{n-1} |j_1\rangle \ldots |j_k\rangle \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} |l\rangle_{s_1} |0\rangle_{s_2} \ldots |0\rangle_{s_k}$$

$$\xrightarrow{(2)} \frac{1}{n^{k/2}} \sum_{j_1\ldots j_k=0}^{n-1} |j_1\rangle \ldots |j_k\rangle \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} |l\rangle_{s_1} |l\rangle_{s_2} \ldots |l\rangle_{s_k}$$

$$\xrightarrow{(3)} \frac{1}{n^{k/2}} \sum_{j_1\ldots j_k=0}^{n-1} |j_1\rangle \ldots |j_k\rangle \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} (-1)^{(2^{j_1}+\cdots+2^{j_k})\cdot l} |l\rangle_{s_1} |l\rangle_{s_2} \ldots |l\rangle_{s_k}$$

$$\xrightarrow{(4)} \frac{1}{n^{k/2}} \sum_{j_1\ldots j_k=0}^{n-1} |j_1\rangle \ldots |j_k\rangle \frac{1}{\sqrt{2^n}} \sum_{l=0}^{2^n-1} (-1)^{(2^{j_1}+\cdots+2^{j_k})\cdot l} |l\rangle_{s_1} |0\rangle_{s_2} \ldots |0\rangle_{s_k}$$

$$\xrightarrow{(5)} \frac{1}{n^{k/2}} \sum_{j_1\ldots j_k=0}^{n-1} |j_1\rangle \ldots |j_k\rangle |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle_{s_1} |0\rangle_{s_2} \ldots |0\rangle_{s_k}$$

Step (1) brings all index registers in a uniform superposition of size $n$, use Theorem 3.4.1 if required, and one system register in a uniform superposition of size $2^n$; Step (2) uses fan-out gates to create entangled copies of the system register; Step (3) applies a phase flip between every pair of index and system register using

**Uncompress** of Lemma 3.4.3, except instead of applying not gates to the system registers, apply phase gates; Step (4) uses fan-out gates to disentangle and uncompute all but one of the system registers; Step (5) applies Hadamard gates on the system register to obtain the one-hot representation of the index registers. Step (3), the step that requires most qubits, requires $\mathcal{O}(n \log(n) \log(\log(n)))$ qubits for every pair of index and system register, of which there are $k$, resulting in the requirement of $\mathcal{O}(kn \log(n) \log(\log(n)))$ qubits. □

Figure 3.8 shows these five steps graphically. Ancilla qubits are omitted for clarity. Note that some of the $j_i$ in the index registers may intersect. The next Filtering step takes care of that.
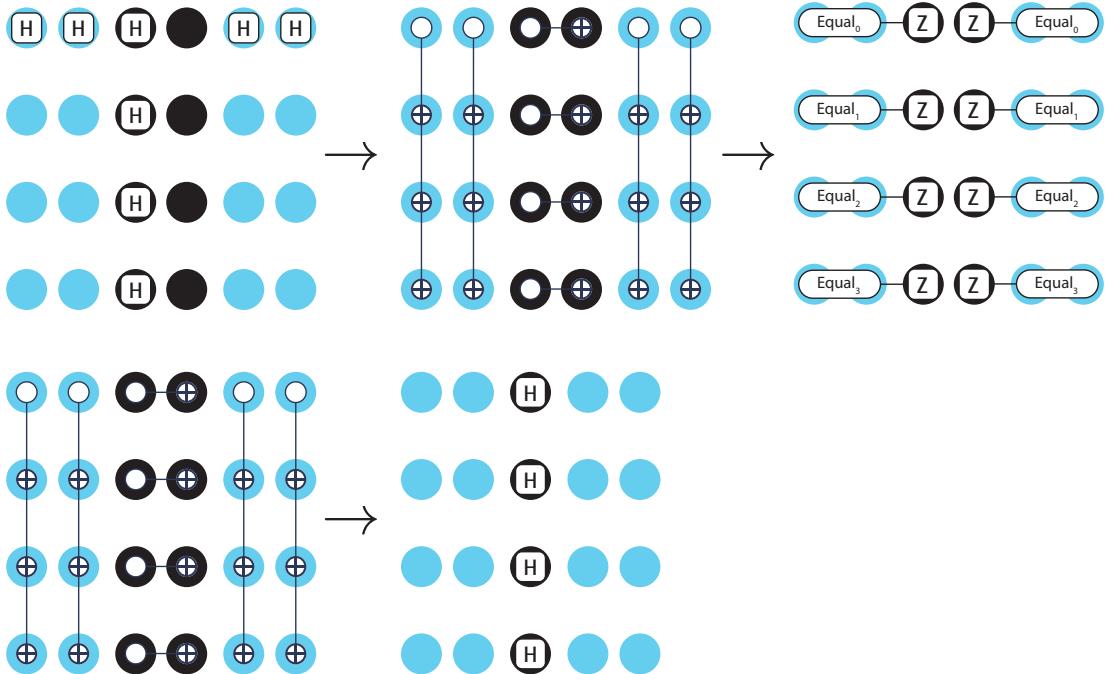


Figure 3.8: Circuit to implement **Filling**, $|0\rangle_{i_1} \ldots |0\rangle_{i_k} |0\rangle_{s_1} \rightarrow \frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle_{i_1} \ldots |j_k\rangle_{i_k} |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle_{s_1}$. This circuit requires $\mathcal{O}(kn \log(n))$ qubits. Here we ilustrate for $n = 4$ and $k = 2$. A grid of 24 qubits is shown: 16 blue index qubits and 8 black system qubits. Each of the five grids represents a single timeslice in the circuit.

**3.4.7.** LEMMA. *An* LAQCC *circuit exists that implements **Filtering**:*

$$\frac{1}{n^{k/2}} \sum_{j_1,\ldots,j_k=0}^{n-1} |j_1\rangle \ldots |j_k\rangle |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle \rightarrow \sqrt{\frac{(n-k)!}{n!}} \sum_{j_1 \neq \ldots \neq j_k}^{n-1} |j_1\rangle \ldots |j_k\rangle |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle.$$

**Proof:**

First note that the state produced by the **Filling** step,

$$\frac{1}{n^{k/2}} \sum_{j_1\dots j_k=0}^{n-1} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle_{s_1},$$

contains states in which some of the indices $j_l$ overlap. Let $|\psi\rangle = \sum_{j_1\neq\dots\neq j_k} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle$, be the state in which none of the indices overlap, the desired output state. Then we can write

$$\frac{1}{n^{k/2}} \sum_{j_1\dots j_k=0}^{n-1} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle_{s_1} = \alpha |\psi\rangle + \beta \left|\psi^\perp\right\rangle,$$

with $\left|\psi^\perp\right\rangle$ containing the states in which at least two of the indices $j_l$ overlap. Note that $\left\langle\psi\middle|\psi^\perp\right\rangle = 0$, so $\alpha$ can be exactly calculated by counting the number of quantum states with distinct $j_i$'s, which gives $|\alpha|^2 = \frac{n!}{(n-k)!n^k}$. Lemma 3.A.2 gives a lower bound on $|\alpha|^2$:

$$|\alpha|^2 = \frac{n!}{(n-k)!n^k} > e^{\frac{-2k^2}{n}},$$

which is at least constant for $k = \mathcal{O}(\sqrt{n})$.

The state $\left|\psi^\perp\right\rangle$ is a superposition of states in which the system register state has Hamming weight less than $k$, because at least two of the $j_i$'s are the same causing a cancellation in the system register. We can use this to create a unitary $U_{flag}$ that flags $\left|\psi^\perp\right\rangle$. We implement this in two steps:

$$\frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle_{s_1} |0\rangle_{\log(k)} |0\rangle$$

$$\xrightarrow{(1)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle_{s_1} \, ||e_{j_1} \oplus \dots \oplus e_{j_k}|\rangle |0\rangle$$

$$\xrightarrow{(2)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle_{s_1} |0\rangle \left|\mathbb{1}_{|e_{j_1}\oplus\dots\oplus e_{j_k}|=k}\right\rangle$$

$$= \alpha |\psi\rangle |1\rangle + \beta \left|\psi^\perp\right\rangle |0\rangle$$

Where $|x|$ denotes the Hamming weight of bitstring $x$. Step (1) follows from a Hamming-weight gate (see Table 3.4), which requires $\mathcal{O}(n \log(n))$ qubits; Step (2) follows from applying an $\text{Exact}_k$ gate, requiring $\mathcal{O}(\log(n)^2)$ qubits. This same step also uncomputes the Hamming-weight gate of the first step.

Lemma 3.3.13 now allows us to amplify $\alpha$ to 1 using the oracle $U_{flag}$. This produces the state

$$\sqrt{\frac{(n-k)!}{(n)!}} \sum_{j_1\neq\dots\neq j_k} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle,$$

using $\mathcal{O}(kn\log(n))$ qubits. $\hfill\square$

To uncompute the index registers, we have to know which one in the system register corresponds to which index register, as any permutation of the index registers results in the same state in the system register. The **Ordering** step imposes an ordering on the index registers, thereby removing the redundancy in the ordering.

**3.4.8.** LEMMA. *An* LAQCC *circuit exists that implements* **Ordering***:*

$$\sqrt{\frac{(n-k)!}{n!}} \sum_{j_1\neq\dots\neq j_k}^{n-1} |j_1\rangle\dots|j_k\rangle\,|e_{j_1}\oplus\dots\oplus e_{j_k}\rangle \rightarrow$$
$$\frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1<\dots<j_k}^{n-1} |j_1\rangle\dots|j_k\rangle\,|e_{j_1}\oplus\dots\oplus e_{j_k}\rangle.$$

*This circuit uses* $\mathcal{O}(k^2\log(n)^2)$ *qubits.*

**Proof:**
The first step of the LAQCC circuit that implements **Ordering** is to evaluate a Greaterthan-gate on all pairs of index registers, which requires $k$ copies of each index register. We require $k$ extra qubits per index register to store the outcome of the Greaterthan-gates. The copies of the index registers are created by doing a fan-out gate. Note that the distribution of the index registers should be set up in such a way that every possible pair can be compared by a Greaterthan-gate.

$$\sqrt{\frac{(n-k)!}{n!}} \sum_{j_1\neq\dots\neq j_k} |j_1\rangle^{\otimes k}|0\rangle^{\otimes k}\dots|j_k\rangle^{\otimes k}|0\rangle^{\otimes k}|e_{j_1}\oplus\dots\oplus e_{j_k}\rangle \xrightarrow{(1)}$$
$$\sqrt{\frac{(n-k)!}{n!}} \sum_{j_1\neq\dots\neq j_k} \left[|j_1\rangle^{\otimes k}|\mathbb{1}_{j_1>j_2}\rangle\dots|\mathbb{1}_{j_1>j_k}\rangle\right]\dots$$
$$\left[|j_k\rangle^{\otimes k}|\mathbb{1}_{j_k>j_1}\rangle\dots\left|\mathbb{1}_{j_k>j_{k-1}}\right\rangle\right]|e_{j_1}\oplus\dots\oplus e_{j_k}\rangle.$$

Each $\mathbb{1}_{j_k>j_{k'}}$ is an indicator variable that evaluates to one if and only if $j_k > j_{k'}$. This step requires $\mathcal{O}(k^2\log(n)^2)$ qubits.

Next, we compute and measure the Hamming weight of the ancilla qubits $|\mathbb{1}_{j_1>j_2}\rangle\dots|\mathbb{1}_{j_1>j_k}\rangle$, using the Hamming-weight gate. We measure the calculated Hamming weights. As all index registers were distinct before measuring, these measurements directly impose an ordering on the index registers.

$$\xrightarrow{\text{(Hammingweight)}} \sqrt{\frac{(n-k)!}{n!}} \sum_{j_1 \neq \dots \neq j_k} \left[ |j_1\rangle \, |\mathbb{1}_{j_1 > j_2} + \dots + \mathbb{1}_{j_1 > j_k}\rangle \right]$$

$$\dots \left[ |j_k\rangle \, \big|\mathbb{1}_{j_k > j_1} + \dots + \mathbb{1}_{j_k > j_{k-1}}\big\rangle \right] |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle$$

$$\xrightarrow{\text{(measure)}} \binom{n}{k}^{-1/2} \sum_{j_1 < \dots < j_k} \left[ |j_1\rangle \, |0\rangle \right] \dots \left[ |j_k\rangle \, |k\rangle \right] |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle$$

This step costs $\mathcal{O}(k^2 \log(k))$ qubits. Assume without loss of generality that the measurement outcomes impose the ordering $j_1 < \dots < j_k$. Otherwise, a permutation of the index registers achieves the same ordering, using the Permutation gate from Table 3.1.

Uncomputing the Hamming weights and the Greaterthan-gates gives the state

$$\binom{n}{k}^{-1/2} \sum_{j_1 < \dots < j_k} \left[ |j_1\rangle \dots |j_k\rangle \right] |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle .$$

$\square$

The **Cleaning** step cleans the index registers for the Dicke state in a similar fashion as in the **Compress** method in the $W$-state protocol. In the cleaning process, we have to take the added ordering of the index registers into account. Suppose the $l$-th qubit of the system register is a 1. If this is the first 1 in the system register, it belongs to index register $j_1$, and if it is the $m$-th 1 it belongs to index register $j_m$. Computing the Hamming weight of the first $l-1$ qubits gives precisely this information. Combined, this shows that if the $l$-th qubit is a 1 and the Hamming weight of the first $l-1$ qubits equals $m$, then the $l$-th qubit should uncompute the $m+1$-th index register $j_{m+1}$.

**3.4.9.** LEMMA. *An* LAQCC *circuit exists that implements **Cleaning**:*

$$\frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \dots < j_k} |j_1\rangle \dots |j_k\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle \rightarrow \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \dots < j_k} |0\rangle \dots |0\rangle \, |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle .$$

*This circuit uses* $\mathcal{O}(n^2 \log(n))$ *qubits.*

**Proof:**
The first step, as described above, is to acquire the Hamming weight from all the substrings of the system register. This requires $n$ copies of the system register as well as a $\log(k)$-qubit register to store the Hamming weight value. The copies

follow from the fanout-gate.

$$
\binom{n}{k}^{-1/2} \sum_{j_1 < \cdots < j_k} |j_1\rangle \ldots |j_k\rangle \, |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle \, |0\rangle_n^{\otimes n-1} \, |0\rangle_{\log(n)}^{\otimes n} \xrightarrow{(1)}
$$

$$
\binom{n}{k}^{-1/2} \sum_{j_1 < \cdots < j_k} |j_1\rangle \ldots |j_k\rangle \, |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle^{\otimes n} \, |0\rangle_{\log(n)}^{\otimes n} \xrightarrow{(2)}
$$

$$
\binom{n}{k}^{-1/2} \sum_{j_1 < \cdots < j_k} |j_1\rangle \ldots |j_k\rangle \, |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle^{\otimes n} \bigotimes_{l=0}^{n-1} \big| |(e_{j_1} \oplus \cdots \oplus e_{j_k})_{[l,n]}| \big\rangle,
$$

where $|(e_{j_1} \oplus \cdots \oplus e_{j_k})_{[l,n]}|$ denotes the Hamming weight of the substring consisting of qubits $l$ up until $n$ of the system register. Step (1) copies the system qubits using fan-out gates; Step (2) computes the Hamming weight of all the qubits 1 up until $j-1$ using the Hammingweight-gate shown in Table 3.4; Step (3) cleans the copies of the system register by applying fan-out. This step is omitted from the equations, but is included in the graphical explanation of the circuit, shown in Figure 3.9 for $n = 4$. Note that at the end of the calculation, it is convenient to teleport the Hamming weight registers next to the system register. There are now $n$ new registers containing the information of the Hamming weight, we will refer to them as the Hamming weight registers. This step requires $\mathcal{O}(n^2 \log(n))$ qubits.
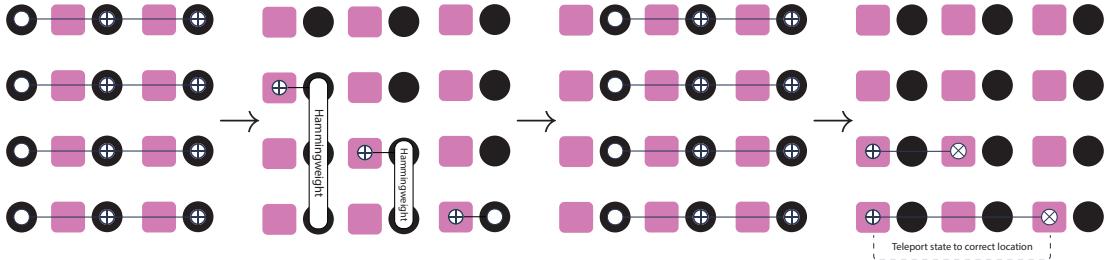


Figure 3.9: Circuit to implement of the Hamming weight calculation of all qubit strings $l$ to $n-1$ in four steps in parallel. The black dots represent system qubits, the pink squares represent $\log(k)$ qubit registers that can count the Hamming weight up until $k$.

The last step that remains is to clean the $k$ index registers. Cleaning the $k$ index registers follows similar steps as the **Compress** method in the $W$-state protocol, with the added Hamming-weight information taken into account. This step requires $k$ copies of the system registers well as $k$ copies of the Hamming-weight registers. Every index register is paired with one copy of the system register and a copy of the $n$ Hamming-weight registers. Cleaning the $j$-th register consists of five steps, similar to the **Compress** method of the $W$-state: Step (1) applies Hadamard gates to bring the index register to phase space, in which

*CNOT*-gates are diagonalized; Step (2) copies the index register; Step (3) uses the information in the Hamming-weight and system register to apply the phases to the correct index register qubits; Step (4) cleans the index register copies; and, Step (5) applies Hadamard gates to reset the index register qubits to the $|0\rangle$ state

Figure 3.10 shows the steps taken to clean a single index register $j$. The black dots represent the qubits in the system register and the upper row of blue dots represent the qubits in index register $j$. The pink squares represent the ancilla Hamming weight register, where each square represents a group of $\log(k)$ qubits. This step requires $\mathcal{O}(nk \log(k) \log(n))$ qubits. At the end of the **Cleaning** operation the state is as desired:

$$\frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \cdots < j_k}^{n-1} |0\rangle \ldots |0\rangle \, |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle.$$

The **Cleaning** step requires $\mathcal{O}(n^2 \log(n))$ qubits. $\qquad\square$
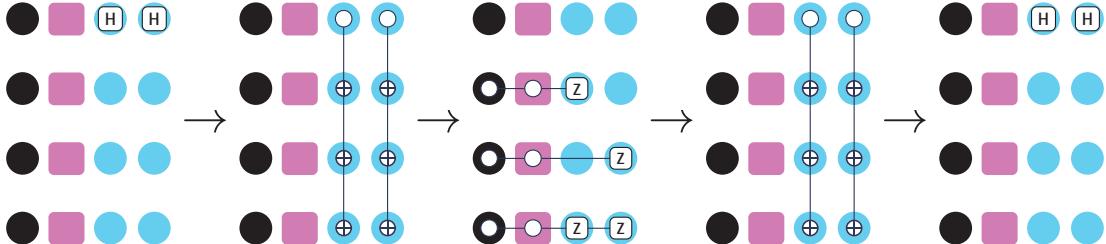


Figure 3.10: Circuit to clean index register $j$. The black dots represent qubits in the system register and the blue dots the index register and its copies. The pink squares represent the ancilla Hamming weight register and its copies. Each pink square represents a group of $\log(k)$ qubits. Each of the five grids represents a single timeslice in the circuit.

**3.4.10.** THEOREM. *For any $n$ and $k = \mathcal{O}(\sqrt{n})$ there exists an* LAQCC *circuit preparing the Dicke-$(n,k)$ state, $|D_k^n\rangle$, using $\mathcal{O}(n^2 \log(n))$ qubits.*

**Proof:**
The circuit combines the circuits resulting from Lemmas 3.4.6, 3.4.7, 3.4.8 and

3.4.9. It consists of four steps:

$$|0\rangle_{i_1} \dots |0\rangle_{i_k} |0\rangle_{s_1} \xrightarrow{(1)} \frac{1}{n^{k/2}} \sum_{j_1,\dots,j_k=0}^{n-1} |j_1\rangle_{i_1} \dots |j_k\rangle_{i_k} |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle_{s_1}$$

$$\xrightarrow{(2)} \sqrt{\frac{(n-k)!}{n!}} \sum_{j_1 \neq \dots \neq j_k}^{n-1} |j_1\rangle \dots |j_k\rangle |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle$$

$$\xrightarrow{(3)} \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \dots < j_k}^{n-1} |j_1\rangle \dots |j_k\rangle |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle$$

$$\xrightarrow{(4)} \frac{1}{\sqrt{\binom{n}{k}}} \sum_{j_1 < \dots < j_k}^{n-1} |0\rangle \dots |0\rangle |e_{j_1} \oplus \dots \oplus e_{j_k}\rangle$$

Step (1) implements **Filling** using Lemma 3.4.6 requiring $\mathcal{O}(kn \log(n) \log(\log(n)))$ qubits; Step (2) implements **Filtering** using Lemma 3.4.7 requiring $\mathcal{O}(kn \log(n))$ qubits; Step (3) implements **Ordering** using Lemma 3.4.8 requiring $\mathcal{O}(k^2 \log(n)^2)$ qubits; Step (4) implements **Cleaning** using Lemma 3.4.9 requiring $\mathcal{O}(n^2 \log(n))$ qubits. After every step ancilla qubits are cleaned, so that they can be reused. As $k = \mathcal{O}(\sqrt{n})$ the largest amount of qubits required for a step is Step (4) requiring $\mathcal{O}(n^2 \log(n))$ qubits. □

Bärtschi and Eidenbenz posed a conjecture on the optimal depth of quantum circuits that prepare the Dicke-$(n,k)$ state. They give an algorithm for generating Dicke-$(n,k)$ states in depth $\mathcal{O}(k \log(\frac{n}{k}))$, given all-to-all connectivity, and conjecture that this scaling is optimal when $k$ is constant. Our result shows that there is a LAQCC implementation in this regime, when one has access to intermediate measurements and feed forward. This does not disprove their conjecture.

### 3.4.4 Dicke states for all $k$ using log-depth quantum circuits

The previous section gave a constant-depth protocol to prepare the Dicke-$(n,k)$ state for $k = \mathcal{O}(\sqrt{n})$. We developed a different method for creating Dicke-$(n,k)$ states which requires logarithmic (in $n$) quantum depth to prepare Dicke-$(n,k)$, but works for arbitrary $k$. We first define what we mean with logarithmic quantum depth:

**3.4.11.** NOTATION. *We let* LAQCC -LOG *refer to the instance* LAQCC$(\mathsf{QNC}^0, \mathsf{NC}^1, \mathcal{O}(\log(n)))$, *similar to* LAQCC *except that we allow for a logarithmic number of alterations between quantum and classical calculations. This results in a circuit of logarithmic quantum depth.*

In this section we show a LAQCC -LOG circuit that creates the Dicke-$(n,k)$ state.

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 of this is the **Uncompress-Compress** method in the $W$-state protocol, that converts numbers from a binary representation to a one-hot representation. Dicke states are a generalization of 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 representation and the factoradic representation. Below we introduce both representations and present quantum circuits that map between the two. Theorem 3.4.21 proves that a LAQCC-LOG circuit can prepare the Dicke-$(n, k)$ state for any $k$.

**Combinatorial number system**

An interesting result showed that any integer $m \geq 0$ can be written as a sum of $k$ binomial coefficients [Bec64]. For fixed $k$, this is even unique as the next lemma shows.

**3.4.12.** 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*

$$m = \binom{c_k}{k} + \binom{c_{k-1}}{k-1} \ldots \binom{c_1}{1} = \sum_{i=1}^{k} \binom{c_i}{i}.$$

This lemma allows for the definition of the combinatorial number representation:

**3.4.13.** DEFINITION. Let $k \in \mathbb{N}$ be a constant. Any integer $m \in \mathbb{N}$ can be represent by a unique string of numbers $(c_k, c_{k-1} \ldots, c_1)$, such that $c_k > c_{k-1} \cdots > c_1 \geq 0$ and $c_k \leq m$. This string is given by the unique decreasing sequence of Lemma 3.4.12. We call this string the *index representation* denoted by $m^{indx(k)}$.

The bit string of $k$ 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 *combinatorial 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 $m^{comb(1)}$. A generalized number conversion is precisely the protocol needed to prepare Dicke states.

A sketch of the protocol would be as follows: given positive integers $k$ and $n$: Create a superposition state

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{i=0}^{\binom{n}{k}-1} |i\rangle |0\rangle \,;$$

Use number conversion to go from label $i$ to $i^{comb(k)}$

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{i=0}^{\binom{n}{k}-1} |i\rangle \left|i^{comb(k)}\right\rangle ;$$

Use number conversion from $i^{comb(k)}$ to $i$ to clean up 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 conversion map from the combinatorial representation to the binary representation is given by Lemma 3.4.12. This calculation requires iterative multiplication and addition, both of which are in $\mathsf{TC}^0$, hence this calculation is in $\mathsf{TC}^0$.

The converse mapping, from binary to combinatorial representation for given $k$, can be achieved by a 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.

### Mapping between factoradic representation and combinatorial number system

A number representation closely related to the combinatorial number representation is the *factoradic representation*. This number system uses factorials instead of binomials to represent numbers.

**3.4.14.** 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 a *factoradic*, or more explicitly an *n-factoradic*. The elements of an $n$-factoradic is called an *n-digit*. An $n$-factoradic $y$ can represent a number $m$ between $0$ and $n! - 1$, in the following way

$$m = \sum_j^n y_j \cdot j!. \tag{3.6}$$

For a given $m \in \{0, \ldots, n! - 1\}$, we call the $n$-factoradic $y$ obeying the equality above, the *factoradic representation of m*. Denote $\text{Fact}(n)$ as the set of all $n$-factoradics.

The following lemma shows that Equation 3.6 is a bijection, showing that the factoradic representation is unique.

**3.4.15.** LEMMA. *For $k \geq 0$ it holds that:*

$$\sum_{i=0}^{k} i \cdot i! = (k+1)! - 1.$$

**Proof:**
Proof by induction.
**BASE STEP**: Let $k$ be 0:

$$0 \cdot 0! = 1! - 1$$

**INDUCTION STEP**: Assume the lemma holds for some $j$, 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. $\qquad\square$

   This identity allows for using factorials as a base for a number system. The next lemma gives a log-space algorithm to convert a factoradic representation to its combinatorial representation.

**3.4.16.** 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:**
The algorithm $\mathcal{A}$ is given $k$ and an $n$-factoradic $y = (y_{n-1}, \ldots, y_0)$. It 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:

$$(\mathcal{A}(y))_{n-j} = y_{n-j}^{comb(k)} = \begin{cases} 1 & \text{if } y_{n-j} < k - H_{>n-j} \\ 0 & \text{otherwise} \end{cases}. \tag{3.7}$$

This conversion requires comparing an $n$-digit with a constant and the Hamming weight of a bitstring. 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, and this can be stored in logarithmic space.

   Now 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 $y_{n-j} \in \{0, \ldots, n-j\}$ that set $y_{n-j}^{comb(k)} = 0$. For the leftmost index $n - j$ such that $y_{n-j}^{comb(k)} = 0$, it holds that $H_{>n-j} = j - 1$, and then there are $n - k$ possible $n$-digits $y_{n-j}$ that set $y_{n-j}^{comb(k)} = 0$. Then, for the second 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 $y_{n-j}$ causing $y_{n-j}^{comb(k)} = 0$. And so forth. This results in $(n-k)!$ different possible choices for the $(n-k)$-many $n$-digits where $y^{comb(k)} = 0$.

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 cause $y_{n-j}^{comb(k)} = 1$. The second leftmost position $n-j$ gives $k-1$ possible choices, and so forth, for a total of $k!$ possible settings of the $k$-many $n$-digits where $y^{comb(k)} = 0$.

Combined, we conclude that, for every $n$-bit string $y^{comb(k)} \in \{0,1\}^n$ of Hamming weight $k$, there are exactly (the same number of) $k!(n-k)!$ $n$-factoradics $y$ such that $\mathcal{A}(y) = y^{comb(k)}$. Hence, a uniformly random $n$-factoradic is mapped by $\mathcal{A}$ to a uniformly random $n$-bit string of Hamming weight $k$, as claimed.    □

This lemma gives a logspace algorithm to convert a uniformly random $n$-factoradic to a uniformly random $n$-bit string of Hamming weight $k$, for any $k$. It is well known that logspace is contained in $\mathsf{TC}^1$, allowing this calculation to be performed in parallel log-depth when one has access to threshold gates [Joh90]. As we saw in Section 3.3.2, we can compute a threshold gate in $\mathsf{LAQCC}$. Hence, an $\mathsf{LAQCC}$-$\mathsf{LOG}$ can perform any $\mathsf{TC}^1$ calculation. We conclude:

**3.4.17.** CONOLLARY. *The following map can be implemented in $\mathsf{LAQCC}$ -$\mathsf{LOG}$.*

$$\frac{1}{\sqrt{n!}} \sum_{y \in Fact(n)} |y\rangle |0\rangle \to \frac{1}{\sqrt{n!}} \sum_{y \in Fact(n)} |y\rangle |\mathcal{A}(y)\rangle.$$

In the next lemma, we show that a $\mathsf{TC}^0$ circuit can implement the inverse of $\mathcal{A}$.

**3.4.18.** LEMMA. *There exists a $\mathsf{TC}^0$ algorithm which, when given an $n$-bit string $y^{comb(k)}$ of Hamming weight $k$, a uniformly-random $k$-factoradic, and a uniformly-random $(n-k)$-factoradic, outputs a uniformly random $n$-factoradic $y$ among those such that $\mathcal{A}(y) = y^{comb(k)}$.*

**Proof:**
The conversion can be done in parallel, generating an $n$-digit for every bit in $y^{comb(k)} = y_{n-1} \ldots y_0 \in \{0,1\}^n$. Recall that we are given as input a uniformly-random $k$-factoradic $O_{k-1}, \ldots, O_0$ and a uniformly-random $(n-k)$-factoradic $Z_{n-k-1}, \ldots, Z_0$.

For every bit position $n-j$, for $1 \le j \le n$, 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 $\mathsf{TC}^0$ [Vol99].

If $y_{n-j}^{comb(k)} = 1$, set $y_{n-j} = O_{k-H_{>n-j}}$. 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 $\mathsf{AC}^0 \subseteq \mathsf{TC}^0$ [Vol99].
$\square$

**3.4.19.** REMARK. The above algorithm establishes a bijection $(y^{comb(k)}, Z, O) \leftrightarrow y$ between triples $(y^{comb(k)}, Z, O)$ with $y^{comb(k)} \in \{0,1\}^n$ of Hamming weight $k$, $Z \in \text{Fact}(n-k)$ and $O \in \text{Fact}(k)$ and $n$-factoradics $y \in \text{Fact}(n)$. Let $(\mathcal{A}(y), \mathcal{Z}(y), \mathcal{O}(y))$ be the image of an $n$-factoradic $y$ under this bijection. The previous lemma shows that one can compute $y$ from $(y^{comb(k)}, Z, O)$ in $\mathsf{TC}^0$. It is not hard to see that the map $(\mathcal{A}(y), y) \mapsto (\mathcal{A}(y), y, \mathcal{Z}(y), \mathcal{O}(y))$ is also in $\mathsf{TC}^0$. Indeed, to find $\mathcal{Z}(y)$ and $\mathcal{O}(y)$, we need only invert the two defining equalities $y_{n-j} = O_{k-H_{>n-j}}$ and $y_{n-j} = k - H_{>n-j} + Z_{n-k-H_{>n-j}}$.

**3.4.20.** COROLLARY. *The following map can be implemented in* LAQCC.

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{y^{comb(k)}} |0\rangle \left|y^{comb(k)}\right\rangle \to \frac{1}{\sqrt{n!}} \sum_{y \in Fact(n)} |y\rangle |\mathcal{A}(y)\rangle$$

*where $y^{comb(k)}$ ranges over all $n$-bit strings of Hamming weight $k$.*

**Proof:**
The transformation consists of three steps:

$$\binom{n}{k}^{-\frac{1}{2}} \sum_{y^{comb(k)}} \left|y^{comb(k)}\right\rangle |0\rangle |0\rangle |0\rangle$$

$$\xrightarrow{(1)} \binom{n}{k}^{-\frac{1}{2}} \sum_{y^{comb(k)}} \left|y^{comb(k)}\right\rangle \left( \bigotimes_{j=0}^{n-k-1} \sum_{i=0}^{j} |i\rangle \right) \left( \bigotimes_{j=0}^{k-1} \sum_{i=0}^{j} |i\rangle \right) |0\rangle$$

$$= \frac{1}{\sqrt{n!}} \sum_{y^{comb(k)}} \left|y^{comb(k)}\right\rangle \left( \sum_{Z \in \text{Fact}(n-k)} |Z\rangle \right) \left( \sum_{O \in \text{Fact}(k)} |O\rangle \right) |0\rangle$$

$$\xrightarrow{(2)} \frac{1}{\sqrt{n!}} \sum_{y \in \text{Fact}(n)} |\mathcal{A}(y)\rangle \left|\hat{Z}(y)\right\rangle \left|\hat{O}(y)\right\rangle |y\rangle$$

$$\xrightarrow{(3)} \frac{1}{\sqrt{n!}} \sum_{y \in \text{Fact}(n)} |\mathcal{A}(y)\rangle |0\rangle |0\rangle |y\rangle$$

Step (1) prepares a uniform superposition over all $n$-factoradics using Theorem 3.4.1. Step (2) is Lemma 3.4.18, and Step (3) follows from Remark 3.4.19. In the above steps we implicitly used that the inverse of the used LAQCC operations are also LAQCC operations. Even though it is unclear if this inverse-property holds in general, it does hold for the considered LAQCC operations. The measurement steps, which might not be reversible, in this algorithm are used to implement fan-out gates. The inverse of a fan-out gate is the fan-out gate itself and hence is contained in LAQCC. $\square$

**3.4.21.** THEOREM. *There exists a* LAQCC *-*LOG*-circuit for preparing Dicke-$(n, k)$ states, for any positive integers $n$ and $k \leq n$, it uses $\mathcal{O}(poly(n))$ qubits.*

**Proof:**
The circuit combines the circuits resulting from Lemma 3.4.16 and Lemma 3.4.18.
    It consists of three steps:

$$
|0\rangle^{\otimes n \log(n)} |0\rangle^{\otimes n} \xrightarrow{(1)} \frac{1}{\sqrt{n!}} \left( \bigotimes_{j=0}^{n-1} \sum_{i=0}^{j} |i\rangle \right) |0\rangle^{\otimes n} = \sum_{y \in \text{Fact}(n)} |y\rangle |0\rangle
$$

$$
\xrightarrow{(2)} \frac{1}{\sqrt{n!}} \sum_{y \in \text{Fact}(n)} |y\rangle |\mathcal{A}(y)\rangle
$$

$$
\xrightarrow{(3)} \binom{n}{k}^{-\frac{1}{2}} \sum_{y \in \text{Fact}(n)} |0\rangle |\mathcal{A}(y)\rangle = |D_k^n\rangle .
$$

Step (1) prepares a uniform superposition over all $n$-factoradics using Theorem 3.4.1; Step (2) is by Corollary 3.4.17; and, Step (3) reverses the algorithm of Corollary 3.4.20. $\qquad \square$

## 3.4.5   Quantum many-body scar states

There is a particular set of states in many-body physics, called many-body scar states, which are highly excited states that exhibit atypically low entanglement [Tur+18]. These states exhibit long coherence times relative to other states at the same energy density and seem to avoid thermalization and thereby they do not follow the eigenstate thermalization hypothesis. This makes studying the lifetime of quantum many body scar states under perturbations particularly interesting. Studying this lifetime is quite challenging, as even though scarred eigenstates often have modest entanglement and therefore have efficient matrix product state representations, perturbations typically couple them to states nearby in energy which typically have volume-law scaling entanglement, making classical simulations difficult.

An overview paper by Gustafson et al. studied methods of preparing quantum many-body scar states on quantum computers, with the goal to simulate time dynamics directly on the quantum system [Gus+23]. They found several approaches for generating quantum many-body scars for a particular model, which require polynomial depth. They look at quantum many-body scar states of the $n$-qubit spin-1/2 Hamiltonian of [IS20]:

$$
H = \lambda \sum_{i=2}^{n-1} (X_i - Z_{i-1} X_i Z_{i+1}) + \Delta \sum_{i=1}^{n} Z_i + J \sum_{i=1}^{n-1} Z_i Z_{i+1}
$$

The quantum many-body scar states of $n$-qubits are given by:

$$|S_k\rangle = \frac{1}{k!\sqrt{\mathcal{N}(n,k)}}(Q^\dagger)^k |\Omega\rangle,$$

where $\mathcal{N}(n,k) = \binom{n-k-1}{k}$, $|\Omega\rangle = |0\rangle^{\otimes n}$ and $k = 0, \ldots, n/2 - 1$. The raising operator $Q^\dagger$ is given by:

$$Q^\dagger = \sum_{i=2}^{n-1} (-1)^i P_{i-1} \sigma_i^+ P_{i+1},$$

with $P_j = |0\rangle\langle 0|$ and $\sigma_j^+ = X_j + Y_j$. They show that up to local $Z$ gates these states are very closely related to Dicke states:

$$\prod_{i \text{ odd}} Z_i |S_k\rangle = |0\rangle \otimes P_{fib}|D_k^n\rangle \otimes |0\rangle,$$

where $P_{fib}$ is known as the Fibonacci constraint, which is a projector that removes all states where there are two ones next to each other:

$$P_{fib} = I - \sum_{i=1}^{n-1} |11\rangle\langle 11|_{i,i+1}.$$

The goal of this section is to show that these states, for $k = \mathcal{O}(\sqrt{n})$ are accessible in LAQCC. First note that by Theorem 3.4.10 there exists a LAQCC protocol to generate $|D_k^n\rangle$ up to $k = \mathcal{O}(\sqrt{n})$. We will show that there is a LAQCC protocol that applies $P_{fib}$ to these $|D_k^n\rangle$ states. Here we write $P_{fib}|D_k^n\rangle = |F_k^n\rangle$ as the part of the state that adheres to the Fibonacci constraint, and $(I - P_{fib})|D_k^n\rangle = |\overline{F_k^n}\rangle$ as the part of the state that does not. Note that both $|F_k^n\rangle$ and $|\overline{F_k^n}\rangle$ are not normalized, therefore $|D_k^n\rangle = \alpha|F_k^n\rangle + \sqrt{1-\alpha^2}|\overline{F_k^n}\rangle$ for some real $\alpha$ dependent on $k$ and $n$. The first step will be to show that there exists a unitary accessible in LAQCC that flags the state $|F_k^n\rangle$.

**3.4.22.** LEMMA. *There exists a unitary $U_{fib}$, accessible in* LAQCC, *that flags all the states that obey the Fibonacci constraint, more precisely:*

$$U_{fib}|D_k^n\rangle|0\rangle = \alpha|F_k^n\rangle|0\rangle + \sqrt{1-\alpha^2}|\overline{F_k^n}\rangle|1\rangle$$

**Proof:**
Add $n$ extra qubits prepared in $|0\rangle$, one for every sequential pair of qubits. For all $i \in \{1, \ldots, n-1\}$, apply a Toffoli gate with control qubits $i$ and $i+1$ and target qubit the $i$-th auxiliary qubit. The second step is to apply the $OR_n$ gate on the $n$ auxiliary qubits. The $n$-th auxiliary qubit is also used as the output qubit. Clean the extra qubits by again applying Toffoli gates. These steps are accessible in LAQCC therefore implements the flag unitary with an LAQCC protocol. □

The second step is to show that $\alpha$ is bounded by a constant in the case that $k = \mathcal{O}(\sqrt{n})$, this is shown using the following two lemma's.

**3.4.23.** LEMMA. *The total number of bitstrings of length n with k ones, such that no two ones are adjacent is given by:*

$$\binom{n-k}{k} + \binom{n-k-1}{k-1}$$

**Proof:**
We can count the number of possible bitstrings, after first noticing that every 1 must be followed by a 0, unless the last bit is a 1. As a result, we have two situations, in the first, we can consider all possible rearrangements $n-k$ elements, consisting of $k$ pairs '10' and $n-2k$ ones. This gives $\binom{n-k}{k}$ possible bitstrings. In the second situation, the last bit is 1. This leaves $k-1$ pairs '10' in a total of $n-k-1$ elements. With the same reasoning, this gives $\binom{n-k-1}{k-1}$ possible bitstrings. Summing the two situations proves the lemma.                    $\square$

We now consider the relative fraction of this type of bitstrings among all possible bitstrings with Hamming-weight $k$.

**3.4.24.** LEMMA. *Let $k = c\sqrt{n}$ for some constant $c > 0$. Then the following inequality holds*

$$\frac{\binom{n-k}{k} + \binom{n-k-1}{k-1}}{\binom{n}{k}} \geq \exp\left(-c^2\right)$$

**Proof:**
We have

$$\frac{\binom{n-k}{k} + \binom{n-k-1}{k-1}}{\binom{n}{k}} > \frac{\binom{n-k}{k}}{\binom{n}{k}} = \frac{(n-k)!/k!(n-2k)!)}{n!/k!(n-k!)} = \frac{(n-k)!^2}{n!(n-2k)!}.$$

Expanding the factorials and only consider the terms that do not cancel, we obtain

$$\frac{(n-k)!}{n!}\frac{(n-k)!}{(n-2k)!} = \frac{(n-k)(n-k-1)\dots(n-(2k-1))}{n(n-1)\dots(n-(k-1))}.$$

Both the numerator and denominator have $K$ terms, which we can pair. next we note that $\frac{a}{b} > \frac{a-1}{b-1}$ whenever $b > a$ (and $b \notin \{0,1\}$). Using this idea, we obtain the following expression:

$$\frac{n-k}{n}\frac{n-k-1}{n-1}\dots\frac{n-(2k-1)}{n-(k-1)} > (\frac{n-k}{n})^k = (1 - \frac{k}{n})^k.$$

Now using that $k = c\sqrt{n}$, we have

$$\left(1 - \frac{c\sqrt{n}}{n}\right)^{c\sqrt{n}} = \left(\left(1 - \frac{c}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{c}}\right)^{c^2} > \exp\left(-c^2\right),$$

which is a constant. □

This allows us to construct the state $|S_k\rangle$ using the steps for the Dicke-state preparation together with Lemma 3.3.13. Note that Lemma 3.3.13 requires us to implement both $U$ and $U^\dagger$, where $U$ implements the initial superposition. The **Ordering** step (Lemma 3.4.8) contains measurements, therefore, it is not obvious how to implement $U^\dagger$. Still, we can work around this, by applying Lemma 3.3.13 between the **Filtering** and **Ordering** step:

**3.4.25.** THEOREM. *For any $n$ and $k = \mathcal{O}(\sqrt{n})$ there exists a* LAQCC *circuit preparing the many-body scar state $|S_k\rangle$, using $\mathcal{O}(n^2 \log(n))$ qubits.*

**Proof:**
We follow the same steps as for the Dicke-state preparation (see Theorem 3.4.10). After the second **Filtering** step however, we apply the unitary $U_{fib}$ together with Lemma 3.3.13 to filter out all states with subsequent ones in the state. Note that by Lemma 3.4.24, the number of states with no consecutive ones is a constant fraction of the total number of strings of Hamming-weight $k$.

Furthermore, the state after the Filtering step,

$$\sqrt{\frac{(n-k)!}{(n)!}} \sum_{j_1 \neq \ldots \neq j_k} |j_1\rangle \ldots |j_k\rangle \, |e_{j_1} \oplus \cdots \oplus e_{j_k}\rangle \,,$$

is still entangled with the index registers. In effect there are many copies of the Dicke-$(n, k)$ state on the system register, each with a different ordering of the index registers, however, this does not affect the fraction of states with no consecutive ones compared to the states with consecutive ones. Next, the **Ordering** and **Cleaning** steps of the protocol work similarly on the resulting state and will give the state $|S_k\rangle$. □

This shows that the states $|S_k\rangle$ can be prepared in LAQCC for $k = \mathcal{O}(\sqrt{n})$, however it does not say anything about larger $k$. The method described above is vitally dependent on the fact that $\alpha$ is a constant when $k$ is at most $\mathcal{O}(\sqrt{n})$ and as far as we know can not be extended beyond this situation. However, this is not to say that this gives a proof that it is not possible for larger $k$. We simply do not know how to extend this technique to include $k = \omega(\sqrt{n})$, and we suspect that other techniques are required to find an algorithm in LAQCC preparing this state for larger $k$.

## 3.5 Open problems

Finally, we highlight several open problems that suggest promising directions for further investigation of the LAQCC model

**Comparing LAQCC and QNC₁.** As mentioned in Section 3.3.4 it seems natural that LAQCC would be bounded by $\mathsf{QNC}_1$, since the model augments constant-depth quantum computation with intermediate $\mathsf{NC}_1$ classical computations. However, as it turns out, it is not very clear that this is true. It would be interesting to find an example where $\mathsf{QNC}_1$ and LAQCC differ, or to show containment of LAQCC.

**Bounding LAQCC.** So far, the only known upper bound on the LAQCC model is BQP. While it seems highly unlikely that LAQCC captures the entirety of Qpoly, no explicit examples are known of states that can be realized within Qpoly but not in LAQCC. Identifying such states—and characterizing their properties—would provide valuable insight into the precise computational power of measurement and feed-forward.

**Optimizing our constructions.** Although our constructions have constant depth, they are unlikely to be optimal. In particular, for the preparation of the Dicke-$(n, k)$ state, the circuit does not scale with $n$ or $k$, but the constant overhead is nevertheless very large. It would be interesting to investigate whether these constructions can be further optimized. One possible direction is to relax the requirement of exact state preparation and instead allow for small infidelity. For example, [PSC24] proposes novel protocols for preparing both the $W$ state and Dicke-$(n, k)$ states with $k = \mathcal{O}(1)$, where a small infidelity is tolerated. This relaxation leads to significantly simpler protocols.

**Finding LAQCC circuits for other types of states.** Perhaps the most compelling direction is to develop new protocols for state preparation within LAQCC. By pursuing entirely different approaches, one might discover constant-depth LAQCC protocols for Dicke-$(n, k)$ states that extend beyond the current restriction of $k = \mathcal{O}(\sqrt{n})$ Another particularly interesting target, closely connected to recently suggested protocols for many-body scar states, is the Bethe ansatz ground state [Sop+22]. The most interesting regime of the Bethe ansatz state is at half filling, $k = \frac{n}{2}$, however, as far as known to the authors, there exists no deterministic efficient method of preparing these states, even though they have relatively low entanglement similar to the $|S_k\rangle$ states. It would be interesting to see if in an easier regime, maybe $k = \mathcal{O}(\sqrt{n})$, one can find a LAQCC circuit for preparing these types of states.

**Complexity of LAQCC\*:** Another interesting direction for future work is to further investigate the inclusion: $\mathsf{StateLAQCC}^{*\mathcal{O}}_{\varepsilon} \subseteq \mathsf{StatePostQPoly}^{\mathcal{O}}_{\varepsilon}$. Maybe one can find an oracle separation between the two classes. One approach is to use a similar oracle as used in [AK07] to separate QMA and QCMA with

respect to an oracle and use a counting argument to argue that $\mathsf{StateLAQCC}_\varepsilon^{*\mathcal{O}} \neq$ $\mathsf{StatePostQPoly}_\varepsilon^{\mathcal{O}}$, for some oracle $\mathcal{O}$ and $\varepsilon = 1 - \frac{1}{\mathrm{poly}(n)}$.

**One round vs multiple rounds of measurements.** The last interesting direction for studying the state complexity of $\mathsf{LAQCC}$ is in the fact that the $\mathsf{LAQCC}$ model allows for a constant number, more than one, of rounds of measurements and corrections. This was required for our three new state generation protocols. However other models considered only one round of measurements and corrections, for instance in the paper [PSC21]. One may wonder if there is a hierarchy of model power allowing one or multiple measurements, and if there is a way to reduce the number of measurements rounds. A starting effort towards classifying types of states based on such a hierarchy can be found in [TVV23b]. It would be interesting to see a more extensive complexity theoretic analysis comparing models with different number of allowed rounds.

## 3.A    Useful lemmas

This section gives two lemmas. The first lemma upper bounds the computational power of $\mathsf{LAQCC}$. The second lemma helps in preparing Dicke states for $k \in \mathcal{O}(\sqrt{n})$.

**3.A.1.** LEMMA. *Let $\Pi = (\Pi_{yes}, \Pi_{no})$ be a decision problem in $\mathsf{NC}^1$. Then there is a uniform log-depth quantum circuit that decides on $\Pi$.*

**Proof:**
Let $B$ be the uniform Boolean circuit of logarithmic depth deciding on $\Pi$. As $\Pi \in \mathsf{NC}^1$, such a circuit exists.

For fixed input size $n$, write $B$ as a Boolean tree of depth $\mathcal{O}(\log(n))$, with at its leaves the $n$ input bits $x_i$ and as root an output bit. This Boolean tree directly translates in a classical circuit using layers of AND, OR and NOT gates.

Each of these gates has a direct quantum equivalent gate, provided that we use ancilla qubits: First replace all OR gates by AND and NOT gates. Then replace all AND gates by Tofolli gates, which has three inputs. The third input is a clean ancilla qubit and will store the AND of the other two inputs. Finally, replace all NOT gates by $X$-gates. $\qquad\square$

**3.A.2.** LEMMA. *Let $n, k \in \mathbb{N}$ and $k < \frac{n}{2}$ then:*

$$\frac{n!}{n^k(n-k)!} > e^{-\frac{2k^2}{n}}$$

**Proof:**
The result follows by a simple computation

$$\frac{n!}{n^k(n-k)!} = e^{\sum_{i=1}^{k} \log(1-\frac{i}{n})}$$

$$> e^{\sum_{i=1}^{k} \frac{-i}{n-i}}$$

$$> e^{\sum_{i=1}^{k} \frac{-i}{n-k}}$$

$$> e^{-\frac{k^2}{n-k}}$$

$$> e^{-\frac{2k^2}{n}},$$

where we use that $\log(1+x) \geq \frac{x}{1+x}$ for $x > -1$.

□

# 3.B    Gate implementations

## 3.B.1    OR-gate

In this section we discuss the implementation of the $OR_n$-gate and why we can implement it using local gates in a nearest-neighbor architecture. We show how the gate works for any basis state $|x\rangle = |x_1\rangle \otimes \ldots \otimes |x_n\rangle$. By linearity, the gate then works for arbitrary superpositions. The OR-gate by Takahashi and Tani consists of two steps: First they apply the OR-reduction introduced in Ref. [HŠ05], which prepares a state on $\log n$ qubits such that the OR evaluated on these $\log n$ qubits yields the same result as the OR evaluated on the original $n$ qubits. Second, they evaluate an exponential circuit on these $\log n$ qubits to calculate the OR-gate. This results in a polynomial-sized circuit for the OR-gate.

Let $m = \lceil \log_2(n+1) \rceil$, then the OR-reduction implements the map

$$|x\rangle |0\rangle^{\otimes m} \rightarrow |x\rangle \bigotimes_{j=1}^{m} \left| \mu_{\phi_j}^{|x|} \right\rangle,$$

where $\varphi_j = \frac{2\pi}{2^j}$ and $\left| \mu_{\varphi}^{|x|} \right\rangle = HR_Z(\varphi \cdot |x|)H |0\rangle$. The OR-reduced state thus depends on the weighted Hamming weight of $x$, more precisely, every $\left| \mu_{\phi_j}^{|x|} \right\rangle$ depends on the entire string $x$. For every $\left| \mu_{\phi_j}^{|x|} \right\rangle$ this requires a copy of $|x\rangle$ which can be created by using the fanout gate. The circuit applying the rotation $R_Z(\varphi \cdot |x|)$ consists of sequential $R_Z$ gates, which are diagonal and hence can be implemented in parallel by Lemma 3.3.11. This results in a circuit of width $\mathcal{O}(n \log(n))$ for creating the OR-reduced state.

We have for every bitstring $x \in \{0,1\}^n$ that

$$\mathsf{OR}_n(x) = \frac{1}{2^{n-1}} \sum_{a \in \{0,1\}^n \setminus \{0^n\}} \mathrm{PA}_n^a(x),$$

where $\mathrm{PA}_n^a(x) = \oplus_{j=0}^{n-1} a_i x_i$ is the parity of $x$, weighted by the non-zero binary vector $a$. Hence, computing the OR of the input is now reduced to computing all parities of the subsets of the inputs. The parity gate is equivalent to a fanout-gate conjugated by Hadamard gates on every qubit, see also Table 3.1.

We now copy the $m$ qubits of the OR-reduced state $2^m = n$ times, using fanout gates. For each copy we require two additional auxiliary qubits. The first will hold the result of the parity computation, for which we already have a nearest-neighbor implementation. We will entangle the second auxiliary qubit using a fanout-gate to obtain a GHZ-state in these qubits.

We now compute in parallel the parity of the subsets of the inputs. For every subset we use a single copy of the inputs and we store the result in the corresponding auxiliary qubit. We then apply a controlled-$R_Z$ gate from the first auxiliary qubit to the second auxiliary qubit in the GHZ state. This prepared the state (omitting other registers)

$$\frac{1}{\sqrt{2}}(|0\rangle^{n-1} + (-1)^{\frac{1}{2^{n-1}} \sum_{a \in \{0,1\}^m \setminus \{0^m\}} \mathrm{PA}_m^a(x)} |1\rangle^{\otimes n-1}) = \frac{1}{\sqrt{2}}(|0\rangle^{n-1} + (-1)^{\mathsf{OR}_n(x)} |1\rangle^{\otimes n-1}).$$

Uncomputing this final state using a fanout-gate gives a single qubit that holds the desired answer in its phase. A single Hadamard gate applied to this qubit will then give the answer in a single qubit.

Combining all steps thus gives an implementation for the OR gate using a geometrically local nearest-neighbor circuit. For more details on the implementation as well as a proof of correctness, we refer to the original proof [TT13].

## 3.B.2 Equality-gate

Define the Equality gate on two $n$-qubit computational basis states as

$$\text{Equality} : |x\rangle |y\rangle |0\rangle \mapsto |x\rangle |y\rangle |\mathbb{1}_{x=y}\rangle.$$

This gate is implemented in three steps: (1) subtract the first register from the second using a subtraction circuit; (2) apply $\mathrm{Equal}_0$ on the second register and store the result in the third register; (3) add the first register to the second, undoing the subtraction computation:

$$|x\rangle |y\rangle |0\rangle \xrightarrow[(1)]{} |x\rangle |y-x\rangle |0\rangle$$

$$\xrightarrow[(2)]{} |x\rangle |y-x\rangle |\mathbb{1}_{x=y}\rangle$$

$$\xrightarrow[(3)]{} |x\rangle |y\rangle |\mathbb{1}_{x=y}\rangle$$

Addition and subtraction both have width $\mathcal{O}(n^2)$, which, as a result, the Equality-gate also has.

### 3.B.3   Greaterthan-gate

Define the Greaterthan gate on two $n$-qubit computational basis states as

$$\text{Greaterthan} : |x\rangle |y\rangle |0\rangle \mapsto |x\rangle |y\rangle |\mathbb{1}_{x>y}\rangle .$$

This gate is implemented in four step: (1) Add an extra clean qubit to the second register and interpret this as an $n+1$-qubit register with most significant bit zero; (2) subtract the first register from the second. The subtraction is modulo $2^{n+1}$; (3) apply a CNOT-gate from most significant bit of the second register to the third register; (4) add the first register to the second, undoing the subtraction computation:

$$
\begin{aligned}
|x\rangle |y\rangle |0\rangle \xrightarrow[(1)]{} & |x\rangle |0y\rangle |0\rangle \\
\xrightarrow[(2)]{} & |x\rangle \left|y - x \bmod 2^{n+1}\right\rangle |0\rangle \\
\xrightarrow[(2)]{} & |x\rangle \left|y - x \bmod 2^{n+1}\right\rangle |\mathbb{1}_{x>y}\rangle \\
\xrightarrow[(3)]{} & |x\rangle |0\rangle |y\rangle |\mathbb{1}_{x>y}\rangle
\end{aligned}
$$

This construction works, as after step (2), the most significant bit of the second register is one, precisely if $x$ is larger than $y$.

Addition and subtraction both have width $\mathcal{O}(n^2)$, which, as a result, the Greaterthan-gate also has.

### 3.B.4   Exact$_t$-gate

Define the Exact$_t$ gate on an $n$-qubit computational basis state as

$$\text{Exact}_t : |x\rangle |0\rangle \mapsto |x\rangle \left|\mathbb{1}_{|x|=t}\right\rangle .$$

Here, $|x|$ denotes the Hamming weight of the $n$-bit string $x$.

This gate follows by combining the Hammingweight-gate and the Equality-gate: First, compute the Hamming weight of $x$ in an ancilla register and then apply the Equality gate to check that this ancilla register equals $t$.

Another approach is to modify the circuit for $OR$ slightly. In the $OR$-reduction step, add a gate $R_Z(-\varphi t)$, which adjusts the angle to be zero precisely if $|x| = t$ (see Theorem 4.6 of [HŠ05]). Then apply the circuit for $OR$ and negate the output. The circuit for $OR$ evaluates to zero, precisely if the input had Hamming weight $t$.

## 3.B.5 Threshold$_t$-gate

Define the Threshold$_t$ gate on an $n$-qubit computational basis state as

$$\text{Threshold}_t : |x\rangle |0\rangle \mapsto |x\rangle \left| \mathbb{1}_{|x| \geq t} \right\rangle .$$

Taking the $OR$ over the outputs of Exact$_j$-gates for all $j \geq t$, gives the Threshold$_t$-gate. An improved implementation with better scaling in $t$ is given in Theorem 2 of [TT13].

A weighted threshold gate uses weights $w_i$ and evaluates to one precisely if $\sum_i w_i x_i \geq t$. Assume without loss of generality that $\sum_i w_i x_i$ evaluates to an integer. Otherwise, we can use the same ideas, but up to some precision.

Use the same $OR$-reduction as for the normal threshold gate. Instead of rotations $R_Z(\varphi)$ controlled by $x_i$, we use rotations $R_Z(w_i\varphi)$ controlled by $x_i$. This implements the weighted threshold gate.

# Part Two

# Catalysis of space

# Chapter 4

# Catalytic computation

In this part of the thesis, we explore the field of *space-bounded complexity*. Here, the central question is not how long it takes to solve a problem, but whether it can be solved using only a limited amount of memory. This is of particular interest in near-term quantum computing, as the number of physically achievable qubits is still very limited, especially in the near-term error-corrected regime. The second resource we introduce for boosting the power of these devices is aimed at reducing the space required to run quantum algorithms. This resource is the presence of a *catalyst*.

A well-known tool in chemistry to facilitate chemical reactions is a catalyst, a substance that speeds up the chemical reaction or lowers the required energy without being altered by the reaction. It can enable processes that, without it, might be too slow, inefficient, or even impossible, all while being restored to its original state at the end of the chemical reaction. The concept of a catalyst is not limited to chemistry and can be found in the field of quantum information theory. There, catalysts in the form of entangled states are widely recognized for their counterintuitive abilities to enable (state) transformations that are otherwise infeasible (see survey by Lipka et al. [LWN24]). A related concept, known as catalytic embedding, was recently introduced in the context of circuit synthesis as an alternative to traditional gate approximation methods in quantum circuit design [Amy+23].

In classical space-bounded computation, Buhrman, Cleve, Koucký, Loff, and Speelman [Buh+14] introduced the notion of *catalytic memory*, leading to the computational model now known as *catalytic computing*. The central idea is that a large memory resource, containing arbitrary data, may be made available to a computation, provided that this memory is returned to its original state at the end of the process. This model has proven valuable for reducing the amount of additional working space required in classical computations.

In the quantum setting, we ask whether an analogous notion of *quantum catalytic memory* could enable similar space reductions. Establishing such a model

could reveal new trade-offs between space and time in quantum algorithms.

In this chapter, we introduce the theory of classical catalytic computing and highlight the key questions it raises for quantum analogues. These questions will guide the investigations of the following two chapters.

## 4.1 The catalytic computing model

Catalytic computing was originally introduced by Buhrman, Cleve, Koucký, Loff, and Speelman [Buh+14], and discusses the following idea: What if one gives a space-bounded machine access to a hard drive that is already full with information. During the computation this machine can change the information on the hard drive as needed, as long as at the end of the calculation the original information is perfectly restored. In this way the additional full hard-drive is used as a catalyst, giving power to the computation without being altered. A graphic representation of this idea can be found in Figure 4.1



Clean space: s

Hard drive as
Catalytic space: c

Alter data on catalytic space
during calculation

Reset the the data on the
harddrive at the end, using the
hard drive as a **catalyst**
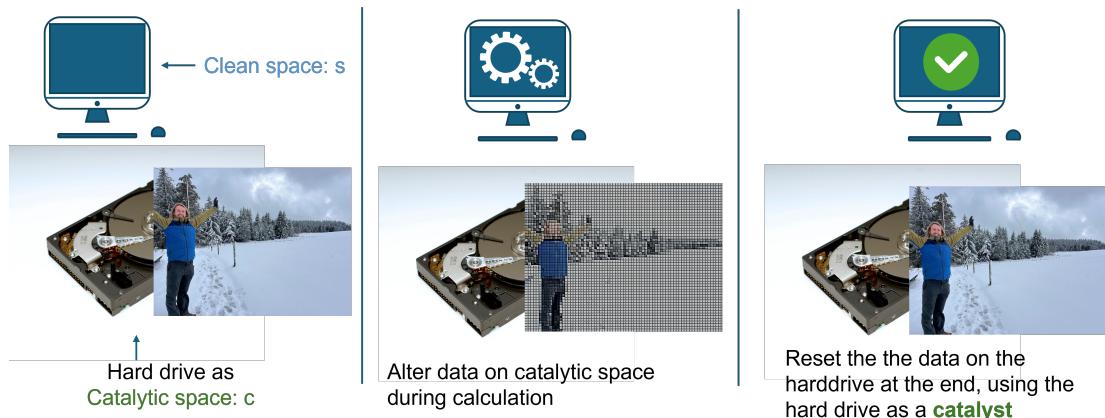
Figure 4.1: An explanatory graphic illustrating catalytic computing. The space-bounded machine (the "clean space": s) is represented by the computer, while the catalyst is depicted as a hard drive containing holiday images. During computation, the machine may temporarily alter these images, but by the end, the hard drive is restored exactly to its original state.

It is not immediately obvious that such an approach is useful at all, as the original contents of the hard drive need to be perfectly recovered. Even worse, this has to happen for any possible instantiation of the hard drive.

To formalize this idea, Buhrman et al. [Buh+14] introduced the *catalytic Turing machine*, which models a computation with both limited clean space and an auxiliary catalytic tape.

**4.1.1.** DEFINITION (Catalytic Turing Machine). A *catalytic Turing Machine* is a space-bounded Turing machine with two work tapes: 1) a read-write work tape

of length $s(n)$ which is initialized to $0^{s(n)}$, and 2) a read-write *catalytic tape* of length $c(n) \leq 2^{s(n)}$ which is initialized to an arbitrary state $\tau \in \{0,1\}^{c(n)}$. On any input $x \in \{0,1\}^n$ and initial catalytic state $\tau$, a catalytic Turing machine has the property that at the end of the computation, the catalytic tape will be in the initial state $\tau$.

With the catalytic Turing machine defined, we can formally introduce the corresponding complexity class, *catalytic space*.

**4.1.2.** DEFINITION (Catalytic space). We write $\mathsf{CSPACE}[s, c]$ as the class of languages that can be recognized by catalytic Turing machines with work space $s := s(n)$ and catalytic space $c := c(n)$.

The main object being studied in [Buh+14] is a more specific parameter setting of catalytic space, which they call *catalytic log-space*:

**4.1.3.** NOTATION. *In the specific setting where $s(n) = \mathcal{O}(\log(n))$ and $c(n) = \mathcal{O}(poly\, n)$, we call the catalytic space class: catalytic log-space, and we write* $\mathsf{CL} := \mathsf{CSPACE}[\mathcal{O}(\log n), poly\, n]$.

Surprisingly, [Buh+14] show that $\mathsf{CL}$ can be much more powerful than $\mathsf{L}$, with the catalytic tape being at least as powerful a resource as non-determinism ($\mathsf{NL} \subseteq \mathsf{CL}$), randomness ($\mathsf{BPL} \subseteq \mathsf{CL}$), and more ($\mathsf{TC}^1 \subseteq \mathsf{CL}$). They also showed that its power is nevertheless limited and falls far short of $\mathsf{PSPACE}$, namely $\mathsf{CL} \subseteq \mathsf{ZPP}$.

Since the original introduction, many works have studied classical catalytic space from a variety of angles, including augmenting catalytic machines with other resources such as randomness or non-determinism [Buh+18; Dat+20; Coo+25; Kou+25], considering non-uniform models such as catalytic branching programs or catalytic communication complexity [Pot17; CM22], analyzing the robustness of classical catalytic machines to alternate conditions [BDS22; Bis+24; Gup+24], and so on.

Perhaps most important, the utility of classical catalytic computation has been strikingly demonstrated in its use as a subroutine in an ordinary space-bounded computation: avoiding linear blowups in space when solving many instances of a problem. The most impactful result is the Tree Evaluation algorithm of Cook and Mertz [CM24], which was the key piece in Williams' recent breakthrough on time and space [Wil25].

## 4.2 Catalysis in quantum computation

A natural question is to unify the catalytic model with the standard quantum computational framework. In this part of the thesis, we take a first step toward that goal by initiating the study of catalytic techniques in the quantum setting. The classical catalytic model enjoys several properties that appear particularly

promising for quantum computation, including reversibility[Dul15; Coo+25] and average-case runtime bounds[Buh+14]. This raises a central question:

> How does the computational power of the catalytic model change when it is allowed to perform quantum operations?

An even more pressing motivation comes from the severe space constraints in quantum computation. In the classical setting, catalytic subroutines have led to remarkable reductions in the space required for certain computations[Wil25; CM24]. If analogous space-saving techniques can be developed in the quantum setting, they could have a significant impact, particularly for the feasibility of computations on near-term quantum devices.

## 4.3   Organization of this part

This part of the thesis contains two chapters. In the first chapter, we will take a detour from quantum algorithms and discuss classical catalytic computing in the presence of errors. In this section we give a full classification of the power of the class of *lossy catalytic space*, a model in which one is not required to reset the catalytic space perfectly, but instead up to some small error, in the form of limited number of bit flips.

In the second chapter, we introduce quantum catalytic space. We discuss the relation between quantum catalytic space and its classical counterpart, and furthermore we show a relation between quantum catalytic computing and the one clean qubit model, finishing with showing that classical catalytic computing is contained in the one clean qubit model.

# Chapter 5

## Lossy catalytic computation

In this chapter, we deviate slightly from our main narrative of aiding quantum computations and focus instead on a problem in classical catalytic computation—specifically, the *robustness* of the catalytic model. This problem is inspired by our broader efforts to connect quantum and catalytic computation.

A natural issue that arises when working with quantum computers is the presence of errors. Even when these errors are not due to imperfections in hardware, they can still occur—for example, from approximating a unitary using a finite gate set, or from measurements performed during a calculation. These errors can disturb the catalyst, making it impossible to recover its original state. The original definition from [Buh+14] is not robust to such errors—it assumes perfect resetting of the catalytic tape. Therefore, there is a natural question to ask:

*What does a model of catalytic computing look like in the presence of error?*

This question was first posed by Gupta et al. [Gup+24]. They initiated the study of *lossy* catalytic computing, a model of catalytic computation wherein the catalytic tape need not be exactly restored to its initial configuration at the end of the calculation. This model, which we refer to as LCSPACE, essentially asks how robust the core definition of catalytic space is to seemingly small relaxations. They start by giving a definition of a *lossy catalytic Turing machine*:

**5.0.1.** DEFINITION (Lossy catalytic Turing machines). A *lossy catalytic Turing machine with $e(n)$ errors* is a catalytic machine where at the end of the computation on any input $x \in \{0,1\}^n$ and initial catalytic state $\tau$, instead of requiring that the catalytic tape be in state $\tau$, the catalytic tape can be in any state $\tau'$ such that $\tau$ and $\tau'$ differ in at most $e(n)$ locations, i.e. $\Delta(\tau, \tau') \leq e(n)$.

This allows for the definition of LCSPACE:

**5.0.2.** DEFINITION (Lossy catalytic space). We write LCSPACE$[s, c, e]$ as the class of languages which can be recognized by lossy catalytic Turing Machines with

workspace $s := s(n)$, catalytic space $c := c(n)$, and $e := e(n)$ errors. Furthermore, we specify lossy catalytic log-space as the specific parameter setting where $s(n) = \mathcal{O}(\log n)$ and $c(n) = \mathcal{O}(poly\, n)$ and write $\mathsf{LCL}[e] := \mathsf{LCSPACE}[\mathcal{O}(\log n), poly\, n, e]$.

To begin, note that $\mathsf{LCL}[e]$ with $e \leq poly(n)$ errors trivially contains the class $\mathsf{SPACE}[e]$ by simply erasing the first $e$ bits of the catalytic tape and using them as free memory. We have not managed to prove that any space-bounded class beyond $\mathsf{L}$ is contained in $\mathsf{ZPP}$. Given that $\mathsf{CL}$ is contained in $\mathsf{ZPP}$, proving an equivalence between $\mathsf{LCL}[e]$ and $\mathsf{CL}$ for $e = \omega(\log n)$ errors would immediately show that $\mathsf{SPACE}[e]$ is contained in $\mathsf{ZPP}$, which is unlikely. The question, then, is to understand where, in the range of $e = 0$ to $e = \mathcal{O}(\log n)$, is the acceptable number of errors that $\mathsf{CL}$ can provably tolerate.

As an initial answer to the previous question, [Gup+24] show that $\mathsf{CL}$ gains no additional power from allowing any constant number of errors on the catalytic tape, i.e., $\mathsf{LCL}[\mathcal{O}(1)] = \mathsf{CL}$. This remains the frontier of our knowledge, and Mertz [Mer23] posed it as an open question to improve this result to any super-constant number of errors, or, alternatively, to provide evidence against being able to prove such a collapse.[1]

## 5.1 Our results

In this chapter, we completely characterize lossy catalytic space in terms of ordinary catalytic space. Let $\mathsf{CSPACE}[s, c]$ denote catalytic machines with free space $s$ and catalytic space $c$, and let $\mathsf{LCSPACE}[s, c, e]$ be the same with up to $e$ errors allowed in resetting the catalytic tape. We show that these $e$ errors are equivalent to an additional $e \log c$ free bits of memory, up to constant factor losses.

**5.1.1.** THEOREM. *Let $s := s(n), c := c(n), e := e(n)$ be such that $e \leq c^{1-\Omega(1)}$. Then*

$$\mathsf{LCSPACE}[\mathcal{O}(s), \mathcal{O}(c), e] = \mathsf{CSPACE}[\mathcal{O}(s + e \log c), \mathcal{O}(c)]$$

Besides characterizing $\mathsf{LCSPACE}[s, c, e]$, the main takeaway of Theorem 5.1.1 is that allowing seemingly minor (superconstant) errors in the resetting condition can give an $\mathsf{LCSPACE}$ machine surprising power. A concrete instantiation of this view is the following direct corollary.

**5.1.2.** COROLLARY. *For any $e := e(n)$,*

$$\mathsf{LCL}[e] = \mathsf{CL} \quad implies \quad \mathsf{SPACE}[\mathcal{O}(e \log n)] \subseteq \mathsf{ZPP}$$

---

[1]We cannot expect an unconditional separation between $\mathsf{CL}$ and any $\mathsf{LCL}[e]$. $\mathsf{LCL}[e]$ is contained in $\mathsf{PSPACE}$, simply by the fact that in $\mathsf{PSPACE}$ one has access to polynomial clean memory that does not have to be reset at the end of the computation. Therefore, this clean memory can act as the catalyst. Now as even separating $\mathsf{PSPACE}$ from e.g. $\mathsf{TC}^1(\subseteq \mathsf{CL})$ remains wide open, we cannot unconditionally separate $\mathsf{CL}$ from $\mathsf{LCL}[e]$ for any $e$.

If we revisit the assumption that we cannot hope to prove $\mathsf{SPACE}[e \log n]$ is in $\mathsf{ZPP}$ for any $e = \omega(1)$, then Corollary 5.1.2 implies the result of [Gup+24] is optimal with respect to $e$; any result of the form $\mathsf{LCL}[\omega(1)] = \mathsf{CL}$ is out of reach.

We also show that our proof extends to catalytic machines with additional power—usual examples include non-determinism, randomness, or non-uniformity—beyond errors; in fact, any "reasonable" classical catalytic setting is sufficient.

**5.1.3.** THEOREM. *Let* $\mathsf{C}\mathcal{B}\mathsf{SPACE}$ *be any catalytic model such that* $\mathsf{SPACE}[s] \subseteq \mathsf{C}\mathcal{B}\mathsf{SPACE}[s, 0]$, *and let* $s := s(n), c := c(n), e := e(n)$ *be such that* $e \leq c^{1-\Omega(1)}$. *Then*

$$\mathsf{LC}\mathcal{B}\mathsf{SPACE}[\mathcal{O}(s), \mathcal{O}(c), e] = \mathsf{C}\mathcal{B}\mathsf{SPACE}[\mathcal{O}(s + e \log c), \mathcal{O}(c)]$$

This also gives a barrier to a more efficient removal of errors using additional resources, as Corollary 5.1.2 also applies to all other variants.

We briefly remark that the $e \leq c^{1-\Omega(1)}$ restriction in all our results is only required to get the constant stretch in the catalytic tape, and a different version holds in the general case:

**5.1.4.** THEOREM. *Let* $s := s(n), c := c(n), e := e(n)$. *Then*

$$\mathsf{LCSPACE}[s, c, e] \subseteq \mathsf{CSPACE}[s + \mathcal{O}(e \log c), c] \subseteq \mathsf{LCSPACE}[s, \mathcal{O}(ec), e]$$

While this version pays an additional factor of $e$ in the catalytic space of the second inclusion, we also keep the number of errors fixed at exactly $e$; thus this result is somewhat incomparable to Theorem 5.1.1.

## 5.2 Main theorem

In this section we prove Theorem 5.1.1. We will do so via a simulation argument for each direction in turn.

### 5.2.1 Simulating errors with space

First, we show that $\mathsf{LCSPACE}[s, c, e] \subseteq \mathsf{CSPACE}[O(s + e \log c), O(c)]$. In fact, we will not need any increase in the length of our catalytic tape.

**5.2.1.** THEOREM. *Let* $s := s(n), c := c(n), e := e(n)$. *Then*

$$\mathsf{LCSPACE}[s, c, e] \subseteq \mathsf{CSPACE}[s + O(e \log c), c]$$

We note that this was also proven in [Gup+24] for the case of $\mathsf{LCL}[O(1)]$, but we will pursue a different proof, based on error-correcting codes, which will allow us to generalize to other catalytic models in Section 5.3.

**Proof:**

Let $M_e$ be an $\mathsf{LCSPACE}[s, c, e]$ machine. We will devise a $\mathsf{CSPACE}[s + O(e \log c), c]$ machine $M_0$ which simulates $M_e$. Note that in this section, we will not use our parameter restriction on $e$; this direction holds for every setting of $s$, $c$, and $e$. We will presume that $e \leq \frac{c}{\log(c)}$, as the inclusion becomes trivial otherwise.

Our simulation will go via an error-correcting code. In particular we will use *BCH codes*[2] ($\mathsf{BCH}$), named after Bose, Ray-Chaudhuri, and Hocquenghem [BR60; Hoc59], which we define as per [DRS04; Dod+06]. We define the mapping $\mathsf{BCH}$ and prove the following lemma in Appendix 5.B.2 (see Corollary 5.B.9, Lemma 5.B.12 and Lemma 5.B.13).

**5.2.2.** LEMMA. *Let $q := 2^{\lceil \log(c+e) \rceil}$. There exists a mapping $\mathsf{BCH} : \mathbb{F}_q^q \to \mathbb{F}_q^q$ with the following operations:*

- ***Encoding:*** $\mathsf{Enc}_{\mathsf{BCH}}$ *takes as input a string $S$ of length $c$, plus an additional $(2e + 1)\lceil \log(c + e) \rceil$ bits initialized in 0, and outputs a codeword $S_{enc}$:*

$$S + [0]_{(2e+1)\lceil \log(c+e) \rceil} \to_{\mathsf{Enc}} S_{enc}$$

  *Furthermore, all outputs $S_{enc}$ generated this way have minimum distance $\delta := 2e + 1$ from one another.*

- ***Decoding:*** $\mathsf{Dec}_{\mathsf{BCH}}$ *takes as input a string $S'_{enc}$ of length $c + (2e + 1)\log(c + e)$, with the promise that there exists a string $S$ of length $c$ such that $\mathsf{Enc}_{\mathsf{BCH}}(S + [0]_{2e \log(c+e)})$ differs from $S'_{enc}$ in at most $\delta/2 - 1 = e$ locations, and outputs the string $S$:*

$$S'_{enc} \to_{\mathsf{Dec}} S + [0]_{(2e+1)\log(c+e)}$$

*Furthermore, both $\mathsf{Enc}_{\mathsf{BCH}}$ and $\mathsf{Dec}_{\mathsf{BCH}}$ are in place replacements of the input strings, they require at most an additional $O(e \log c)$ free space of memory.*

We now move on to the simulation of our $\mathsf{LCSPACE}[s, c, e]$ machine $M_e$. Our $\mathsf{CSPACE}[s + O(e \log c), c]$ machine $M_0$ acts as follows:

1. **Initialization:** use the function $\mathsf{Enc}_{\mathsf{BCH}}$ to encode the initial state $\tau$ of the catalytic tape into a codeword, using $(2e + 1)\lceil \log(c + e) \rceil$ additional bits from clean space,

$$\tau + [0]_{(2e+1)\lceil \log(c+e) \rceil} \to_{\mathsf{Enc}} \tau_{enc}.$$

---

[2]Technically because of our parameters, they can even be considered Reed-Solomon codes, which are a special case of BCH codes; nevertheless we follow the presentation of the more general code form.

2. **Simulation:** Run $M_e$ using clean space $s$ and the first $c$ bits of $\tau_{enc}$ as the catalytic tape. When $M_e$ finishes the calculation, we record the answer in a bit of the free work tape. The catalytic tape is, at this point, in a state $\tau'_{enc}$ which differs in at most $e$ locations from $\tau_{enc}$.

3. **Cleanup:** use the function $\mathsf{Dec}_{\mathsf{BCH}}$ to detect and correct our resulting catalytic tape $\tau'_{enc}$:

$$\tau'_{enc} \to_{\mathsf{Dec}} \tau + [0]_{(2e+1)\lceil \log(c+e)\rceil}$$

Once we finish this process, we output our saved answer and halt.

The correctness of $M_0$ is clear, as it gives the same output as $M_e$. By our error guarantee on $M_e$ and the correctness of $\mathsf{Dec}$, our catalytic tape is successfully reset to $\tau$. Our catalytic memory is $c$ as before, while for our free workspace we require $s$ bits to simulate $M_e$, an additional $(2e+1)\lceil \log(c+e)\rceil = (2+o(1))e\log c$ zero bits for our codewords, and $O(e\log c)$ space for $\mathsf{Enc}_{\mathsf{BCH}}$ and $\mathsf{Dec}_{\mathsf{BCH}}$, for $s + O(e\log c)$ space in total. □

**1.** NOTE. *There is an alternative proof of this point, one which gets better parameters and relies on an interesting characterization of space, namely the reversibility of space. This proof is a simplification and extension of the one originally provided in [Gup+24], and we provide it in Appendix 5.A.*

## 5.2.2 Simulating space with errors

We now show the other direction of Theorem 5.1.1, i.e. $\mathsf{CSPACE}[s + e\log c, c] \subseteq \mathsf{LCSPACE}[\mathcal{O}(s), \mathcal{O}(c), \mathcal{O}(e)]$.

**5.2.3.** THEOREM. *Let $s := s(n), c := c(n), e := e(n)$, and $\epsilon > 0$ be such that $e = o(c^{\epsilon/(1+\epsilon)})$. Then*

$$\mathsf{CSPACE}[s + e\log c, c] \subseteq \mathsf{LCSPACE}[s + \log c, (1 + o(1))c, (1 + \epsilon)e]$$

Since $s \geq \log c$ by the definition of a catalytic machine, this achieves the reverse direction of Theorem 5.1.1 with very small blowups in $s$ and $c$, and for $e$ bounded by a small polynomial in $c$ we get a negligible error blowup as well. Note that we allow $\epsilon > 1$, and so our proof is not limited to $e < c^{1/2}$; however, we will pay for larger values of $e$ in the error blowup, and for $e = c^{1-o(1)}$ this factor becomes superconstant.

To understand our construction, we will first prove a version with looser space parameters. This result is incomparable to Theorem 5.2.3; although we lose a factor of $e$ in our catalytic space, in exchange we have no restrictions on $e$ and no loss in $e$ either. In conjunction with Theorem 5.2.1, this also proves Theorem 5.1.4.
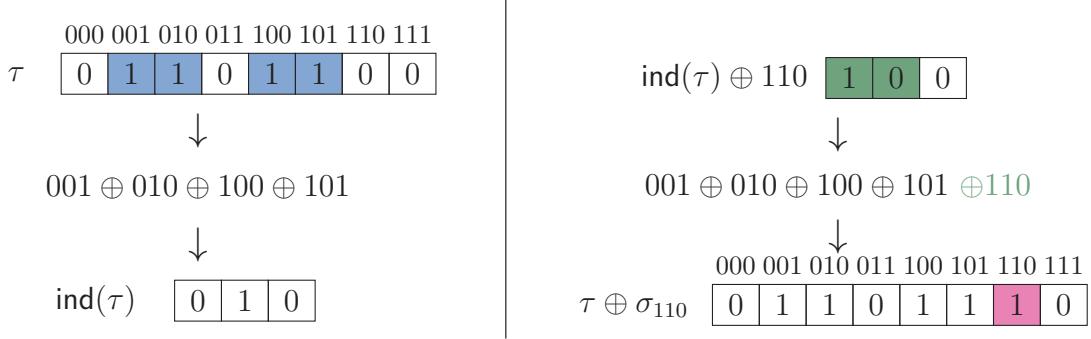
Figure 5.1: Example of our construction in Lemma 5.2.5 for $k = 3$ and $\tau = 01101100$: 1) calculating $\mathsf{ind}(\tau)$ based on the positions of the 1s in $\tau$ (blue); 2) how flipping one bit of $\tau$ (magenta) allows us to change $\mathsf{ind}(\tau)$ (changes in green).

**5.2.4.** Theorem. *Let $s := s(n), c := c(n), e := e(n)$ be such that $c$ is a power of 2. Then*

$$\mathsf{CSPACE}[s + e \log c, c] \subseteq \mathsf{LCSPACE}[s, c + e(c + \log c), e]$$

**Proof:**
Let $M_0$ be a $\mathsf{CSPACE}[s + e \log c, c]$ machine. We will devise a $\mathsf{LCSPACE}[s, c + e(c + \log c), e]$ machine $M_e$ which simulates $M_0$.

   Throughout this proof, we will associate $[2^k]$ with $\{0,1\}^k$ in the usual manner, i.e. subtracting 1 and taking the binary representation, and so we will use them interchangeably. Our workhorse is the following folklore[3] construction:

**5.2.5.** Lemma. *For every $k$, there exists a mapping $\mathsf{ind} : \{0,1\}^{2^k} \to \{0,1\}^k$, computable in space $k+1$, such that the following holds: for any $\tau \in \{0,1\}^{2^k}$ and any $y \in \{0,1\}^k$,*

$$\mathsf{ind}(\tau \oplus \sigma_y) = \mathsf{ind}(\tau) \oplus y$$

*where $\sigma_y$ is the vector of length $2^k$ with a single 1 in position $y$.*

   Intuitively, Lemma 5.2.5 gives us an easily computable mapping where any value of the $k$-bit output string can be given as output by flipping one bit of the $2^k$-bit input string.

**Proof of Lemma 5.2.5:**
Let $\tau \in \{0,1\}^{2^k}$ be indexed by bitstrings $z \in \{0,1\}^k$. We will define our mapping $\mathsf{ind}$ as the entrywise sum of all indices $z$ where $\tau_z = 1$, i.e.

$$\mathsf{ind}(\tau)_j = \bigoplus_{\substack{z \in \{0,1\}^k \\ z_j = 1}} \tau_z$$

---

[3]This construction is based on the solution to the so-called "almost impossible chessboard puzzle"; interested readers can find the setup and solution in videos on the YouTube channels 3Blue1Brown (`https://www.youtube.com/watch?v=wTJI_WuZSwE`) and Stand-up Maths (`https://www.youtube.com/watch?v=as7Gkm7Y7h4`). It can also be seen as the syndrome of the Hamming code.

This is clearly computable in space $k+1$, as we need only store $z$ and our current sum. Now note that for any $y$, flipping the entry $\tau_y$, i.e. $\tau \oplus \sigma_y$, flips every $\mathsf{ind}(\tau)_j$ entry where $y_j = 1$ and leaves all other $\mathsf{ind}(\tau)_j$ entries unchanged, which gives $\mathsf{ind}(\tau) \oplus y$ as claimed. $\qquad\square$

We now show how to simulate our $\mathsf{CSPACE}[s + e \log c, c]$ machine $M_0$ by an $\mathsf{LCSPACE}[s, c + e(c + \log c), e]$ machine $M_e$. First, let $\tau_0$ be the first $c$ bits of catalytic memory, which we will set aside for simulating $M_0$. We will break the remaining $e \cdot (c + \log c)$ bits of our catalytic tape of $M_e$ into $e$ blocks $B_1 \ldots B_e$ of size $2^k + k$ each, where $k = \log c$ (recall that $c$ is a power of 2 by assumption). Within block $B_i$, let $\tau_i$ be the first $2^k$ bits and $\mathsf{mem}_i$ be the remaining $k$ bits. Our algorithm performs as follows:

1. **Initialization:** for each block $i \in [e]$, calculate $z_i = \mathsf{ind}(\tau_i)$, set $y_i = \mathsf{mem}_i \oplus z_i$, and update $\tau_i$ to

$$\tau_i' \leftarrow \tau_i \oplus \sigma_{y_i}$$

   By Lemma 5.2.5, after this step we have that

$$\mathsf{ind}(\tau_i') = z_i \oplus y_i = \mathsf{mem}_i \qquad \forall i \in [e]$$

   Finally, zero out each block $\mathsf{mem}_i$:

$$\mathsf{mem}_i \leftarrow 0^k \qquad \forall i \in [e]$$

2. **Simulation:** run $M_0$ on catalytic tape $\tau_0$ with the work memory from $M_e$ plus $\{\mathsf{mem}_i\}_{i \in [e]}$, for a total of

$$s + ek = s + e \log c$$

   free bits as necessary.

3. **Cleanup:** when we reach the end of $M_0$'s computation, record the answer on the free work tape and reset all the blocks $\mathsf{mem}_i$ using $\mathsf{ind}$:

$$\mathsf{mem}_i \leftarrow \mathsf{ind}(\tau_i') \qquad \forall i \in [e]$$

   We then return the saved answer and halt.

The correctness of $M_e$ is clear, as we output the same value as $M_0$. We require $c + e(c + \log c)$ catalytic bits plus $s$ free bits for our simulation, while $\mathsf{ind}$ can be computed in space $k = \log c \leq s$ by assumption; Thus all our memory is as claimed.

We also claim that our lossy catalytic condition is satisfied. Each $\tau_i'$ is at most one error away from $\tau_i$ in the initialization phase, and is never altered again, giving

a total of $e$ errors. By the property of $M_0$, there are no errors made to $\tau_0$ during the simulation step. Lastly, by the property that $\mathsf{ind}(\tau_i') = \mathsf{mem}_i$, the cleanup step exactly resets the blocks $\mathsf{mem}_i$, meaning no further errors are introduced to the catalytic tape. $\qquad\square$

We now return to Theorem 5.2.3, which requires only a small modification of the above proof, namely to break the catalytic tape into more, smaller blocks, which reduces its required length at the cost of a few extra errors. This modification works because the number of pure bits represented is logarithmic in the length of the block, and so making the blocks smaller barely affects the number of bits represented; for example, $c/2$ bits in a block still let you represent $\log(c) - 1$ bits, so half the size only loses one bit per block.

**Proof of Theorem 5.2.3:**
Let $M_0$ be a $\mathsf{CSPACE}[s + e \log c, c]$ machine. We will devise a $\mathsf{LCSPACE}[s, (1 + o(1))c, (1 + \epsilon)e]$ machine $M_e$ which simulates $M_0$, where $\epsilon$ satisfies $e = o(c^{\epsilon/(1+\epsilon)})$.

We will have the same approach as Theorem 5.2.4, but now we use $(1 + \epsilon)e$ blocks of length $2^{k'} + k'$, where

$$k' = \left\lceil \frac{\log c}{1 + \epsilon} \right\rceil$$

Clearly, we make at most $(1 + \epsilon)e$ errors by the above analysis, while our free space is

$$s + (1 + \epsilon)e \cdot k' = s + (1 + \epsilon)e \cdot \left\lceil \frac{\log c}{1 + \epsilon} \right\rceil \geq s + e \log c$$

Finally, we analyze our catalytic memory. Our $\tau_i'$ blocks give us a total usage of

$$(1 + \epsilon)e \cdot 2^{k'} = (1 + \epsilon)e \cdot 2^{\lceil \log c/(1+\epsilon) \rceil} \leq (1 + \epsilon)e \cdot (2c^{1/(1+\epsilon)}) \ll c$$

where the last line follows because $e = o(c^{\epsilon/(1+\epsilon)})$. We will use our memory $\{\tau_i\}_{i \in [(1+\epsilon)\log c]}$ for the simulation of $M_0$, plus enough extra catalytic memory $\tau_0$ needed to reach $c$ total bits. Since $M_0$ exactly resets its catalytic tape this introduces no new errors, and together with the $\mathsf{mem}_i$ blocks, this gives us a total catalytic memory of

$$c + (1 + \epsilon)e \cdot k' = c + e \log c + \mathcal{O}(1) = (1 + o(1))c$$

which completes the proof. $\qquad\square$

## 5.3   Further consequences

With this, we have concluded our main theorem and proof. We now move to corollaries and extensions.

## 5.3.1   Lossy catalytic logspace with superconstant errors

As stated in the introduction, it immediately follows from Theorem 5.1.1 that proving $\mathsf{LCL}[e] = \mathsf{CL}$ is likely difficult, if not false, for superconstant values of $e$.
**Proof of Corollary 5.1.2:**
This follows immediately from the fact that

$$\mathsf{LCSPACE}[\mathcal{O}(\log n), poly\, n, e] = \mathsf{CSPACE}[\mathcal{O}(\log n + e\log(poly\, n)), poly\, n]$$
$$= \mathsf{CSPACE}[\mathcal{O}(e\log n), poly\, n]$$
$$\supseteq \mathsf{SPACE}[\mathcal{O}(e\log n)]$$

combined with the fact that $\mathsf{CL} \subseteq \mathsf{ZPP}$ by [Buh+14]. □

## 5.3.2   Lossy catalytic space with other resources

As mentioned in Section 5.1, there are many extensions of the base catalytic model besides $\mathsf{LCSPACE}$, such as randomized, non-deterministic, and non-uniform $\mathsf{CSPACE}$. So far, however, there has been little discussion of classes where more than one such external resource has been utilized. In this section we observe that our proof of Theorem 5.1.1 carries through no matter what base classical catalytic model we are using, even if we are granted additional resources which the errors can depend on.
**Proof sketch of Theorem 5.1.3:**
As earlier, we need to show both directions. We will prove the same two equivalences as in Theorems 5.2.1 and 5.2.3, namely

1. $\mathsf{LC\mathcal{B}SPACE}[s, c, e] \subseteq \mathsf{C\mathcal{B}SPACE}[s + \mathcal{O}(e\log c), c]$

2. $\mathsf{C\mathcal{B}SPACE}[s + e\log c, c] \subseteq \mathsf{LC\mathcal{B}SPACE}[s, (1 + o(1))c, (1 + \epsilon)e]$

In both cases we only need check two computations. First we simulate our machine $M_0/M_e$ via a machine $M_e/M_0$ (respectively) which is given direct access to the appropriate amount of work and catalytic memory; by definition this can be done irrespective of $\mathsf{C\mathcal{B}SPACE}$. Second is our two mappings needed to reset the catalytic tape at the end; since $\mathsf{SPACE}[s]$ can implement both our BCH codes and the mapping $\mathsf{ind}$, by assumption $\mathsf{C\mathcal{B}SPACE}$ can do so as well. □

## 5.3.3   Open problems

**Errors in expectation.**

A related question asked in [Mer23] is whether or not $\mathsf{CL}$ is equivalent to $\mathsf{CL}$ with $\mathcal{O}(1)$ errors allowed *in expectation* over all starting catalytic tapes. This

represents a different notion of distance between catalytic tapes, in opposition to Hamming distance, that may be more applicable to settings such as quantum computation. This question has received some attention in a related form by Bisoyi et al. [Bis+24], who introduce *almost* catalytic machines, which perfectly reset some catalytic tapes and are completely unrestricted on others.

However, no general results are known for expected errors—the results in [Bis+24] are very structured—and all techniques in our paper fail to restore the tape in pathological cases where a few starting tapes end up with potentially many errors. Furthermore, a barrier result was pointed out by an anonymous reviewer.[4]

### Exact simulation space requirements.

In the current simulation of errors using clean space, we use $4e \log c$ clean space. By contrast, in our simulation of clean space using errors, we use only $(1 + \epsilon)e$ more errors. If errors can be simulated in clean space $e \log c$ instead, then there is only very low overhead in switching between the two perspectives. This would tighten the correspondence between errors and space that we establish. However, since the distance between two codewords required to correct $e$ errors is $2e + 1$, a different error-correcting code would be necessary to reach clean space $e \log c$.

## 5.A    Simulating errors with space via reversibility

In this section we give an alternate proof of simulating LCSPACE via CSPACE, with sharper parameters than those in Theorem 5.2.1.

**5.A.1.** THEOREM. *Let* $s := s(n), c := c(n), e := e(n)$. *Then*

$$\mathsf{LCSPACE}[s, c, e] \subseteq \mathsf{CSPACE}[s + (e + 1) \log c, c].$$

For this proof, we need to invoke a property of space-bounded machines known as *reversibility*, which we define now.

**5.A.2.** DEFINITION. A Turing machine $M$ is (strongly) *reversible* if the following conditions hold:

---

[4]The main idea is that allowing virtually unlimited error in an exponentially small fraction of catalytic tapes gives us a strong form of the "compress-or-random" framework of previous papers; we can simulate a randomized algorithm using the catalytic tape as our source of randomness, and in the exponentially unlikely event the tape is not sufficiently entropic we simply erase it and run brute force. Formalizing this intuition and combining it with the results of [Coo+25; Kou+25] gives a derandomization barrier to showing even $\mathsf{LCL}[\mathcal{O}(1)] = \mathsf{CL}$ in this lossy setting, namely that randomized $\mathsf{TC}^1$, which is not even known to be in $\mathsf{P}$, would reduce to the lossy code problem.

1. For any input $x$, every node $v$ in its configuration graph $G_x$ has both in-degree and out-degree at most one. Let $for_x(v)$ indicate the unique node with a directed edge $(v, for_x(v))$, and let $back_x(v)$ indicate the unique node with a directed edge $(back_x(v), v)$.

2. There exist machines $M_\rightarrow$ and $M_\leftarrow$ such that for every node $v$ in the configuration graph of $M$, $M_\rightarrow(x)$ sends $v$ to $for_x(v)$ and $M_\leftarrow$ sends $v$ to $back_x(v)$.

A classical result of Lange, McKenzie, and Tapp [LMT00] shows that every SPACE$[s]$ machine can be made reversible with no additional space. Dulek [Dul15] showed the same result for catalytic machines, while Gupta et al. [Gup+24] extended this to catalytic machines with error; both of the latter results use a very similar Eulerian tour argument to [LMT00].

**5.A.3.** LEMMA. *Let $M$ be a CSPACE$[s, c]$ (resp. LCSPACE$[s, c, e]$) machine recognizing language $L$. Then there exists a reversible CSPACE$[s, c]$ (resp. LCSPACE$[s, c, e]$) machine $M'$ which recognizes $L$.*

In light of Lemma 5.A.3, it seems that there is nothing interesting to be said about LCSPACE; after all, can we not simply reverse our machine to the starting point, wherein there are no errors on the catalytic tape? While this is technically true, there may be many different starting configurations which reach the same halting state $(\tau, v)$. All such start states can and will be reached by running $M_\leftarrow$ from $(\tau, v)$ for long enough, but without knowledge of which particular start state we began with, this naïve reversing procedure cannot reset our catalytic tape free of error.

Nevertheless, a small tweak on this idea, using our additional $(e+1)\log c$ bits, immediately works.

**Proof of Theorem 5.A.1:**

Let $M_e$ be a LCSPACE$[s, c, e]$ machine, and by Lemma 5.A.3 we will assume $M_e$ is reversible. We will devise a CSPACE$[s + (e + 1)\log c, c]$ machine $M_0$ which simulates $M_e$.

We will assume without loss of generality that all start and end states of a catalytic machine $M$ are distinguished; for example, we traditionally assume any state with an all-zeroes work tape is a start state. We write $\mathsf{start}(\tau)$ to indicate the unique start state of $M$ with initial catalytic tape $\tau$, while we write $\mathsf{end}_x(\tau)$ to indicate the unique end state reached by $M$ from initial state $\mathsf{start}(\tau)$ on input $x$.

Now let $S_{x,(\tau,v)} := \{\mathsf{start}(\tau_i)\}_i$ be the set of start states such that $\mathsf{end}_x(\tau_i) = (\tau, v)$. Since $M_e$ is an LCSPACE$[s + \log c, c, e]$ machine, each $\tau_i$ can differ from $\tau$ in at most $e$ locations, and thus

$$|S_{x,(\tau,v)}| \leq \binom{c}{\leq e} \leq \frac{c^{e+1}}{2}$$

Our machine $M_0$ thus works as follows:

1. initialize a counter $num\_start$ with $\log\binom{c}{\leq e}$ bits to 0

2. simulate $M_e$ using $s \log c$ work bits and $c$ catalytic bits, incrementing $num\_start$ each time we encounter a start state $\mathsf{start}(\tau_i)$, until we reach an end state $(\tau, v)$

3. record our answer and run $M_e$ in reverse, decrementing $num\_start$ each time we encounter a start state

4. halt when we reach a start state and $num\_start = 0$, and return our recorded answer

Clearly our algorithm outputs the correct answer, resets the catalytic tape exactly, and uses at most $s + 1 + (e+1) \log c - 1$ bits of work memory plus $c$ bits of catalytic memory. $\qquad \square$

We defer this discussion to the appendix for two reasons. First, the error-correcting approach more directly applies in both directions of Theorem 5.1.1; while Lemma 5.A.3 connects to Theorem 5.2.3 and can be applied before the final resetting step, this does not provide any qualitative or quantitative gain. Second, the reliance on reversibility makes the proof unsuitable to our later generalizations from Section 5.3.2; in particular, randomized, non-deterministic, and non-uniform catalytic computations are only reversible in a limited sense, one which rules out using Lemma 5.A.3.

# 5.B   Space-efficient linear algebra on finite fields

## 5.B.1   The space complexity of solving linear systems

We prove the space efficiency of various common arithmetic and linear algebra operations necessary in order to encode and decode BCH codes. First, we introduce the concept of well-endowed rings [BCP83]. This allows us to use earlier results to argue about the efficiency of various operations on rings without having to reprove those ourselves. The fields of interest are fields of the form $GF(p^{r_n})$ for a fixed prime $p$ and a sequence $r_n$. Our results will apply to a field whose size increases asymptotically. Hence the uniformity of the calculations involved is important. But we assume that $p$ is fixed for all fields.

All these results are expressed in their asymptotic complexity in terms of the size of the ring or a length function, which may be seen as a measure of the number of bits necessary to write down a value in a ring.

**5.B.1.** DEFINITION. A length function $\rho$ for a ring $R$ is a function satisfying that for any $x, y \in R$

1. $\rho(x + y) \leq \max\{\rho(x), \rho(y)\} + \mathcal{O}(1)$

2. $\rho(xy) \le \rho(x) + \rho(y) + \mathcal{O}(\log \max\{\rho(x), \rho(y)\})$

An example is the number of bits of an integer.

From here we can define well-endowed rings as those with efficient implementations of addition, negation and multiplication.

**5.B.2.** DEFINITION. A ring $R$ with length function $\rho$ is *well endowed* if there is a succinct uniform representation in which it has efficient implementations of addition, negation and multiplication. Addition and negation are considered efficient if they can be implemented in (logspace-uniform) $\mathsf{NC}^0$ and multiplication is considered efficient if it can be implemented in (logspace-uniform) $\mathsf{NC}^1$. The parameter for $\mathsf{NC}^1$ functions is always the length function of the ring.

We now argue that basic arithmetic can be done space efficiently. This is done in the following steps. First, we argue that the polynomial ring $GF(p)[\xi]$ is well endowed and therefore we can perform polynomial addition, negation and multiplication efficiently. Then we argue that we can use this to compute the remainder of polynomial division efficiently. This allows us to find an irreducible polynomial to represent the field $GF(p^{r_n})$ in order to perform addition, negation and multiplication efficiently. We finally show that we can evaluate multiplicative inverses inefficiently and use this to do division. With inefficiently we mean in space $\mathcal{O}(\log |F|)$ for a field $F$ whereas addition, negation and multiplication can be performed in space $\mathcal{O}(\log \log |F|)$.

**5.B.3.** LEMMA. *For fixed $p$, the ring $GF(p)[\xi]$ is well endowed.*

**Proof:**
We argue that finite fields are well-endowed rings. First observe that $GF(p)$ for a fixed $p$ is always well endowed since the size of the ring is independent of $n$ so addition, negation and multiplication can be performed in $\mathsf{NC}^0$. By Proposition 3.9 from [BCP83] this means that polynomials over $GF(p)[\xi]$ are also well endowed. The length function here is $\mathcal{O}(d)$ for a polynomial of degree $d$. Since they are well endowed, one can perform addition, negation and multiplication in space $\mathcal{O}(\log d)$ for polynomials. □

We use this to compute the remainder.

**5.B.4.** LEMMA. *Given polynomials $N(\xi)$ and $D(\xi)$ in $GF(p)[\xi]$ of degree at most $r_n$, it is possible to compute the remainder $R(\xi)$ using an additional $r_n + \mathcal{O}(\log r_n)$ space. If we can overwrite $N(\xi)$ in place, the additional space necessary is $2\lceil \log r_n \rceil + \mathcal{O}(1)$.*

**Proof:**
Suppose that $\chi \in GF(p)$ is the leading coefficient of $D(\xi)$, we can compute and store $\chi^{-1}$ in constant space since $p$ is constant. We perform a kind of Gaussian elimination to compute the remainder:

1. If the degree of $D(\xi)$ exceeds the degree of $N(\xi)$ then return $N(\xi)$.

2. Let $\psi$ be the leading coefficient of $N(\xi)$, and let $z_1$ be the highest degree of $N(\xi)$ and $z_2$ be the highest degree of $D(\xi)$. Compute $N(\xi) - \psi\chi^{-1}\xi^{z_1-z_2}D(\xi)$ overwriting $N(\xi)$ in the process, this requires $\log(r_n)$ space to write down $z_1 - z_2$. Since we use a fixed field $GF(p)$, this can be done in constant depth.

3. return to step 1.

The maximum number of repetitions is $r_n$, therefore we require $\log r_n$ space for a counter. We use $\mathcal{O}(1)$ to store $\psi$ and $\chi$. We then compute $N(\xi) - \chi\psi^{-1}\xi^{z_1-z_2}D(\xi)$ is done coefficient by coefficient. Overall, we manage to compute the remainder in space $r_n + \mathcal{O}(1)$ by copying the final remainder to a new part of the space and then updating it in place during every iteration. If we can overwrite $N(\xi)$ in the process, then the additional space required is only $2\log(r_n)$ to keep track of a counter and compute $z_1 - z_2$. $\qquad\qquad\Box$

We can now search for irreducible polynomials.

**5.B.5.** LEMMA. *Given a sequence of positive integers $r_n$ and a constant prime $p$, it is possible to find a degree $r_n$ irreducible polynomial in $GF(p)[\xi]$ in space $3r_n + \mathcal{O}(\log r_n)$.*

**Proof:**
It costs $\mathcal{O}(d)$ space to store a polynomial over $GF(p)$ of degree at most $d$. Therefore, one can iterate over all such polynomials. If we store two such polynomials and iterate over all pairs, the first can be a candidate irreducible polynomial, while the second can be a candidate factor of the first polynomial. By using Lemma 5.B.4 to test whether or not the candidate irreducible polynomial is divisible by the candidate factor in additional space $r_n + \mathcal{O}(\log r_n)$, we can test if the candidate irreducible polynomial is irreducible. Irreducible polynomials are guaranteed to exist, so we must find one eventually. Counting up the total space cost we have a total of $3r_n + \mathcal{O}(\log r_n$ $\qquad\qquad\Box$

Together these results allow us to do division in $GF(p^{r_n})$.

**5.B.6.** LEMMA. *Given a sequence of finite fields $F_n = GF(p^{r_n})$ for a constant prime $p$, it is possible to compute the multiplicative inverse of an element $x \in GF(p^{r_n})$ in additional space $4r_n + \mathcal{O}(\log r_n)$ counting the space needed to store the irreducible polynomial.*

**Proof:**
If $x = 0$ then there is no multiplicative inverse. Otherwise, try multiplying $x$ by every possible $y$ and taking the remainder using Lemma 5.B.4 in place in space $\mathcal{O}(r_n)$ until one finds a $y$ such that $xy = 1$. It takes $r_n$ space to iterate over

all possible $y$. For every $x, y$, we use another register to store $xy$. Storing $xy$ needs an additional $2r_n$ space, since we first need to compute the product as a product of polynomials and only take the remainder later. Computing $xy$ uses an additional space $\mathcal{O}(\log r_n)$ since the ring of polynomials over $GF(p)$ is well endowed. Finally, we can use Lemma 5.B.4 to take the remainder in place in only $\lceil \log r_n \rceil + \mathcal{O}(1)$. $\qquad \square$

We can now finally solve linear systems.

**5.B.7. Lemma.** *Given a sequence of finite fields $F_n = GF(p^{r_n})$ for a constant prime $p$, it is possible to solve a linear system of $t_n$ equations and $t_n$ unknowns in $t_n r_n + r_n + \mathcal{O}(\log r_n)$ space if $t_n = \mathcal{O}(|F_n|)$ and we count the space used to store the irreducible polynomial for our representation of $GF(p^{r_n})$.*

**Proof:**
We presume that the $t_n$ linear equations are given as an input, and therefore written on some input tape. The goal is to find $t_n$ variables from $GF(p^{r_n})$ satisfying those equations in minimal space. A simple approach is as follows: we use of brute force search, iterating over all possible $t_n$ variables until a solution is found. Note that this is not time efficient, but it is space efficient. Writing down a trial solution costs $t_n r_n$ space. Trying a solution only requires additional space to write down the intermediate result of the equation being tested; this requires $r_n$ space to write down an element from $GF(p^{r_n})$ and additionally $\mathcal{O}(\log(r_n))$ space to perform multiplication and addition. If one equation is not satisfied, we reject the trial solution and go to the next one. If all solutions have been tried and non satisfy the linear system we output 0. $\qquad \square$

## 5.B.2 An overview of BCH codes

The codes used to correct errors in our catalytic tape are so-called *Bose–Chaudhuri–Hocquenghem (BCH)* codes, as described by [Dod+06]. They showed that there is a sublinear time algorithm for encoding and decoding BCH codes, we analyze their algorithm and show that it can also be performed in place using at most logarithmic additional space. A BCH code has the following components:

1. An alphabet represented by a 'small' field $GF(q)$.

2. Codeword length $n = q^m - 1$. Each position of the codeword is represented by a member of $F^*$, where $F^*$ is the multiplicative group of $F = GF(q^m)$ for a fixed value $m$. We may call $F = GF(q^m)$ the larger field.

3. Distance $\delta$.

And we make the following choices.

1. We set $q = p^{r_n}$ for a prime number $p$ and $r_n$ that depends on the size of the input tape of the machine. Here $p$ is fixed and we set it to $p = 2$ in this work.

2. $m = 1$, therefore the small field equals the large field $F = GF(q)$.

3. $\delta = 2e + 1$. It is well known that one needs a distance of at least $2e + 1$ to be able to correct $e$ errors.

Together these choices form a $[2^{r_n} - 1, 2^{r_n} - 1 - \delta, \delta]$-code over an alphabet of size $2^{r_n}$. Ensuring that $p = 2$ means that we can interpret the catalytic tape as a sequence of elements in $GF(q) = GF(2^{r_n})$, where $r_n$ will be chosen later. Furthermore, we wish to have the property that by extending a word by a small amount we can turn any word into a codeword. We observe that codewords are defined as words that satisfy the following property for $i = 1, \ldots, \delta - 1$.

$$s_i = \sum_{x \in F^*} d_x x^i = 0 \tag{5.1}$$

Here the $d_x$ represents the value of the codeword stored at position $x$. Now presume we have a word of length $n$ then we extend the word by adding entries, we call the list of added entries $C \subseteq F^*$. The added values can be set arbitrarily, therefore we obtain the following equations:

$$s_i = \sum_{x \in F^*} d_x x^i = \sum_{x \in F^* \backslash C} d_x x^i + \sum_{x \in C} d_x x^i = s_i' + \sum_{x \in C} d_x x^i = 0 \tag{5.2}$$

for

$$-s_i' = \sum_{x \in \mathcal{F}^* \backslash C} d_x x^i. \tag{5.3}$$

We observe that for every value of $i = 1, \ldots, \delta - 1$, we obtain an equation. Each equation is linear in the $d_x$ for $x \in C$. These are the new data points we must calculate in order to turn an arbitrary word into a codeword. Overall this yields the encoding linear system with parameter $\delta - 1$ and $i = 1, \ldots, \delta - 1$

$$\sum_{x \in C} d_x x^i = -s_i' \tag{5.4}$$

In order to argue that a solution to this system always exists, we need the small field to equal the large field of the BCH code. This means $m = 1$. This is necessary because the value $s_i'$ lies in the large field of the code while the values of $d_x$ lie in the small field of the code. If these are the same, we can treat this as a linear algebra problem.

**5.B.8.** LEMMA. *By setting $|C| = 2e = \delta - 1$, adding this many members of the alphabet of a BCH code with $m = 1$, it is always possible to turn any string into a codeword.*

**Proof:**
As discussed, it is sufficient to show that Equation 5.4 always has a solution. In order to see this, observe that Equation 5.4 forms a linear system over the field $GF(q)$ and that the matrix of this system is a Vandermonde matrix. Vandermonde matrices are always invertible. Thus a solution to this system always exists. □

**5.B.9.** COROLLARY. *Let $S$ be a data string of $n$ bits and $e \leq \frac{1}{2}n/\log(n)$, then there exists a BCH code, with distance $\delta = 2e + 1$ and codeword length $n + (2e + 1)\lceil\log(n + e)\rceil$.*

**Proof:**
We set the number of bits required to represent one letter of the code word to $r_n = \lceil\log(n + e)\rceil$ therefore $q = 2^{\lceil\log(n+e)\rceil}$, therefore the alphabet is of size $2^{\lceil\log(n+e)\rceil}$. We break the initial data string $S$ into blocks of length $\lceil\log(n + e)\rceil$, these blocks form the initial letters of the word. If $\lceil\log(n+e)\rceil \nmid n$, we pad the last block of $S$ with additional 0's, this requires at most $\lceil\log(n+e)\rceil - 1$ additional bits. This gives a word consisting of $\lceil\frac{n}{(\lceil\log(n+e)\rceil)}\rceil$ letters. Now we use Lemma 5.B.8 and add $2e$ letters of size $\lceil\log(n + e)\rceil$, using $2e\lceil\log(n + e)\rceil$ of additional bits, such that these new members abide by Equation 5.4. This creates a codeword of length $n + (2e + 1)\lceil\log(n + e)\rceil$ as required. □

Given that this code exists and has the correct space complexity we will show that it can be space efficiently computed. Even before doing encoding and decoding, it is required to do an initialization step:

---
**Algorithm 1** Initialization
---
1: **Input**: $r \in \mathbb{N}$
2: Compute an irreducible polynomial of degree $r$ in $GF(2)[\xi]$ via the procedure described in the proof of 5.B.5
3: Pick and save an element that is not 0 and not 1 in $GF(2^r)$. We can always pick this to be the polynomial $\xi$.
4: **return** An irreducible polynomial of degree $r$ and a generator of the multiplicative group of $GF(2^r)$.

---

We now argue the initialization can be done space efficiently.

**5.B.10.** LEMMA. *Given a sequence of fields $F_n = GF(2^{r_n})$, Algorithm 1 can be performed in space $3r_n + \mathcal{O}(\log r_n)$.*

**Proof:**

We review each step of Algorithm 1 and review their space cost:

1. For step 1, use Lemma 5.B.5 to find an irreducible polynomial in space $3r_n + \mathcal{O}(\log r_n)$. Only $r_n$ is needed to store the result.

2. For step 2, we can pick and save the element of $GF(2)[\xi]$ corresponding to $\xi$. This uses $\mathcal{O}(1)$ space if always done the same. This does not work for $r_n = 1$, but we can assume always $r_n > 2$.

$\square$

Encoding requires solving the linear equations given by Equation 5.4, finding the values $d_x$ for $x \in C$, the additional blocks that were appended. Solving these linear equations requires first calculating the quantities $s_i'$, given by Equation 5.3. We use the following algorithm to calculate a specific value $s_i$. By stopping prematurely, we can compute $s_i'$.

---

**Algorithm 2** ComputeChecks

1: **Input**: Integer $i$ such that $0 < i < \delta$, an integer $t$ such that $0 \leq t < q$, generator $\xi$ of $GF(2^{r_n})$, and a data string $S$.
2: Open five registers to store elements of $GF(2^{r_n})$ labelled $I, P, M, E, O$ for Index, Power, Multiplication, Sum, End and Out.
3: Set all registers to 0.
4: Open an additional register to store elements of $\{0, 1, \ldots, \delta - 1\}$ called $C$.
5: $E \leftarrow \xi^t$ via iterated multiplication. Use register $M$ as a counter.
6: $I \leftarrow \xi$
7: $P \leftarrow I^i$ via iterated multiplication. Use register $C$ as a counter in this process.
8: $M \leftarrow P * d_x$.
9: $O \leftarrow O + M$
10: $P, M \leftarrow 0$.
11: $I \leftarrow I * \xi$
12: Return to step 8 until $I = E$.
13: **return** The value $s_i$ in register $S$ computed on the word.

---

Now we give the space complexity of Algorithm 2.

**5.B.11.** LEMMA. *Algorithm 2 calculates $s_i$, or $s_i'$ by $t < q$, on input $i$, $t$ (stopping point), and generator $\xi$. This algorithm uses space $5r_n + \lceil \log \delta \rceil + \mathcal{O}(\log r_n)$, counting space $r_n$ to store the output.*

**Proof:**

Every element of $GF(2^{r_n})$ uses space $r_n$. We use five of these. We also use one registers of size $\lceil \log \delta \rceil$. Multiplication and addition use overhead $\mathcal{O}(\log r_n)$.   $\square$

We present the BCH encoding algorithm, and argue that it is space efficient.

---

**Algorithm 3** $Encode_{BCH}$

---

1: **Input:** A data string $S$.
2: **Initialization**: We assume that Algorithm 1 has been performed in advance.
3: Compute and store the $s'_i$.
4: Solve the linear system given by Equation 5.4.
5: Append the solution to data string $S$ creating $S_{enc}$.
6: **return** $S_{enc}$ a BCH codeword.

---

**5.B.12.** LEMMA. *It is possible to encode a word of length $2^{r_n} - \delta r_n$ with an alphabet $F_n = GF(2^{r_n})$ and distance $\delta$, such that $\delta = 2e+1$, as a codeword of length $2^{r_n}$ with an additional space overhead of $\mathcal{O}(\delta \log n) = (2\delta + 5)r_n + \mathcal{O}(\log r_n) + \lceil \log(\delta) \rceil$, where we set, $r_n = \log(e + n) \le \log(n) + 1$. Furthermore this encoding procedure is done in place. This implements the function $Enc_{BCH}$.*

**Proof:**
We look at the space complexity of every step of the encoding procedure.

1. Initialization costs $3r_n + \mathcal{O}(\log r_n)$ space by Lemma 5.B.10.

2. For step 3, use Algorithm 2. This means we store $\delta r_n$ values and use $5r_n + \lceil \log \delta \rceil + \mathcal{O}(\log r_n)$ space.

3. For step 4, we use Lemma 5.B.7 which uses $\delta r_n + r_n + \mathcal{O}(\log r_n)$ space.

Overall, this adds up to a space cost $\mathcal{O}(\delta \log n) = (2\delta + 5)r_n + \mathcal{O}(\log r_n) + \lceil \log(\delta) \rceil$. Note also that encoding only blocks to the existing word, making this computation done in place. □

   That completes encoding. We now move to our analysis of decoding. We review the mathematics of the decoding.
   We now describe the theory of the decoding algorithm. Decoding follows the procedure described in [DRS04; Dod+06] with some simplifications since we prioritize space over time. First, the syndrome $\mathsf{syn}(p)$ of a message $p$ is computed. The syndrome is defined as the collection of the $s_i$ values defined before. From the syndrome we compute the support of the error, $\mathsf{supp}(p) = \{(x, p_x)_{x:p_x \ne 0}\}$ which is defined as the value of the error $p_x$ together with its position $x$. Then the error can be 'subtracted' from the word to give back the original codeword. The error correction method only works if the number of errors is at most $(\delta - 1)/2$ and hence we set $\delta = 2e + 1$. It is important for space efficiency that we store only the support of the error, instead of a full error string which would require too much space. The support on the other hand uses exactly $\mathcal{O}(e \log c)$ space.

The decoding algorithm is a variation of Berlekamp's BCH decoding algorithm. First, define the following polynomials using $M = \{x \in \mathcal{F}^* | p_x \neq 0\}$

$$\sigma(z) = \prod_{x \in M}(1 - xz) \qquad \omega(z) = \sigma(z) \sum_{x \in M} \frac{p_x xz}{1 - xz} \qquad (5.5)$$

which both have degree at most $|M| \leq (\delta - 1)/2$. Here $\sigma(z)$ is known as the error locator polynomial since the multiplicative inverses of its roots are the locations of the errors. Similarly, $\omega(z)$ is known as the evaluator polynomial since it gives the error since $\omega(x^{-1}) = p_x \prod_{y \in M, y \neq x}(1 - yx^{-1})$. Note that since these polynomials have no common zeroes, $\gcd(\sigma(z), \omega(z)) = 1$.

It turns out that $\sigma(z)$ and $\omega(z)$ are the almost unique solutions to the congruence (with parameter $\delta - 1$)

$$S(z)\sigma(z) \equiv \omega(z) \pmod{z^\delta} \qquad (5.6)$$

where $S(z) = \sum_{i=1}^{\delta-1} s_i z^i$. Suppose that $\sigma'(z), \omega'(z)$ are other solutions to this congruence then

$$\omega(z)\sigma'(z) \equiv \sigma(z)S(z)\sigma'(z) \equiv \sigma(z)\omega'(z) \pmod{z^\delta}. \qquad (5.7)$$

Therefore if we restrict the degree of both $\omega(z)$ and $\sigma(z)$ to be polynomials of degree at most $(\delta - 1)/2$ then as polynomials it is also true that $\omega(z)\sigma'(z) = \sigma(z)\omega'(z)$ and therefore that $\omega(z)/\sigma(z) = \omega'(z)/\sigma'(z)$. So if we also require that $\omega(z)$ and $\sigma(z)$ are relatively prime and $\sigma(z)$ has constant coefficient 1, then $\omega(z), \sigma(z)$ are unique. We call the linear system over $GF(q)$ from Equation 5.6 the decoding linear system with parameter $\delta$.

After setting the constant term of $\sigma(z)$ to be 1, the above congruence gives a linear system with $\delta$ unknowns and $\delta$ equations with coefficients in the field $GF(q)$. We use almost the same procedure as described in the encoding step and making use of Lemma 5.B.7 in order to solve this system. If less than $(\delta - 1)/2$ errors are made, a solution is guaranteed to exist. However, we cannot force $\sigma(z)$ and $\omega(z)$ to be coprime in the linear system and as a result the solution may not be unique. Suppose $\sigma(z)$ and $\omega(z)$ have a common factor $\tau(z)$. One simple way of asuring that $\sigma(z)$ and $\omega(z)$ are coprime, is by calculating $gcd(\sigma(z), \omega(z))$ and rejecting the solution if it is not 1. It is promised that there is a unique solution to Equation 5.6 with this property, therefore when we iterate over all possible solutions, we can add this additional step as a requirement.

Having solved for polynomials $\sigma(z)$ and $\omega(z)$ we can iterate over all possible values of $z \in \mathcal{F}^*$ to find all roots to $\sigma(z)$ and then compute their inverses using a similar procedure to that described in the encoding step. The evaluation of this polynomial can be done space efficiently, similar to the evaluation of $s_i$ but much simpler in fact. Afterwards, we can evaluate $\omega(z)$ to compute the errors. This is not necessary when $q = 2$ and the error is guaranteed to be 1. Once these

---

**Algorithm 4** $Decode_{BCH}$

---

1: **Input:** Corrupted data string $S'$
2: **Initialization**: We assume that Algorithm 1 has been performed in advance.
3: Compute the $s_i$'s using Algorithm 2.
4: Use Lemma 5.B.7 to solve linear system 5.6 with the additional constrained that $gcd(\sigma(z), \omega(z)) = 1$.
5: set $j$ to the degree of $\sigma(z)$.
6: **for** $i \leftarrow 0; i < j; i \leftarrow i + 1$ **do**
7:     Find the $i$th root, $x_i^{-1}$ of the error locator polynomial according to some ordering.
8:     Compute the quantity $\alpha_{x_i}^{-1} = \left( \prod_{y \in M, y \neq x} (1 - yx^{-1}) \right)^{-1}$
9:     Evaluate the evaluator polynomial and compute the errors by multiplying $\omega(x^{-1})$ by $\alpha_{x_i}^{-1}$.
10:     Correct $S'$ in location $x_i$ using $\alpha_{x_i}^{-1}\omega(x^{-1})$.
11: **end for**
12: **return** $S_{enc}$

---

have been computed, storing the support of the error is space efficient and the data string can be corrected. This completes the decoding step. This procedure is performed by the following algorithm, and we give it space complexity.

**5.B.13.** LEMMA. *Algorithm 4 can be performed with space overhead $\mathcal{O}(e \log n) = (6 + 4e)r_n + \mathcal{O}(\log^2 r_n)$, for $r_n = \log(e + c) \leq \log(c) + 1$, including the cost of the initialization using Algorithm 1. Furthermore, this decoding procedure is done in place. This implements the function $Dec_{BCH}$.*

**Proof:**
We review the cost of Algorithm 4 step-by-step. Steps that use a trivial amount of space are omitted.

2. Initialization costs $3r_n + \mathcal{O}(\log r_n)$ space by Lemma 5.B.10.

3. By Lemma 5.B.11, the cost of Algorithm 2 is $5r_n + \lceil \log \delta \rceil + \mathcal{O}(\log r_n)$ and storing the $s_i$'s requires $\delta r_n$ space.

4. There are $\delta - 2$ unknowns of size $r_n$ therefore solving the linear system requires $(\delta - 2)r_n + r_n + \mathcal{O}(\log(r_n))$ space by Lemma 5.B.7, a further $\delta r_n$ space is required to compute $gcd(\sigma(z), \omega(z))$. Note that this space stays occupied because we have to remember both polynomials.

5. Saving $j$ requires $\log \delta$ space.

7. Finding the $i$th root is achieved by iterating over elements of $GF(2^{r_n})$, requiring $r_n + \mathcal{O}(\log r_n)$ space . Inverting the element $x_i$ takes $r_n + \mathcal{O}(\log r_n)$ space additionally.

8. Computing $\alpha_{x_i}^{-1}$ additionally takes $r_n + \mathcal{O}(\log r_n)$ space.

9. Evaluating $\omega(x_i^{-1})$ requires $r_n + \mathcal{O}(\log r_n)$ space.

10. Given $\alpha_{x_i}^{-1}$, $\omega(x_i^{-1})$, and $x_i$, correcting $S'$ requires $\mathcal{O}(\log r_n)$ space.

This covers all steps of Algorithm 4 with significant space costs. We ignore $\mathcal{O}(\log \delta)$ space terms here, since these are all $\mathcal{O}(\log n)$. This adds up to $(5 + 4\delta)r_n + \mathcal{O}(\log r_n)$ space. Note that the correction of corresponding errors is done in place on the codeword. □

# Chapter 6
## Quantum catalytic computation

In this chapter, we initiate the systematic study of catalytic techniques in the quantum setting. Our main goal is to define and study a new complexity-theoretic model, which we will call *quantum catalytic space* (QCSPACE). Loosely speaking, we equip a space-bounded quantum computational model with an additional catalytic tape containing an arbitrary quantum state.

We start by giving a concrete definition of this computational model. This includes a discussion on the set of states that are considered possible initializations of the quantum catalytic tape. Important in this discussion is the idea that quantum catalytic techniques could also be used outside this specific model to aid space-bounded computation (without the presence of a catalyst), similar to the catalytic subroutines developed in the classical setting [CM24; Wil25].

We also discuss the underlying computational model on which we define quantum catalytic space. We give a definition for both quantum circuits and quantum Turing machines, and show that these models are equivalent. This equivalence follows from a somewhat surprising result: we find a polynomial-time bound for quantum catalytic log-space computations; the classical analog of this is the largest open question in catalytic computing.

Furthermore, we prove that quantum catalytic log-space can simulate log-depth threshold circuits, a class which is known to contain (and believed to strictly contain) quantum log-space, thus showcasing the power of quantum catalytic space. Finally, we show that both unitary quantum catalytic log-space and classical catalytic log-space can be simulated in the one-clean qubit model.

## 6.1 Preliminaries

In this chapter, we require additional information on quantum channels. We will give a short description of what we need, however, if the reader is interested in more information on this, we would like to refer them to [NC10]. In addition to

quantum channels, we also introduce *quantum Turing machines*, an extension of Turing machines that can perform quantum operations.

### 6.1.1   Quantum channels

We denote the identity channel on $d$ qubits by $\mathcal{I}_d$, or just $\mathcal{I}$ when $d$ is clear from context.

The *Choi matrix* of a channel $\Phi$ that acts on an input space $\mathcal{H}$ of dimension $d$ is defined by the action of $\Phi$ on the first register of a maximally entangled state, in $\mathcal{H} \otimes \mathcal{H}$

$$J(\Phi) := (\Phi \otimes \mathcal{I}_d) \left( \frac{1}{d} \sum_{i,j=1}^{d} |i\rangle \langle j| \otimes |i\rangle \langle j| \right) = \frac{1}{d} \sum_{i,j=1}^{d} \Phi \left( |i\rangle \langle j| \right) \otimes |i\rangle \langle j| .$$

**6.1.1. DEFINITION.** The trace distance between two density matrices $\rho, \sigma \in D(\mathcal{H})$ is defined by:

$$||\rho - \sigma||_1 = Tr[\sqrt{(\rho - \sigma)^\dagger (\rho - \sigma)}],$$

where $A^\dagger$ denotes the conjugate transpose of the matrix $A$, i.e., $A^\dagger = \bar{A}^T$.

It is well known that no physical process can increase the trace distance between two states:

**6.1.2. LEMMA** (Contractivity under CPTP maps [NC10, Theorem 9.2]). *Let* $\Phi \in \mathcal{C}(D(\mathcal{H}))$ *and* $\rho, \sigma \in D(\mathcal{H})$ *then the trace distance between* $\rho$ *and* $\sigma$ *cannot increase under application of* $\Phi$:

$$||\Phi(\rho) - \Phi(\sigma)||_1 \leq ||\rho - \sigma||_1$$

**Quantum circuits**

We will require a more specific definition of quantum circuits than Definition 2.2.3. For this, we use similar definitions to those provided by [FR21], which readers may refer to for more details.

**6.1.3. DEFINITION.** Let $s := s(n), t := t(n), k := k(n)$, let $\mathcal{K}$ be a family of Turing machines, and let $\mathcal{G}$ be a set of $k$-local operators. A $\mathcal{K}$-uniform space-$s$ time-$t$ family of quantum circuits over $\mathcal{G}$ is a set $\{Q_x\}_{x \in \{0,1\}^n}$, where each $Q_x$ is a sequence of tuples $\langle i, g, j_1 \ldots j_k \rangle \in [t] \times \mathcal{G} \times [s]^k$ such that there is a deterministic TM $M \in \mathcal{K}$ which, on input $x \in \mathcal{X}$, outputs a description of $Q_x$.

The execution of $Q_x$ consists of initializing a vector $|\psi\rangle$ to $|0^s\rangle$ within $\mathcal{H}_s$ and applying, for each step $i \in [t]$ in order, each gate $g$ to qubits $j_1 \ldots j_k$ such that $\langle i, g, j_1 \ldots j_k \rangle \in Q_x$. The output of $Q_x$ is the value obtained by measuring the first qubit at the end of the computation.

If $\mathcal{G}$ consists of unitary operators, we call these *unitary circuits* and call each $g$ a *gate*. If $\mathcal{G}$ additionally consists of measurements together with postprocessing and feed forward by (classical) $\mathcal{K}$-machines, we call these *general circuits* and call each $g$ a *channel*.

**Quantum Turing machines**

Our fundamental computation model in quantum computing will be the quantum analogue of Turing machines [Deu85; BV97], which we define informally below.

**6.1.4.** DEFINITION (Quantum Turing machine). A *quantum Turing machine* is a classical Turing machine with an additional quantum tape and quantum register. The quantum register does not affect the classical part of the machine in any way, except that the qubits in the quantum register can be measured in the computational basis. On doing so, the values read from the measurement are copied into the classical register, from where they can be used to affect the operation of the machine. The quantum Turing machine can perform any gate from its quantum gate set on its quantum registry. We assume this gate set is fixed and universal. Finally, the tape head on the quantum tape can swap qubits between the quantum registry and the position that the quantum tape head is located at. This applies a two-qubit SWAP gate.

We define a number of complexity classes in Section 2.3.3 of the preliminaries with respect to the quantum circuit model. These complexity classes can also be defined with respect to quantum Turing machines. It is known that their definitions are equivalent for quantum polynomial time computations [Yao93] and logarithmic-space computations [FR21].

## 6.2 Quantum catalytic space

The first goal of this chapter is to find a proper definition of quantum catalytic space. There are many choices that have to be made in the model, but we begin with our general definition up front, leaving questions of machine model, uniformity, gate set, and initial catalytic tapes. These will be discussed and clarified in the rest of this section.

**6.2.1.** DEFINITION (Quantum catalytic machine). A *quantum catalytic machine* with workspace $s := s(n)$, catalytic space $c := c(n)$, uniformity $\mathcal{K}$, gate set $\mathcal{G}$, and catalytic set $\mathcal{A}$ is a $\mathcal{K}$-uniform quantum machine $M$ with operations from $\mathcal{G}$ acting on two Hilbert spaces, $\mathcal{H}_s$ and $\mathcal{H}_c$, of dimensions $2^s$ and $2^c$ respectively. The latter space, called the *catalytic tape*, will be initialized to some $\rho \in \mathcal{A} \subseteq D(\mathcal{H}_c)$. We require that for any $\rho \in \mathcal{A}$, if we initialize the catalytic tape to state $\rho$, then on any given input $x \in \{0,1\}^n$, the execution of $M(x)$ halts with the state $\rho$ on the catalytic tape.

This gives rise to the following complexity classes:

**6.2.2.** Definition (Quantum catalytic complexity). $\mathsf{QCSPACE}[s, c]$ is the class of Boolean functions which can be decided with probability 1, by a quantum catalytic machine with work memory $s$ and catalytic memory $c$.

$\mathsf{BQCSPACE}[s, c]$ is the class of Boolean functions which can be decided with probability 2/3, by a quantum catalytic machine with work memory $s$ and catalytic memory $c$.

We further specify to the case of quantum catalytic logspace:

**6.2.3.** Definition (Quantum catalytic logspace).

$$\mathsf{QCL} = \bigcup_{k \in \mathbb{N}} \mathsf{QCSPACE}[k \log n, n^k]$$

$$\mathsf{BQCL} = \bigcup_{k \in \mathbb{N}} \mathsf{BQCSPACE}[k \log n, n^k]$$

## 6.2.1 Machine model

We begin by defining the two natural choices of base model for quantum catalytic machines, namely *Turing machines* and *circuits*.

**6.2.4.** Definition (Quantum catalytic Turing machine). A *quantum catalytic Turing machine* is defined as in Definition 6.2.1, using quantum Turing machines as our machine model. We write $\mathsf{QCSPACEM}$ (respectively $\mathsf{BQCSPACEM}$, $\mathsf{QCLM}$, and $\mathsf{BQCLM}$) to refer to $\mathsf{QCSPACE}$ with quantum Turing machines.

**6.2.5.** Definition (Quantum catalytic circuits). A *quantum catalytic circuit* is defined as in Definition 6.2.1 with time-$2^{\mathcal{O}(s)}$ quantum circuits, that are $\mathsf{SPACE}(s)$ uniform, as our machine model. We write $\mathsf{QCSPACEC}$ (respectively $\mathsf{BQCSPACEC}$, $\mathsf{QCLC}$, and $\mathsf{BQCLC}$) to refer to $\mathsf{QCSPACE}$ using quantum catalytic circuits.

Given that $\mathsf{CL}$ and related classes are defined in terms of (classical) Turing machines, the option of circuits seems surprising and perhaps unnatural. For example, Definition 6.2.5 imposes a time bound as part of its definition, while for $\mathsf{CL}$ there is no known containment in polynomial time. For quantum circuits and quantum Turing machines without access to the catalytic tape, an equivalence has long been known; however, Definition 6.2.5 only allows for circuits of length $2^{\mathcal{O}(s)}$, while a generic transformation on $s + c$ qubit registers would give a circuit of length $2^{\mathcal{O}(s+c)}$, i.e. requiring an exponential overhead.

The main result of this chapter is to show that these models are in fact equivalent:

**6.2.6.** THEOREM. *For $s = \Omega(\log n), c = 2^{\mathcal{O}(s)}$*

$$\mathsf{QCSPACEM}[\mathcal{O}(s), \mathcal{O}(c)] = \mathsf{QCSPACEC}[\mathcal{O}(s), \mathcal{O}(c)]$$

$$\mathsf{BQCSPACEM}[\mathcal{O}(s), \mathcal{O}(c)] = \mathsf{BQCSPACEC}[\mathcal{O}(s), \mathcal{O}(c)]$$

For the rest of this section, we will deal with all auxiliary issues, namely the choice of catalytic tapes and gate set, for quantum circuits alone; while all proofs can be made to hold for quantum Turing machines without much issue, this is also obvious by Theorem 6.2.6, which we will prove in Section 6.4.

## 6.2.2   Catalytic tapes

We now move to discussing the choice of initial catalytic tapes $\mathcal{A}$. Perhaps the most immediate choice is to put no restrictions on $\mathcal{A}$ and allow our catalytic tapes to come from the set of all density matrices in $D(\mathcal{H}_c)$; this will ultimately be our definition.

**6.2.7.** DEFINITION. We fix the catalytic set in Definition 6.2.1 to be $\mathcal{A} = D(\mathcal{H}_c)$.

While this is a natural option, encompassing every possible state on $c$ qubits, there are other choices one can make. We propose four natural options—density matrices and three others—and show that all four are equivalent, thus justifying our choice.

**6.2.8.** DEFINITION. We define the following catalytic sets:

- Density is the set of all density matrices $\rho \in D(\mathcal{H}_c)$.

- Pure is the set of all pure states $|\psi\rangle \in \mathcal{H}_c$.

- PauliProd $= \{|\mathrm{PP}\rangle : |\mathrm{PP}\rangle = \bigotimes_{i=1}^{c} |\phi\rangle_i\}$, is the set of tensor products of eigenstates of the single-qubit Pauli operators, where $|\phi\rangle_i \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\circlearrowright\rangle, |\circlearrowleft\rangle\} \subset \mathcal{H}_2$.

- EPR $= \{\frac{1}{\sqrt{2^c}} \sum_{i=0}^{2^c-1} |i\rangle |i\rangle\} \subset \mathcal{H}_c \otimes \mathcal{H}_c$ is the unique state of $c$ EPR pairs, where the catalytic tape will be formed of one half of each EPR pair; the other halves are retained as a reference system, which cannot be operated on by the quantum circuit. The quantum circuit is of the form $Q_x = \tilde{Q}_x \otimes \mathcal{I}_c$, acting as the Identity on the second set of halves of the EPR pairs that is inaccessible to the circuit.

**6.2.9.** REMARK. We briefly comment on the fourth set, i.e. EPR. Using classical catalytic techniques as a subroutine has proven to be very useful, for instance, in giving an algorithm for tree evaluation in $\mathcal{O}(\log n \log(\log n))$ space [CM24]. One can also consider using analogous quantum catalytic techniques as subroutines for quantum computations, albeit this does not appear straightforward due to inherent quantum limitations. This complication can be effectively modeled by considering the initial state of the catalytic tape to be the halves of $c$ EPR pairs.

We will now prove that the four classes of quantum catalytic circuits with initial catalytic states restricted to one of the four sets $D(\mathcal{H}_c)$, $\mathcal{H}_c$, PauliProd, and EPR respectively, are all equivalent. For this we first require the following fact about the Pauli matrices.

**6.2.10.** FACT. Any $2^d \times 2^d$ complex matrix can be written as a linear combination of rank-1 outer products of states from PauliProd over $d$ qubits. In other words, the complex span of the set of $d$-qubit tensor products of Pauli eigenstates equals the set of $2^d \times 2^d$ complex matrices.

**Proof:**
Note that all four Pauli matrices can be written as a linear combination of two of the Pauli eigenstates:

$$
\begin{aligned}
I &= |0\rangle\langle 0| + |1\rangle\langle 1|, \quad X = |+\rangle\langle +| - |-\rangle\langle -|, \\
Z &= |0\rangle\langle 0| - |1\rangle\langle 1|, \quad Y = |\circlearrowright\rangle\langle\circlearrowright| - |\circlearrowleft\rangle\langle\circlearrowleft|.
\end{aligned}
$$

The four Pauli matrices form a basis for $2 \times 2$ complex matrices. Consequently, Pauli strings of length $d$—i.e., tensor products of $d$ Pauli matrices—form a basis for $2^d \times 2^d$ matrices. $\qquad\square$

Now we can state the theorem:

**6.2.11.** THEOREM. *Let* $\mathsf{QCC}_A$ *denote quantum catalytic circuits with initial catalytic tapes coming from A. Then, the following four classes of quantum catalytic circuits are equivalent:*

$$
\mathsf{QCC}_{\mathsf{Density}} = \mathsf{QCC}_{\mathsf{Pure}} = \mathsf{QCC}_{\mathsf{PauliProd}} = \mathsf{QCC}_{\mathsf{EPR}}
$$

**Proof:**
First note the obvious implications: for any quantum catalytic circuit $\Phi$,

$$
\begin{aligned}
\Phi \in \mathsf{QCC}_{\mathsf{Density}} &\implies \Phi \in \mathsf{QCC}_{\mathsf{Pure}} \\
\Phi \in \mathsf{QCC}_{\mathsf{Pure}} &\implies \Phi \in \mathsf{QCC}_{\mathsf{PauliProd}}
\end{aligned}
$$

these follow due to the fact that $\mathsf{PauliProd} \subset \mathsf{Pure} \subset \mathsf{Density}$. To finish the proof, we will further show the following two implications.

$$(1) \quad \Phi \in \mathsf{QCC}_{\mathsf{PauliProd}} \implies \Phi \otimes \mathcal{I}_c \in \mathsf{QCC}_{\mathsf{EPR}}$$
$$(2) \quad \Phi \otimes \mathcal{I}_c \in \mathsf{QCC}_{\mathsf{EPR}} \implies \Phi \in \mathsf{QCC}_{\mathsf{Density}}$$

We first prove implication (1). Let $\Phi$ be a circuit from $\mathsf{QCC}_{\mathsf{PauliProd}}$ and consider the action of $\Phi \otimes \mathcal{I}_c$ (where the Identity operator acts on the inaccessible halves of the EPR pairs) on the state $\frac{1}{2^c} |0\rangle \langle 0| \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j|$:

$$\Phi \otimes \mathcal{I}_c \left( \frac{1}{2^c} |0\rangle \langle 0| \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j| \right) = \frac{1}{2^c} \sum_{i,j} \Phi\left( |0\rangle \langle 0| \otimes |i\rangle \langle j| \right) \otimes |i\rangle \langle j|,$$

because $\Phi$ being a channel is a linear operator. By theorem 6.2.10, $|i\rangle \langle j|$ can be written as a linear combination of rank-1 projectors onto $\mathsf{PauliProd}$ states. Since $\Phi$ is catalytic with respect to $\mathsf{PauliProd}$, it follows that

$$\frac{1}{2^c} \sum_{i,j} \Phi\left( |0\rangle \langle 0| \otimes |i\rangle \langle j| \right) \otimes |i\rangle \langle j| = \eta \otimes \frac{1}{2^c} \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j|,$$

for some state in $\eta \in D(\mathcal{H}_s)$. This shows that $\Phi \in \mathsf{QCC}_{\mathsf{EPR}}$.

Implication (2) requires a similar approach. Let $\tilde{\Phi} \in \mathsf{QCC}_{\mathsf{EPR}}$, then we can write $\tilde{\Phi} = \Phi \otimes \mathcal{I}_c$. For a given input state $|0\rangle \langle 0| \in \mathcal{H}_s$ the action of $\Phi \otimes \mathcal{I}_c$ must satisfy

$$\Phi\left( \frac{1}{2^c} \sum_{i,j} |0\rangle \langle 0| \otimes |i\rangle \langle j| \right) \otimes |i\rangle \langle j| = \eta \otimes \frac{1}{2^c} \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j|,$$

for some state in $\eta \in D(\mathcal{H}_s)$. Since the catalytic state of $c$ EPR pairs is returned perfectly unaffected for every choice of input state, the effective channel of $\Phi$ can also be written as a tensor product channel: $\Phi = \Gamma_s \otimes \Xi_c{}^{[1]}$, with the action of $\Xi_c$ being

$$\frac{1}{2^c} \sum_{i,j} \Xi_c\left( |i\rangle \langle j| \right) \otimes |i\rangle \langle j| = \frac{1}{2^c} \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j| \ .$$

Note that although the effective channel factorizes into a tensor product across the work and catalytic registers, without the catalytic tape much larger circuits may be required to implement $\Gamma_c$. Moving forward, this implies that the Choi matrix of $\Xi_c$ is

$$J(\Xi_c) = \sum_{i,j} \Xi_c\left( |i\rangle \langle j| \right) \otimes |i\rangle \langle j| = \sum_{i,j} |i\rangle \langle j| \otimes |i\rangle \langle j| = J(\mathcal{I}),$$

---

[1]It seems that the catalyst does not offer any improvement, because we can write $\Phi$ as a tensor product of the action on the logspace clean qubits and the action of the catalyst, however this does not need to hold. Only the action as a whole is writable as a tensor product, it might actually consist of intermediate steps that are not of tensor product form, therefore $\Gamma_s$ might only have an efficient circuit description in the presence of a catalyst.

and therefore the effective channel $\Xi_c$ is the identity channel. This gives that for any state $\rho \in \mathcal{H}_c$ it must hold that on input $|0\rangle \langle 0|$, the channel $\Phi$ must act as follows:

$$\Phi(|0\rangle \langle 0| \otimes \rho) = \eta \otimes \rho$$

<div align="right">□</div>

**6.2.12.** REMARK. In the proof that these channel definitions are equivalent we actually showed that any channel under one definition also furnishes an instance of the other definitions. This means that they are also operationally equivalent. These equivalence proofs therefore have to hold for any type of machine model that has to adhere to the same restrictions in restoring the state of catalytic space. In particular it also holds for quantum Turing machines.

### 6.2.3   Gate set

When discussing quantum circuits, a fundamental issue is the underlying gate set. Unlike the classical case, unitary operations form a continuous space, and finite-sized circuits over finite gate sets cannot implement arbitrary unitaries. However, there do exist finite gate sets of constant locality (that is, fan-in) which are quantum universal, in the sense that any $n$-qubit unitary may be approximated to any desired precision $\epsilon$ in $\ell_2$-distance by a product of $l = \mathcal{O}(poly \log \frac{1}{\epsilon})$ gates from the universal gate set; this is the celebrated Solovay-Kitaev theorem [Kit97; DN06; NC10]. From the standpoint of complexity classes, Nishimura and Ozawa [NO09] also showed that polynomial-time quantum Turing machines are exactly equivalent to finitely generated uniform quantum circuits.

We note that Definitions 6.2.5 and 6.1.3 do not make reference to any fixed universal gate set. A potential issue that arises in this regard is that the complexity class being defined may depend in an intricate way on the chosen universal gate set, since it may not be possible to perfectly reset every initial catalytic state under our uniformity and resource constraints. If we relax the notion of catalyticity to mean that the initial catalytic state only has to be reset to within $\epsilon$ trace distance at the end of the computation, one can use the Solovay-Kitaev theorem to see that every choice of gate set leads to the same complexity class in Definition 6.2.5. This interesting model resembles classical catalytic space classes with small errors in resetting, and we leave it as an open question to determine how it relates to the exact resetting model.

Returning to our setting that requires the quantum catalytic machine to perfectly reset the catalytic space to its initial state at the end of the computation, we will restrict our attention to the case of universal quantum gate sets that are infinite (for the complexity-theoretic properties of circuit families over such gate sets, see e.g. [NO02]). In this case, our definition is robust to the choice of gate set since any unitary may be implemented exactly by finite-sized circuits

over such gate sets. Consequently changing the gate set does not change the set of catalytic states that can be reset exactly by the machine. This results in well-defined catalytic complexity classes independent of the specific choice of gate set.

### 6.2.4 Uniformity

Similar to gate sets, the question of uniformity is quite subjective, as different uniformity conditions will lead to different levels of expressiveness for our machines.

**6.2.13.** DEFINITION. We fix the uniformity in Definition 6.2.1 to be $\mathcal{K} = \mathsf{SPACE}[\mathcal{O}(s)]$.

We choose $\mathsf{SPACE}[\mathcal{O}(s)]$ as it is the largest class of classical machines a $\mathsf{QCSPACE}[s,c]$ machine should seemingly contain by default. Thus we believe the choice of $\mathsf{SPACE}[\mathcal{O}(s)]$-uniformity is best suited to removing classical uniformity considerations from taking the forefront of the discussion regarding quantum catalytic space.

The question of how uniformity affects the power of $\mathsf{QCSPACE}$ is left to future work; we only comment briefly here on natural alternative choices. Perhaps the most immediate would be to consider $\mathsf{CSPACE}[s,c]$ uniformity, as it mirrors our quantum machine. As we will see later, it is not clear how to prove $\mathsf{QCSPACE}[s,c]$ contains $\mathsf{CSPACE}[s,c]$ directly, an interesting technical challenge that would be rendered moot by building it into the uniformity. Similarly we avoid $\mathsf{P}$-uniformity because it is not known, and even strongly disbelieved, that $\mathsf{CL}$ contains $\mathsf{P}$.

## 6.3 Main results

Now that we have a proper definition of the quantum catalytic model, we can state our main results. Our main technical contribution is to show that, somewhat surprisingly, quantum Turing machines and quantum circuits are equivalent even in the catalytic space setting:

**6.3.1.** THEOREM (Prove in Section 6.4.3). *Let $L \subseteq \{0,1\}^*$ be a language, and let $s := s(n)$ and $c := c(n)$. Then $L$ is computable by a quantum catalytic Turing machine with workspace $\mathcal{O}(s)$ and catalytic space $\mathcal{O}(c)$ iff $L$ is computable by a family of quantum catalytic circuits with workspace $\mathcal{O}(s)$ and catalytic space $\mathcal{O}(c)$.*

While this translation is straightforward in other settings, $\mathsf{QCL}$ has no *a priori* polynomial time bound, and so there is no obvious way to define the length of a catalyic circuit without running into trouble. However, we prove that the result of Buhrman et al. [Buh+14] which shows that $\mathsf{CL}$ takes polynomial time on average can be strengthened in the quantum case, to show that $\mathsf{QCL}$ *always* takes polynomial time without any error:

**6.3.2.** THEOREM (Prove in Section 6.4). $\mathsf{QCL} \subseteq \mathsf{EQP}$

Such a result is considered the "holy grail" of classical catalytic computing.

In terms of class containments, we focus on two questions: the relationship of quantum and classical catalytic space, and the relationship of catalytic space to the one-clean qubit model ($\mathsf{DQC_1}$), a pre-existing object of study in quantum complexity which bears a strong resemblance to catalysis. We show that, while $\mathsf{CL} \subseteq \mathsf{QCL}$ is surprisingly out of reach at the moment, this can be shown for an important subclass of $\mathsf{CL}$, one which captures the strongest known classical containment:

**6.3.3.** THEOREM. $\mathsf{TC}^1 \subseteq \mathsf{QCL}$

Prove in Section 6.5. As a consequence, we show that $\mathsf{TC}^1$ constitutes a natural class of functions for which catalysis gives additional power to quantum computation.

We also show that unitary $\mathsf{QCL}$ ($\mathsf{Q_UCL}$) and classical $\mathsf{CL}$ are both contained in $\mathsf{DQC_1}$:

**6.3.4.** THEOREM. $\mathsf{BQ_UCL} \subseteq \mathsf{DQC_1}$

Prove in Section 6.6.

**6.3.5.** THEOREM. $\mathsf{CL} \subseteq \mathsf{DQC_1}$

Prove in Section 6.6. Note that we use a version of $\mathsf{DQC_1}$ defined using a logspace controller instead of a polynomial time controller as may also be done. These results show how much of the power of $\mathsf{DQC_1}$ comes from avoiding the limitation of the resetting condition on the "dirty" workspace.

## 6.4   $\mathsf{QCL}$ upper bounds

In this section, we will finally return to the question of our quantum machine model, showing that Turing machines and circuits are equivalent. One major stepping stone is to show that log-space quantum catalytic Turing machines adhere to a polynomial runtime bound for *all* possible initializations of the catalytic tape.

Before all else, a remark is in order as to why such a restriction should hold for a seemingly stronger model, i.e. **QCLM**, when it is not in fact known for $\mathsf{CL}$. While quantum catalytic space has access to more powerful computations, i.e. quantum operations, it also has the much stronger restriction of resetting arbitrary density matrices rather than arbitrary bit strings. This restriction gives rise to a much stronger upper bound argument, and in fact rules out one of the main techniques available to classical Turing machines, namely compress or random arguments (see c.f. [Dul15; Coo+25]).

## 6.4.1 Polynomial average runtime bound

We begin by showing an analogue of the classical result of [Buh+14], i.e. the average runtime of a quantum catalytic machine for a random initial catalytic state $\rho$ is polynomial in the number of work qubits. We note that the runtime of a quantum Turing machine need not be a deterministic function of the input; $M$ has access to quantum states and intermediate measurements, from which it is possible to generate randomness which might influence the time that machine takes to halt.

**6.4.1.** DEFINITION. Given a quantum catalytic Turing machine $M$, a fixed input $x \in \{0,1\}^n$, and an initial catalytic tape $\rho$, we denote by $T(M, x, \rho)$ the distribution of runtimes of $M$ on input $x$ and initial catalytic tape $\rho$.

For an averaging argument to hold, we need to have a quantum notion of non-overlapping configuration graphs.

**6.4.2.** LEMMA. *Let $M$ be a quantum catalytic Turing machine, and let $\{\tau_i\}_i$ form an orthonormal basis for $D(\mathcal{H}_c)$. For all $i$ and $t$, let $\rho_{i,t}$ be the density matrix describing the state of the classical tape, quantum tape, and internal state of $M$ at time step $t$ on initial catalytic tape $\tau_i$. Then if $M$ is absolutely halting, all elements of the set $\{\rho_{i,t}\}_{i,t}$ are orthogonal.*

**Proof:**
We first consider the states $\rho_{i,t}$ for a fixed $i$. Assume instead that there exists some times $t$ and $t'$ where the states are not orthogonal. This means that the state at time step $t$ can be written as a superposition between the state in time step $t'$ and the state $\rho_{i,t} = p\rho_{i,t'} + (1-p)\eta$ for some $p > 0$. This forms a loop in the configuration graph where part of the state is back at time step $t'$. The amplitude of the part of the state in this loop will shrink over time, but never go to zero. The part of the state that is stuck in the loop will never reach the halting state, therefore this is in contradiction with the assumption that the quantum Turing machine is absolutely halting.

Next we consider the states $\rho_{i,t}$ for different $i$. By definition of a quantum Turing machine, the transformations $M$ can apply to the entire state of the machine is given by some quantum channel. By Lemma 6.1.2 we know that the trace distance between the entire state of the machine for separate instances of the catalytic tape can only decrease by this quantum channel. Therefore we know that if two instances start out to be orthogonal and end to be orthogonal, they have to remain orthogonal through the entire calculation. □

**6.4.3.** LEMMA. *Let $M$ be a quantum catalytic Turing machine with workspace $s$ and catalytic space $c$, let $\{\rho_i\}_i$ form an orthonormal basis for $D(\mathcal{H}_c)$, and define*

$T_{max}(M, x, \rho)$ *to be the maximum runtime of machine $M$ on input $x$ on starting catalytic tape $\rho$. Then*

$$\mathbb{E}_i[T_{max}(M, x, \rho_i)] \leq 2^{\mathcal{O}(s)}$$

**Proof:**

Our catalytic machine is defined by a $\mathsf{SPACE}[\mathcal{O}(s)]$ machine, defined by a tape of length $\mathcal{O}(s)$ and an internal machine of size $\mathcal{O}(1)$, which acts on $\mathcal{H}_s$ and $\mathcal{H}_c$, which can be addressed into using $\log s$ and $\log c$ bits respectively. Since these quantities plus the Hilbert spaces $\mathcal{H}_s$ and $\mathcal{H}_c$ define the dimensionality of our machine, by Lemma 6.4.2 we have that

$$\sum_{\rho \in \{\rho_i\}} T_{max}(M, x, \rho) \leq \mathcal{O}(2^{2(s+c+\mathcal{O}(s)+\mathcal{O}(1)+\log s+\log c)})$$

and therefore the lemma follows because $|\{\rho_i\}| \leq 2^{2c}$ and $2(s + \mathcal{O}(s) + \mathcal{O}(1) + \log s + \log c) = \mathcal{O}(s)$. $\qquad\square$

This already gives us a nice containment for our $\mathsf{QCSPACE}[s, c]$ classes.

**6.4.4.** COROLLARY. $\mathsf{QCLM} \subseteq \mathsf{ZQP}$

**6.4.5.** COROLLARY. $\mathsf{BQCLM} \subseteq \mathsf{BQP}$

## 6.4.2 Equal running times

We now take a further leap, showing that the initial catalytic tape does not affect the (distribution of the) runtime of our machine $M$ for a fixed input $x$.

**6.4.6.** DEFINITION. Let $M$ be a quantum catalytic Turing machine, and let $n \in \mathbb{N}$. We define $T_{max}(M, n)$ to be the maximum of the support of $T(M, x, \rho)$, maximized over $x$ and $\rho$.

We can first show that given $M$ and only one single copy of a state $\eta \in \mathcal{H}_c$, this probability distribution can be approximated up to arbitrary precision for any $x$.

**6.4.7.** LEMMA. *Given catalytic Turing machine $M$ and a single copy of a quantum state $\eta \in \mathcal{H}_c$, $T(M, x, \eta)$ can be approximated up to arbitrary precision for any $x$.*

**Proof:**

Because $M$ is a quantum catalytic Turing machine it has to reset the quantum state initialized in its catalytic tape perfectly. Therefore we can use the following approach: first fix some input $x$, then run the catalytic machine given $x$ as input and $\eta$ on its catalytic tape and record the running time. When the machine

halts, $\eta$ should be returned on the catalytic tape. This means the test can be performed again given the same inputs. This test can be run arbitrarily often giving an arbitrary approximation to $T(M, x, \eta)$. $\qquad\square$

This gives us the following observation about states with different halting times:

**6.4.8.** LEMMA. *Let $M$ be a quantum catalytic Turing machine, and let $\rho_1, \rho_2 \in D(\mathcal{H}_c)$. Assume there exists $x \in \{0, 1\}^n$ such that $T(M, x, \rho_1) \neq T(M, x, \rho_2)$. Then $||\rho_1 - \rho_2||_1 = 1$, where $|| \cdot ||_1$ is the trace distance.*

**Proof:**
The Helstrom bound states that the optimal success probability of any state discrimination protocol given one copy of an unnown state is:

$$P_{success} = \frac{1}{2} + \frac{1}{2} \cdot ||\rho_1 - \rho_2||_1$$

By Lemma 6.4.7, we know that $T(M, x, \rho)$ can be approximated to any precision with only one copy of $\rho$. Given a copy of either $\rho_1$ or $\rho_2$ at random, one can estimate $T(M, x, \rho)$ and perfectly discriminate between the cases $\rho = \rho_1$ and $\rho = \rho_2$ giving a protocol with $P_{success} = 1$. Therefore it follows that

$$\frac{1}{2} + \frac{1}{2}||\rho_1 - \rho_2||_1 = 1$$

and hence $||\rho_1 - \rho_2||_1 = 1$. $\qquad\square$

Lemma 6.4.8 is sufficient to show that the halting time of a quantum catalytic Turing machine is independent of the initial state in the catalytic tape:

**6.4.9.** THEOREM. *Let $M$ be a quantum catalytic Turing machine with $s$-qubit workspace and $c$-qubit catalytic space, and let $x \in \{0, 1\}^n$. Then there exists some value $t := t(n)$ such that $T(M, x, \rho) = t$ for all $\rho \in D(\mathcal{H}_c)$.*

**Proof:**
Assume for contradiction that there exist $\rho_1, \rho_2$ such that $T(M, x, \rho_1) \neq T(M, x, \rho_2)$. By Lemma 6.4.8 it holds that $||\rho_1 - \rho_2||_1 = 1$. Consider the state $\rho' = \frac{1}{2}\rho_1 + \frac{1}{2}\rho_2$, and note that only one of $T(M, x, \rho') \neq T(M, x, \rho_1)$ or $T(M, x, \rho') \neq T(M, x, \rho_2)$ can hold, by transitivity. Without loss of generality, let $T(M, x, \rho') \neq T(M, x, \rho_1)$, and so $||\rho' - \rho_1||_1 = 1$ by Lemma 6.4.8. However, by definition we have that

$$||\rho' - \rho_1||_1 = ||(\frac{1}{2}\rho_1 + \frac{1}{2}\rho_2) - \rho_1||_1 = \frac{1}{2}$$

which is a contradiction. $\qquad\square$

Putting theorem 6.4.3 and theorem 6.4.9 together immediately shows that the runtime of $M$ is bounded by a polynomial in $n$ for every input $x$ and initial catalytic state $\rho$:

**6.4.10.** THEOREM. *Let $M$ be a quantum catalytic Turing machine with workspace $s$ and catalytic space $c$. Then the maximum halting time is bounded by $2^{\mathcal{O}(s)}$.*

This strengthens Corollary 6.4.4 to remove the randomness in the output probability; this is the quantum equivalent of showing $\mathsf{CL} \in \mathsf{P}$, considered the holy grail of open problems in classical catalytic computing:

**6.4.11.** COROLLARY. $\mathsf{QCLM} \subseteq \mathsf{EQP}$

### 6.4.3   Turing machines and circuits

We finally prove Theorem 6.2.6 and show the equivalence of our two definitions of quantum catalytic machines. To do this, we observe, without proof, that Theorem 6.4.9 extends to any *classical observable feature* of the initial catalytic state by the same proof. We will apply this to one other aspect, namely the transition applied at a given timestep $t$:

**6.4.12.** LEMMA. *Let $M$ be a quantum catalytic Turing machine, and let $x \in \{0,1\}^n$. Then for every time $t$, there exists a fixed operation $g$ applied by $M$ at time $t$ for every $\rho \in \mathcal{H}_c$.*

This is sufficient to prove Theorem 6.2.6, which we will restate for convenience,

**6.2.6.** THEOREM. *For $s = \Omega(\log n), c = 2^{\mathcal{O}(s)}$*

$$\mathsf{QCSPACEM}[\mathcal{O}(s), \mathcal{O}(c)] = \mathsf{QCSPACEC}[\mathcal{O}(s), \mathcal{O}(c)]$$

$$\mathsf{BQCSPACEM}[\mathcal{O}(s), \mathcal{O}(c)] = \mathsf{BQCSPACEC}[\mathcal{O}(s), \mathcal{O}(c)]$$

**Proof:**
We only prove the equivalence between $\mathsf{QCSPACEC}$ and $\mathsf{QCSPACEM}$; the same proof applies to $\mathsf{BQCSPACEC}$ and $\mathsf{BQCSPACEM}$. Certainly $\mathsf{QCSPACEC}[s, c]$ is contained in $\mathsf{QCSPACEM}[\mathcal{O}(s), \mathcal{O}(c)]$, since $\mathsf{QCSPACEC}$ circuits are $\mathsf{SPACE}[\mathcal{O}(s)]$ uniform and can be directly simulated by a $\mathsf{QCSPACEM}$ machine.

Conversely, given a $\mathsf{QCSPACEM}[s, c]$ machine $M$, we wish to find an equivalent quantum catalytic circuit in $\mathsf{QCSPACEC}[\mathcal{O}(s), \mathcal{O}(c)]$. For this, we transform the transition function of the quantum Turing machine into a quantum channel; since the transition only takes a finite number of (qu)bits as input, this can always be done, and we have our transitions act on the same space $\mathcal{H}_s \otimes \mathcal{H}_c$ as $M$. Then, by using a method similar to that from the proof of Lemma 6.6.8, to make the machine oblivious, the tape head movement of the quantum Turing machine will be fixed. If our circuit is the transition function channel copied to all locations where the tape heads end up, we completely simulate the quantum Turing machine. We know that $T_{max}(M, n)$ is always at most $2^{\mathcal{O}(s)}$ for a machine $M$ by Theorem 6.4.10, and so the number of such transition function channels is also at most $2^{\mathcal{O}(s)}$. Therefore, we can simulate $M$ using a quantum circuit of length $2^{\mathcal{O}(s)}$ as claimed.                                    $\square$

# 6.5 Simulation of $\mathsf{TC}^1$

In this section we show that $\mathsf{QCL}$ can simulate Boolean threshold circuits. As in the classical world, the ability to simulate $\mathsf{TC}^1$ is also a reason to believe that catalytic logspace is strictly more powerful than logspace. This follows from the fact that $\mathsf{QL} = \mathsf{PL}$ [Wat98], which is itself contained in $\mathsf{TC}^1$:

**6.5.1.** LEMMA. $\mathsf{QL} \subseteq \mathsf{TC}^1$

Since $\mathsf{TC}^1$ can compute powerful functions such as determinant, this containment is largely believed to be strict. Thus Theorem 6.3.3 gives us a candidate class of problems for separating $\mathsf{QL}$ from $\mathsf{QCL}$.

## 6.5.1 Reversibility and obliviousness

In [Buh+14] the authors showed that $\mathsf{TC}^1$ can be simulated by *transparent register programs*, which themselves are computable in $\mathsf{CL}$; thus our goal is to extend the $\mathsf{CL}$ simulation of transparent programs to $\mathsf{QCL}$. More broadly, we show that *reversible, oblivious, time-bounded* $\mathsf{CL}$ is enough to simulate transparent programs, and such a model is structured enough that, while we cannot show that all of $\mathsf{CL}$ is in $\mathsf{QCL}$, we can at least prove the containment for this small fragment.

We first make the following definitions which we use for our simulations. We begin by recalling a result of Dulek [Dul15] which shows that catalytic Turing machines can be made *reversible* (see c.f. [Coo+25] for a proof)

**6.5.2.** THEOREM. *For every catalytic machine $M$ with space $s$ and catalytic space $c$, there exist catalytic machines $M_\rightarrow$, $M_\leftarrow$ with space $s+1$ and catalytic space $c$ such that for any pair of configurations $(\tau_1, v_1)$, $(\tau_2, v_2)$ of $M_\rightarrow$ and $M_\leftarrow$, if $M_\rightarrow$ transitions from $(\tau_1, v_1)$ to $(\tau_2, v_2)$ on input $x$, then $M_\leftarrow$ transitions from $(\tau_2, v_2)$ to $(\tau_1, v_1)$ on input $x$.*

We will also need to consider *oblivious* machines, i.e. ones where the tape head movement is solely a function of the input length $|x|$ and does not depend at all on the content of the catalytic tape c. While any Turing machine can be made oblivious, it requires relaxing the definition of obliviousness to not forcing the machine to halt at the same time on every input; we simply require that every machine that continues to run carries out its execution in an oblivious manner. We will bar this restriction in this section.

**6.5.3.** DEFINITION. We say that a $\mathsf{CL}$ machine is *totally oblivious* if the following holds. Let $t, q, h$ be special registers on the free work tape, all initialized to 0, representing the time, state, and tape heads of the machine. At each point in time our machine consist of one *mega-step*: for every setting of $t, q, h$ there is a fixed transformation, computable in logspace, which the machine applies to the

catalytic tape and to $q, h$, and a mega-step consists of applying each of these operations, conditioned on the values of $t, q, h$ on the free work tape, in order. At the end of every mega-step we increment $t$, and our machine halts iff $t$ reaches a predetermined step $T$.

Totally oblivious machines are ones that in essence apply the same bundle of transformations at every time step, with the information about which one to to actually apply being written on the free work tape, and the halting behavior being determined only by the clock.

Such machines are clearly in poly-time bounded CL (see c.f. [Coo+25] for a discussion of this class), since the clock must fit on the free work tape. This causes issues when we seek total obliviousness in tandem with reversibility; in general it is not known, and is highly unlikely, that a polynomially time-bounded Turing machine can be made reversible while remaining polynomially time-bounded.

However, there is an important class of algorithms which is both reversible and totally oblivious: *clean register programs.* For our purposes we will use a very restricted version of clean register programs (see c.f. [Mer23] for a discussion).

**6.5.4. DEFINITION.** A *register program* $\mathcal{P}$ is a list of instructions $P_1 \ldots P_t$ where each $P_i$ either has the form $R_j \mathrel{+}= x_k$ for some input variable $x_k$ or has the form $R_j \mathrel{+}= q_i(R_1 \ldots R_m)$ for some polynomial $q_i$. A register program *cleanly computes* a value $v$ if for any initial values $\tau_1 \ldots \tau_m$, the net result of running $\mathcal{P}$ on the registers $R_1 \ldots R_m$, where each $R_j$ is initialized to the value $\tau_j$, is that $R_1 = \tau_1 + v$ and $R_j = \tau_j$ for all $j \neq 1$.

If we think of these registers as being written on the catalytic tape, it is clear that clean register programs are totally oblivious, as the instruction at every moment in time is based only on the timestep. This is nearly immediate, although we note a few minor complications here. We need to preprocess the catalytic tape to ensure our registers have values over the same ring as our register program; for example, if we represent numbers mod $p$ using $\lceil \log p \rceil$ bits, some initial values will exceed $p$. This can be handled obliviously by observing that for either $\tau$ or $\bar{\tau}$, half the registers are already correct, and so we take one full pass over $\tau$ to keep a count of which case we are in, store this as a bit $b$ (1 iff we need to flip $\tau$), and XOR $\tau$ with $b$ at the beginning and end of the computation. We subsequently ignore all blocks which are initialized to improper values; when we go to operate on register $R_j$, say, as we obliviously pass over the whole catalytic tape we will count how many *valid* registers we have seen, and act only when we see the counter reach $j$.

Besides being totally oblivious, however, such programs are also *reversible*, as every step of the form $R_j \mathrel{+}= c$ can be inverted by a step of the form $R_j\mathrel{-}= c$. Thus such programs appear highly constrained in terms of what they can and cannot achieve. Nevertheless, such programs are sufficient to compute $\mathsf{TC}^1$.

**6.5.5.** LEMMA ([Buh+14]). *Let $L$ be a language in* $\mathsf{TC}^1$. *Then $L$ can be decided by a clean register program, and, hence, by a totally oblivious reversible* $\mathsf{CL}$ *machine.*

### 6.5.2 Simulation by $\mathsf{QCL}$ machines

We now show that reversibility plus total obliviousness is sufficient for simulation by $\mathsf{QCL}$.

**6.5.6.** LEMMA. *Let $L$ be a language which can be computed be a totally oblivious reversible* $\mathsf{CL}$ *machine. Then $L \subseteq \mathsf{QCL}$.*

**Proof:**
Let $M$ be a totally oblivious reversible $\mathsf{CL}$ machine. We will treat our quantum catalytic tape as a superposition over classical catalytic tapes, i.e. a superposition over computational basis states. It is thus sufficient to show that the operation of machine $M$ can be simulated by a fixed quantum circuit containing Toffoli gates, as such a circuit will correctly operate on each of our catalytic basis states in each branch of the superposition.

By total obliviousness, every step that $M$ takes is a fixed transformation conditioned on the value of $t$, $q$, and $h$; since we additionally know that such a step is reversible, it must be isomorphic to a Toffoli gate applied to a fixed position of the catalytic tape conditioned on some fixed mask applied to $t$, $q$, and $h$, and furthermore each transformation can be computed by our logspace controlling machine. Since these operations are fixed for each timestep, we can move $t$ to our space controlling machine and have it construct a circuit, comprised of Toffoli gates on $q$, $h$, and the catalytic tape, of polynomial length. □

This is sufficient to prove our main result for this section:

**6.3.3.** THEOREM. $\mathsf{TC}^1 \subseteq \mathsf{QCL}$

**Proof:**
Combine Theorem 6.5.5 with Theorem 6.5.6. □

## 6.6 Simulating catalytic space in $\mathsf{DQC_1}$

Lastly we will discuss the relationship between catalytic computing and a pre-existing yet closely related quantum model, namely the one clean qubit setting. We will introduce the model and then prove that it can simulate unitary $\mathsf{QCL}$. Finally we will show that classical $\mathsf{CL}$ is also contained in the one clean qubit model.

## 6.6.1 One clean qubit model

In the one-clean qubit model, first introduced by Knill and Laflamme [KL98], a quantum machine is given a single input qubit initialized in the zero state and $n$ qubits initialized in the maximally mixed state. We will formalize the definition of this computational model:

**6.6.1.** DEFINITION (One clean qubit). Let $\{Q_x\}_x$ be a log-space uniform family of unitary quantum circuits. The *one clean qubit model* is a model of computation in which $Q_x$ is applied to the $n + 1$-qubit input state

$$\rho = |0\rangle\langle 0| \otimes \frac{I_n}{2^n},$$

where $n = |x|$ and $I_n$ operator is the identity on $n$ qubits. After execution of $Q_x$ the first qubit is measured, giving output probabilities:

$$p_0 = 2^{-n} \operatorname{Tr}[(|0\rangle\langle 0| \otimes I)Q_x(|0\rangle\langle 0| \otimes I)Q_x^\dagger],$$
$$p_1 = 1 - p_0$$

**6.6.2.** REMARK. Two points stand out in this definition. First, note that $Q_x$ are unitary circuits, and hence do not allow intermediate measurements; such measurements would allow for resetting the qubits initialized in the maximally mixed state, making the model significantly stronger. Second, in this paper we consider log-space uniform families of unitary circuits, rather than the more common deterministic polynomial-time uniform families, in order to align more closely with the QCL model that we study.

The one-clean qubit model is a probabilistic model of computation, and hence we typically talk about computing a function $f(x)$ in terms of success probability for computing $f(x)$ being bounded away from $1/2$. The exact bound on the error probability does not matter; while we often use $2/3$ in defining e.g. BQP, even a $1/poly(n)$ gap is sufficient as there we can employ standard error-correction to boost our success, namely by running the algorithm multiple times and taking a majority vote of the outcomes. However, this is not known to be possible in the one-clean qubit model, as such a machine can only reliably run once.

**6.6.3.** DEFINITION ([KL98; She06]). $\mathsf{DQC}_1$ is the set of all languages $L =\subset \{0,1\}^*$ for which there exists a one-clean qubit machine $M$ and a polynomial $q(n)$, such that on input $x \in L$ of length $n = |x|$,

- if $x \in L_{yes}$ then the output probability $p_1 \geq \frac{1}{2} + \frac{1}{q(n)}$

- if $x \in L_{no}$ then the output probability $p_0 \geq \frac{1}{2} + \frac{1}{q(n)}$

On the other hand, somewhat surprisingly, the one-clean qubit model is robust to the number of clean qubits allowed, up to a logarithmic number:

**6.6.4.** LEMMA ([SJ08]). $\mathsf{DQC}_k = \mathsf{DQC}_1$ *for* $k = \mathcal{O}(log(n))$, *where* $\mathsf{DQC}_k$ *means having access to $k$ clean qubits instead of one.*

## 6.6.2   Containment of unitary $\mathsf{QCL}$ in $\mathsf{DQC_1}$

We now move on to establishing a formal connection between $\mathsf{QCL}$ and $\mathsf{DQC_1}$. A $\mathsf{QCL}$ machine is allowed to apply intermediate measurements to its quantum tape as well as its catalytic tape, which is not possible in $\mathsf{DQC_1}$; however, if we restrict the $\mathsf{QCL}$ machine to not make any intermediate measurements we can show that such a machine can in fact be simulated by the one-clean qubit model.

**6.6.5.** DEFINITION ($\mathsf{Q_UCL}$). A $\mathsf{Q_UCL}$ machine is a $\mathsf{QCL}$ machine in which the quantum circuit is unitary. In the final step, the unitary the $\mathsf{Q_UCL}$ machine measures the first qubit, which then gives the outcome of the calculation. Similarly we define $\mathsf{BQ_UCL}$ to be $\mathsf{BQCL}$ with the unitary restriction.

Using this definition we can give the following proof of containment:

**6.3.4.** THEOREM. $\mathsf{BQ_UCL} \subseteq \mathsf{DQC_1}$

**Proof:**
Let $C$ be a log-space uniform $\mathsf{BQ_UCL}$ quantum channel. Since $C$ is unitary up until the last measurement step, it preserves all possible density matrices from the catalytic tape, and in particular it preserves the maximally mixed state $I_n$. Let $U$ be the unitary part of $C$. The action of $U$ on the work-tape and the catalytic tape, with the catalytic tape initialized in $I_n$, is:

$$U \left|0\right\rangle \left\langle 0\right|_w \otimes \frac{I_n}{2^n} U^\dagger = (\sqrt{p_0} \left|0\right\rangle \left\langle 0\right|_{w_0} \left|\psi_0\right\rangle \left\langle \psi_0\right|_w + \sqrt{p_1} \left|1\right\rangle \left\langle 1\right|_{w_0} \left|\psi_1\right\rangle \left\langle \psi_1\right|_w) \otimes \frac{I_n}{2^n}$$

with $|p_1| \geq 2/3$ in a 'yes' instance and $|p_0| \geq 2/3$ in a 'no' instance. Note that this calculation is of the exact form of a $\log(n)$-clean qubit machine and that the output probabilities are a constant bounded away from $1/2$; hence this problem is in $\mathsf{DQC_k}$, and by Lemma 6.6.4 is therefore in $\mathsf{DQC_1}$ $\qquad\square$

## 6.6.3   Containment of $\mathsf{CL}$ in $\mathsf{DQC_1}$

We aim to show that $\mathsf{CL} \subseteq \mathsf{DQC_1}$. The idea is that $\mathsf{CL}$, as per Theorem 6.5.2, can always be made reversible. While as discussed before we cannot maintain reversibility and total obliviousness, a $\mathsf{CL}$ machine can also always be made 'almost oblivious' while maintaining reversibility; the tape head movements are independent of the input, but the machine does not know when to halt. Instead, after any given amount of time, we know that the machine has halted on a fraction $1/poly(n)$ of possible initial catalytic states. Since the $\mathsf{DQC_1}$ model can be interpreted as sampling from a uniform distribution of computational basis states, this shows the probability of finding the correct output is $1/2 + 1/poly(n)$, which is sufficient for the proof.

**6.6.6.** DEFINITION. A *non-halting reversible oblivious* catalytic Turing machine is a reversible oblivious catalytic Turing machine that need not halt absolutely. In particular, for every input $x$ and initial catalytic state $c$ there exists a time $t(x, c)$ where the correct output has been written to the output tape and the catalytic tape has been reset to its initial state. In addition, the output state has an additional binary cell that indicates whether or not the output has been determined yet, or is still 'unknown' by the machine.

**6.6.7.** DEFINITION. We say a reversible oblivious catalytic Turing machine *halts with polynomial success probability* if there exists polynomials $p, q$ such that for any valid input $x$ to a promise problem, after time $p(|x|)$ the output tape of the catalytic Turing machine contains the correct output to the problem on a fraction of at least $1/q(|x|)$ when the initial catalytic tapes are taken uniformity at random. After time $p(|x|)$, the output tape of the catalytic Turing machine never contains the wrong answer, but it may leave the output undetermined.

We show that any CL machine can be transformed into a reversible oblivious catalytic Turing machine that halts with polynomial success probability.

**6.6.8.** LEMMA. *Any catalytic Turing machine $M$ that has a logarithmic clean space and polynomial size catalytic tape can be turned into a non-halting oblivious reversible catalytic Turing machine $M^o$ with a logarithmic clean tape and polynomial catalytic tape.*

**Proof:**
By [Dul15; Coo+25], $M$ can always be assumed to be reversible. We claim we can also make $M$ oblivious by sacrificing the condition that $M$ is absolutely halting. This also interferes with what is meant by the machine being catalytic, but the new machine no longer needs to be catalytic.

To make the machine oblivious, we make two modifications. The first applies to operations on the clean tape. The second applies to operations on the catalytic tape. On the clean tape, we double the size of the clean tape of $M$, breaking it up into pairs. The first entry of the pair stores the original data while the second keeps track of the position of where the tape head is 'supposed' to be. Then by iterating over all positions on the clean tape of the Turing machine in every step of the original Turing machine, operations on the clean tape of the Turing machine can be made oblivious. Similarly, for operations on the catalytic tape, we can maintain an additional part of the clean tape that keeps track of the position of the catalytic tape head position. By iterating over all possible positions of the catalytic tape head and checking if the tape head is 'really there', we can make catalytic tape operations oblivious.                                                      □

We call the machine formed this way $M^o$ for oblivious $M$. Since the catalytic and clean tape are no more than polynomial length, this procedure adds at most

a polynomial factor to the runtime. However, since the runtime of $M$ may be super-polynomial and an oblivious machine has the same runtime for all inputs $x$ of the same length and catalytic tapes $c$, the machine does not have enough clean space to keep a clock to know whether or not it has terminated. This means we cannot assume it halts. However, we can show that it is halting with sufficient probability:

**6.6.9.** LEMMA. *For any language L in* CL *that is recognized by a catalytic Turing machine M, there exists a reversible oblivious catalytic Turing machine N that halts with polynomial probability that also recognizes L. Furthermore, N also uses* $\mathcal{O}(\log |x|)$ *clean space and polynomial catalytic space.*

**Proof:**
We observe that $M^o$ in Lemma 6.6.8 is simulated step-by-step, meaning that not only do we reach the same outcome, but up to a fixed transformation and a slower runtime, $M^o$ passes through the same intermediate states. If we consider a modified version of $M$ in the first place, we can ensure that the machine halts with polynomial success probability. We modify the original machine $M$ to form the machine $M'$ in the following ways:

1. The machine $M'$ repeats the original machine $M$ $l(n) = 2^{s(n)}$ times where $s(n)$ is the length of the clean tape. Each repetition is called a cycle.

2. The space used for writing the output of $M$ originally is extended by one additional bit. This bit starts out in 0, signifying the output is 'undetermined'. These two bits together are called the output state.

3. After each cycle, the machine cleans itself. This means that it resets the clean space to the all 0 state, reversing the computation except the output state, which is left unaltered except for the first cycle.

4. After the first cycle, the correct output of the computation is written to the output tape and the second bit of the output state is flipped, signifying the output is 'determined'. For every subsequent cycle, a counter that counts up to $2l(n)$ is incremented by 1, but the output tape is left unchanged.

We claim that the machine $M'^o$ halts with polynomial probability. Suppose that the runtime of a cycle of $M'$ on input $x$ and catalytic tape $c$ is $f(x, c)$. The expected runtime of a cycle of $M'$, $f(x)$, over a uniform distribution of catalytic tapes for fixed $x$ is at most $l(n)$ by close analysis of the polynomial expected time bound given in [Buh+14]. Let us examine the output at time $t(x) = t(|x|) = l(n) + 1$. Then, by Markov's inequality on a uniform distribution

of possible catalytic tapes

$$\mathbb{P}(\text{output undetermined}) \leq \frac{f(x)}{t(x)}$$
$$\leq \frac{l(n)}{t(x)}$$
$$= \frac{l(n)}{l(n)+1} = 1 - \frac{1}{l(n)+1}$$

This means that at time $t(n)$, the probability that an output is written to the clean tape is at least $1/(l(n) + 1) = 1/poly(n)$. After the first cycle, the output must be correct. However, afterwards, we have no control over what is written onto the output tape. In making the machine reversible and oblivious, it may later change the value in the output tape, including incorrect values. This is why we repeat each cycle many times. This is $M'$ stalling to preserve the correctness of the output. Since the cycle is repeated $2l(n)$ times and each cycle uses time at least one time step to increment the counter, this means that at time $t(x) = l(n) + 1$ the value in the output tape is guaranteed to be correct or undetermined.

Let $N = M'^o$. Then $N$ is reversible, oblivious and halts with polynomial probability. Since $t(x) = l(n) + 1$ and $l(n)$ or any upper bound on $l(n)$ (which is sufficient) is readily computable, this completes the proof. $\square$

This completes all technical components necessary to show that $\mathsf{CL} \subseteq \mathsf{DQC_1}$.

**6.3.5.** THEOREM. $\mathsf{CL} \subseteq \mathsf{DQC_1}$

**Proof:**
The maximally mixed state of $\mathsf{DQC_1}$ can be interpreted as uniformly randomly sampling computational basis states. If we take these basis states to be the catalytic tape and use the fact that $\mathsf{DQC_1}$ is unchanged if we allow a logarithmic number of clean qubits, then we can run the machine $N$ from Lemma 6.6.9 by using unitary gates instead of reversible, oblivious operations. When we measure the output bit at the end, we get either an undetermined state or the correct output with certainty. If we get an undetermined state, we output a random bit and thus output the correct answer with probability $1/2$. If not, then we output the correct answer, which occurs with probability at least $1/poly(n)$. $\square$

## 6.7 Open problems

Finally, we identify a number of interesting avenues to further explore the power of quantum catalytic space, and understand its relation to various (quantum) complexity classes.

**QCL subroutines.**   Remarkably, classical catalytic subroutines can already be used to achieve analogous space savings in QCL. Is it possible to identify genuinely quantum subroutines to achieve savings beyond those attained by classical generalizations? This is not so straightforward because the subset of qubits being reused in a catalytic subroutine could become entangled with qubits that cannot be accessed by the subroutine. Therefore, there might be a non-trivial and inaccessible reference system with respect to which the catalytic property must hold. While we show the presence of such an inaccessible reference system does not change the model we define, designing quantum catalytic subroutines (cf. classical results in [CM24; Wil25]) stands out as a fertile direction for future work.

**Robust QCL.**   As discussed in Chapter 5, quantum computations are inherently error-prone. In our definition of QCL, we did not account for possible errors affecting the catalyst. This omission causes the definition of QCSPACE to depend on the choice of gate set. In contrast, the strict reset requirement led both to an equivalence between the different sets of states used to initialize the catalytic tape and to the time bound established for QCL. It would be interesting to investigate whether these properties persist when the reset condition is relaxed to allow a small trace distance error, potentially removing the gate-set dependence. Another direction is to explore whether the techniques developed in Chapter 5 extend to the quantum setting. In this case, rather than focusing on trace distance, we would be concerned with low-weight Pauli errors, akin to those typically addressed in quantum error correction.

**CL vs QCL.**   While we have started investigating the question, we still have no simple answer to the relationship between CL and QCL; in fact we have not even ruled out that CL contains QCL. The primary challenge is that while any CL machine runs in polynomial time in expectation over the catalytic tape, QCL machines *always* run in polynomial time. We do not know how to fit in pathological cases where CL runs in exponential time, for example, into QCL. Similarly, a problem or oracle that can separate QCL from CL would also be of interest[2].

**QNC[1] vs QCL.**   Starting with Barrington's Theorem [Bar89], a landmark result in space complexity, a classical line of work [BC92; Buh+14] has shown that polynomial-size formulas over many different gate sets can be computed using only logarithmic space, using a reversible, algebraic characterization of computation.

---

[2]A candidate oracle for showing a separation between QCL and CL is the oracle relative to which CL and PSPACE are equivalent, as shown in [Buh+14]. This oracle uses the fact that the initial catalytic tapes of CL are either compressible or random, using the oracle differently for either situation. This type of adaptive usage of the oracle, based on the given catalytic state, seems not to translate to the quantum setting due to Theorem 6.4.9.

Such a result in the quantum case, i.e. $\mathsf{QNC}^1 \subseteq \mathsf{QL}$, appears far out of reach, as this would imply e.g. novel derandomizations in polynomial time. However, such techniques are also key to the study of catalytic computation, and so perhaps we can show $\mathsf{QNC}^1$ or a similar quantum circuit class is contained in $\mathsf{QCL}$. This would give a clear indication of the power of quantumness in catalytic computation.

**QCL vs DQC$_1$.**   While we seem to find that $\mathsf{Q_UCL}$ or $\mathsf{QCL}$ without intermediate measurements is contained in $\mathsf{DQC}_1$, it is unclear if this still holds when we allow intermediate measurements.

# Part Three

# Guiding states

# Chapter 7
## Guided local Hamiltonian problem

In the final part of this thesis, we investigate a problem central to quantum chemistry: estimating the smallest eigenvalue (ground state energy) of a local Hamiltonian. To this end, we introduce the notion of a *semi-classical guiding state* — a quantum state that admits an efficient classical description and has non-trivial overlap with the ground space of a local Hamiltonian. We begin by motivating and introducing the problem, as well as the role of guiding states in addressing it.

Quantum chemistry and quantum many-body physics are generally regarded as two of the most promising application areas of quantum computing [Aar09; Bau+20]. Whilst perhaps the original vision of the early pioneers of quantum computing was to simulate the *time-dynamics* of quantum systems [Ben80; Fey82], for many applications one is interested in *stationary* properties. One particularly noteworthy quantity is the *ground state energy* (which corresponds to the smallest eigenvalue) of a local Hamiltonian describing a quantum mechanical system of interest, say a small molecule or segment of material.

The precision to which one can estimate the ground state energy plays a crucial role in practice: for instance, in chemistry the relative energies of molecular configurations enter into the exponent of the term computing reaction rates, making the latter exceptionally sensitive to small (non-systematic) errors in energy calculations. Indeed, to match the accuracy obtained by experimentation for such values one aims for an accuracy that is smaller than so-called *chemical accuracy*, which is about 1.6 millihartree.[1] This quantity – which reads as a constant – is defined with respect to a (physical) Hamiltonian whose norm grows polynomially in the system size and particle dimension, and thus chemical accuracy is in fact a quantity that scales inverse polynomially in the system size when one considers (sub-)normalized Hamiltonians, which is often the case in the quantum computing / Hamiltonian complexity literature.

---

[1] This quantity, which is ≈1 kcal/mol, is chosen to match the accuracy achieved by thermochemical experiments.

# 7.1   Local Hamiltonian problem

The problem of estimating the ground state energy of a Hamiltonian is formalized in complexity theory. A physical Hamiltonian is understood to be a Hamiltonian that only has local interactions, therefore, one considers what is known as a *local Hamiltonian.*

**7.1.1.** DEFINITION (Local Hamiltonian). A $k$-local Hamiltonian $H$ acting on $n$-qubits is a hermitian matrix that can be written as a sum of $T = poly(n)$ local terms:

$$H = \sum_{i=0}^{T-1} h_i,$$

where every $h_i$ is a $k$-local observable, in other words, a hermitian matrix that only acts non-trivially on at most $k$-qubits.

**7.1.2.** REMARK. The terms $h_i$ can always be further broken down into a linear combination of Pauli strings, where the highest weight string is at most the locality of $h_i$. Furthermore, in physics the Hamiltonians considered are often spatially bounded, meaning that there is some topology to which the qubits adhere, say on a line, and interactions are limited to be between direct neighbors. Note that we do not restrict the terms $h_i$ to be spatially bounded.

The decision variant of estimating the smallest eigenvalue of a local Hamiltonian up to some additive error is known as the *local Hamiltonian problem.*

**7.1.3.** DEFINITION (Local Hamiltonian problem). The      $k$-local      Hamiltonian problem $\mathsf{LH}(k, \delta)$ is:
**Input:** A $k$-local Hamiltonian $H$ with $||H|| \leq 1$ acting on $n$ qubits and threshold parameters $a, b \in \mathbb{R}$ such that $b - a \geq \delta > 0$
**Promise:** The ground state energy is promised to be $\lambda_0(H) \leq a$ or greater than $\lambda_0(H) \geq b$
**Output:**

- if $\lambda_0(H) \leq a$, output yes.

- if $\lambda_0(H) \geq b$, output no.

The local Hamiltonian problem is well-known to be $\mathsf{QMA}$-hard when the required accuracy $\delta$ scales inversely with a polynomial, where $\mathsf{QMA}$ is the quantum analogue of the class $\mathsf{NP}$, also known as Quantum Merlin Arthur. Therefore, it is generally believed that, without any additional help or structure, quantum computers are not able to accurately estimate the smallest eigenvalues of general local Hamiltonians, and there is some evidence that this hardness carries over to those Hamiltonians relevant to chemistry and materials science [OGo+22]. A natural question to ask is then the following:

> How much 'extra help' needs to be provided in order to accurately estimate ground state energies using a quantum computer?

## 7.2   Guiding states

In the quantum chemistry community, it is often suggested that this extra help could come from a classical heuristic that first finds some form of *guiding state*: a classical description of a quantum state that can be used as an input to a quantum algorithm to compute the ground state energy accurately [Liu+22]. Concretely, this comes down to the following two-step procedure [GL22]:

- Step 1 (Guiding state preparation): A classical heuristic algorithm is applied to obtain a *guiding state* $|u\rangle$, which is hoped to have 'good'[2] fidelity with the ground space.

- Step 2: (Ground state energy approximation): The guiding state $|u\rangle$ is used as input to Quantum Phase Estimation (QPE) to efficiently and accurately compute the corresponding ground state energy.

There is something special about Step 2 — it is a unique strength of quantum computers to be able to resolve an eigenvalue (within additive $1/poly(n)$ precision) of a local Hamiltonian given just a guiding state $|u\rangle$ to the corresponding eigenvector (via QPE).

Secondly, in general, one does *not* expect a good[3] guiding state for arbitrary local Hamiltonian $H$ to exist, as this would imply QCMA = QMA. And even if such a guiding state *did* exist, finding it can still be hard. For example, minimizing $\text{tr}(H\rho)$ over the "simplest" quantum ansatz of tensor product states, i.e. $\rho = \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n$ for $\rho_i \in D(\mathbb{C}^2)$, remains NP-hard (seen by letting $H$ be a diagonal Hamiltonian encoding a classical 3-SAT instance).

## 7.3   Guided local Hamiltonian problem

The second step of the above-mentioned procedure was first formally studied by Gharibian and Le Gall[GL22]. They introduced the *Guided k-local Hamiltonian problem (*GLH$(k, \delta, \zeta)$*)*, which can informally be stated as follows (with formal Definition 7.5.4): given a $k$-local Hamiltonian $H$, an appropriate classical 'representation' of a guiding state $|u\rangle$ promised to have $\zeta$-fidelity with the ground space

---

[2]'Good' here means at least inverse polynomial in the number of qubits the Hamiltonian acts on.

[3]"Good" here meaning a state $|\psi\rangle$ with inverse polynomial fidelity with a ground state, and with a succinct classical description allowing $|\psi\rangle$ to be prepared efficiently.

of $H$, and real thresholds $b > a$ (s.t. $b - a \geq \delta$), decide if the ground state energy of $H$ lies above or below the interval $[a, b]$ . In their work they show two results with respect to $k - \mathsf{GLH}(k, \delta, \zeta)$:

- For any constant $k$, $\mathsf{GLH}(k, \delta, \zeta)$ can be efficiently solved *classically* within *constant* precision, i.e. for $\delta \in \Theta(1)$ and $\zeta \in \Theta(1)$.

- In contrast, $\mathsf{GLH}(6, \delta, \zeta)$ is $\mathsf{BQP}$-hard for *inverse polynomial* precision, i.e. $\delta \geq 1/\,poly(n)$, and $\zeta = 1/2 - 1/\,poly(n)$.

The latter regime of inverse-polynomial precision turns out to be the relevant one for solving quantum chemistry problems in practice.

Four important problems were left open in [GL22]:
Is $\mathsf{GLH}(k, \delta, \zeta)$ still $\mathsf{BQP}$-hard when the guiding state has larger fidelity $\zeta$ with the ground space[4], and in particular for $\zeta$ arbitrarily close to 1?
Is $\mathsf{GLH}(k, \delta, \zeta)$ still $\mathsf{BQP}$-hard for locality $k < 6$?
How difficult is $\mathsf{GLH}(k, \delta, \zeta)$ when we are interested in estimating excited state energies instead of the ground state energy?
Are there physically motivated Hamiltonians for which $k$-GLH is still $\mathsf{BQP}$-hard?

In this part of the thesis, we continue the agenda toward Step 2 above by resolving the first three open questions:

- First, we show that $\mathsf{BQP}$-hardness continues to hold even for $\zeta = 1 - 1/\,poly(n)$ (in the regime of inverse polynomial precision), i.e. even when we are promised the guiding state $|u\rangle$ is a remarkably good approximation to the ground state.

- Second, we show that $\mathsf{BQP}$-hardness continues to hold even for $k = 2$. (Note that for $k = 1$, the problem can be solved efficiently classically, even without a guiding state.)

- Third, we extend the $\mathsf{BQP}$-hardness results to the case when one is interested in estimating energies of excited states, rather than just the ground-state. To do so we introduce the *Guided Local Hamiltonian Low Energy problem* ($\mathsf{GLHLE}(k, c, \delta, \zeta)$ formal definition 7.5.5), where one is asked to estimate the $c$'th eigenvalue of $H$ given an appropriate guiding state. Interestingly, we are only able to show $\mathsf{BQP}$-completeness in this setting by showing that the first point holds, i.e. the $\mathsf{BQP}$-hardness in the regime $\zeta \in [\frac{1}{2} + \Omega(1/poly(n)), 1 - \Omega(1/poly(n))]$.

The fourth question will not be discussed in this thesis. However, in [Cad+23a] hardness of this problem was proven for *physically motivated* Hamiltonians. They include XY model (constraints of the form $XX + YY$), Heisenberg model (constraints of the form $XX + YY + ZZ$), the antiferromagnetic $XY$ model and the antiferromagnetic Heisenberg model (i.e. "Quantum Max Cut" [GP19]).

---

[4]A local-Hamiltonian does not need to have a unique groundstate, therefor we require fidelity with the entire groundspace, and not with a specifick state.

## 7.4 Main result

We can summarize our result in the following theorems:

**7.4.1.** THEOREM (BQP-hardness). *For any fidelity $\zeta \in (0, 1-1/poly(n))$, locality $k \geq 2$, some integer $0 \leq c \leq \mathcal{O}(poly(n))$ (c'th excited state), and precision $\delta = 1/poly(n)$, GLHLE$(k, c, \delta, \zeta)$ (guided local Hamiltonian low energy problem) is BQP-hard.*

The proof of this theorem follows from Propositions 7.7.1, 7.7.3 and 7.7.8.
   Furthermore, the problem is contained in BQP when:

**7.4.2.** THEOREM (Containment in BQP of GLHLE).

   (i) *GLHLE$(k, 0, \zeta, \delta)$ is contained in BQP for $k = \mathcal{O}(\log(n))$, $\zeta = \Omega(1/poly(n))$, and $\delta = \Omega(1/poly(n))$.*

   (ii) *GLHLE$(k, c, \zeta, \delta)$ for $c \geq 1$ is contained in BQP when $k = \mathcal{O}(\log(n))$, $\zeta = \frac{1}{2} + \Omega(1/poly(n))$, and $\delta = \Omega(1/poly(n))$.*

The reason for the separation of Theorem 7.4.2 into parts (i) and (ii) is as follows: when the fidelity of the guiding state with the target eigenstate is sufficiently above $1/2$ (in this case $1/2+1/poly(n)$), then by inputting this state to quantum phase estimation and measuring, one can choose the most frequently observed output to be the estimate of the energy of the target state (since we know that the fidelity with any other eigenstate will be smaller than the fidelity with the target state). On the other hand, if the fidelity is not sufficiently above $1/2$, then it might be the case that the guiding state has significant fidelity with other eigenstates, and that the energies of these states will be measured with equal or higher probability than that of the target eigenstate. In this case, it is impossible to decide (in polynomial time) which energy corresponds to the target state, and which to the other, unwanted states, unless the target state is the groundstate (case (i)), in which case we can employ the variational principle and simply choose the smallest energy. A formal proof of Theorem 7.4.2 can be found in Section 7.8.

## 7.5 Preliminaries

As we saw in the Section 7.1, the local Hamiltonian problem is defined with respect to a promise on its input. Therefore, all complexity classes in this part of the thesis will be defined with respect to promise problems (and not languages). Remember that a (promise) problem $A = (A_{yes}, A_{no})$ consists of two non-intersecting sets $A_{yes}, A_{no} \subseteq \{0,1\}^*$ (the YES and NO instances, respectively). We have that $A_{inv} = \{0,1\}^* \setminus A_{yes} \cup A_{no}$ is the set of all invalid instances, and we do not care how a verifier behaves on problem instances $x \in A_{inv}$ (it can accept or reject

arbitrarily, see the paragraph 'oracle access' for a more elaborate discussion on what this entails). All previously defined classes, such as NP, can be understood in this chapter as a promise class.

## 7.5.1   Semi-classical states

Before we formally introduce the guided local Hamiltonian problem, we need to define what we mean by a guiding state. To do so, we borrow the definition given by Gharibian and Le Gall [GL22]. They require a guiding state to have two properties: First, a guiding state must have an efficient classical description (which is given as an input to the problem). Second, they require that it is efficient to classically sample from the distribution described by the state, given its description[5]. States with these two properties are called "semi-classical" states. In Chapter 8 we will dive deeper into what types of "semi-classical" states are actually interesting in this setting. For this part of the thesis, the following definitions are sufficient. One type of semi-classical state we use in this chapter is a polynomial-size variant of the notion of subset states, first introduced in [GKS15].

**7.5.1.** DEFINITION (Semi-classical subset state). We say that a normalized state $|u\rangle \in \mathbb{C}^{2^n}$ is a semi-classical subset state if there is a subset $S \subseteq \{0,1\}^n$ with $|S| = poly(n)$ such that

$$|u\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle \,.$$

A semi-classical subset state can be efficiently described by the description of $S$. It is clear that we can efficiently sample from the probability distribution that outputs $x \in \{0,1\}^n$ with probability $|\langle x|u\rangle|^2$, i.e., according to the uniform distribution over $S$.

We next introduce a generalized version of a semi-classical subset state.

**7.5.2.** DEFINITION (Semi-classical encoded state). We say that a normalized state $|u\rangle \in \mathbb{C}^{2^m}$, where $n < m$ and $m \in \mathcal{O}(n)$, is a semi-classical encoded state if there is a subset $S \subseteq \{0,1\}^n$ with $|S| = poly(n)$ and a set of isometries $V_1, V_2, \ldots, V_n$, where each $V_i$ maps a 1-qubit state to an $\mathcal{O}(1)$-qubit state, such that

$$|u\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} V_1(|x_1\rangle) \otimes V_2(|x_2\rangle) \otimes \cdots \otimes V_n(|x_n\rangle).$$

A semi-classical encoded state is indeed a semi-classical subset state if the encoding is trivial (i.e. $V_1 = V_2 = \cdots = V_n = I$). A semi-classical encoded state can be described by $S$ and isometries $V_1, V_2 \ldots, V_n$. We can also efficiently sample from the semi-classical encoded state, as we show in the following lemma.

---

[5]The requirement of sampling access for a guiding state is motivated by the existence of an efficient classical algorithm for the GLH problem with constant precision, given a guiding state with sampling access, as shown in [GL22].

**7.5.3.** LEMMA. *Given the description of an m-qubit semi-classical encoded state $|u\rangle$, we can classically efficiently sample from the probability distribution that outputs $x \in \{0,1\}^m$ with probability $|\langle x|u\rangle|^2$.*

**Proof:**
Assume we are given the description, $S \subseteq \{0,1\}^n$ and $V_1, V_2, ..., V_n$, of the semi-classical encoded state

$$|u\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} V_1(|x_1\rangle) \otimes V_2(|x_2\rangle) \otimes \cdots \otimes V_n(|x_n\rangle).$$

Let $P(y_0, y_1, ..., y_{i-1}) = |(\langle y_0, y_1, ..., y_{i-1}| \otimes I) |u\rangle|^2$ be the probability that the measurement outcome of the first $i$ qubits of $|u\rangle$ in the computational basis is $y_0, y_1, ..., y_{i-1}$. For each $i \in [m]$, we can efficiently calculate $P(y_0, y_1, ..., y_{i-1})$ because $|S| = poly(n)$ and $V_1(|x_1\rangle) \otimes V_2(|x_2\rangle) \otimes \cdots \otimes V_n(|x_n\rangle)$ is a product state of $\mathcal{O}(1)$-qubit states. Then, we can also efficiently calculate the conditional probability

$$P(z|y_0, y_1, ..., y_{i-1}) = \frac{P(y_0, y_1, ..., y_{i-1}, z)}{P(y_0, y_1, ..., y_{i-1})}.$$

If the bits $y_0, y_1, ..., y_{i-1}$ have already been sampled, we compute $P(z|y_0, y_1, ..., y_{i-1})$ and sample the next bit by tossing a coin with bias $P(0|y_0, y_1, ..., y_{i-1})$. In this way, we can classically efficiently sample from the probability distribution that outputs $x$ with probability $|\langle x|u\rangle|^2$.

$\square$

## 7.5.2   Formal definition of $\mathsf{GLH}(k, \delta, \zeta)$

Given the definitions of semi-classical states, we can formally state the guided local Hamiltonian problem[6]:

**7.5.4.** DEFINITION (Guided Local Hamiltonian problem). The $k$-local guided Hamiltonian problem $\mathsf{GLH}(k, \delta, \zeta)$ is:
**Input:** A $k$-local Hamiltonian $H$ with $\|H\| \leq 1$ acting on $n$ qubits, the description of a semi-classical encoded state $|u\rangle \in \mathbb{C}^{2^n}$, threshold parameters $a, b \in \mathbb{R}$ such that $b - a \geq \delta > 0$, and a bound on the fidelity $\zeta$.
**Promises:** $\|\Pi_0 |u\rangle\|^2 \geq \zeta$, where $\Pi_0$ denotes the projection on the ground space of $H$; furthermore, either $\lambda_0(H) \leq a$ or $\lambda_0(H) \geq b$ holds.
**Output:**

---

[6]This definition of GLH is very similar to the definition of $\mathrm{GLH}^*(k, a, b, \delta)$ in [GL22]. The difference is that while the guiding states used in [GL22] are restricted to semi-classical subset states (Definition 7.5.1), in our definition we use the more general concept of semi-classical encoded states (Definition 7.5.2).

- If $\lambda_0(H) \leq a$, output YES.

- If $\lambda_0(H) \geq b$, output NO.

We next define the guided local Hamiltonian low energy (GLHLE) problem, which can be viewed as a generalization of GLH by considering arbitrary eigenstates of Hamiltonians. For an $n$-qubit Hamiltonian $H$, we denote $\Pi_c$ the projector onto the space spanned by the states of $H$ that have energy $\lambda_c(H)$.

**7.5.5.** DEFINITION (Guided Local Hamiltonian Low Energy). The $k$-local guided Hamiltonian low energy problem $\mathsf{GLHLE}(k, c, \delta, \zeta)$ is:
**Input:** A $k$-local Hamiltonian $H$ on $n$ qubits such that $\|H\| \leq 1$, the description of a semi-classical encoded state $|u\rangle \in \mathbb{C}^{2^n}$, threshold parameters $a, b \in \mathbb{R}$ such that $b - a \geq \delta > 0$, a bound on the fidelity $\zeta$, and a constant $c \in \mathbb{N}_{\geq 0}$.
**Promise:** $\|\Pi_c |u\rangle\|^2 \geq \zeta$, where $\Pi_c$ denotes the projection on the subspace spanned by the $c$th eigenstate(s), ordered by non-decreasing energy, of $H$, and either $\lambda_c(H) \leq a$ or $\lambda_c(H) \geq b$ holds.
**Output:**

- If $\lambda_c(H) \leq a$, output YES.

- If $\lambda_c(H) \geq b$, output NO.

**7.5.6.** REMARK. When $c = 0$, $\mathsf{GLHLE}(k, 0, \delta, \zeta)$ is the same problem as $\mathsf{GLH}(k, \delta, \zeta)$.

Our BQP-hardness proof for Theorem 7.4.1 will be based on the construction by Gharibian and Le Gall [GL22], therefore, we first discuss their construction and then show how we improve on it.

## 7.6   Gharibian and Le Gall's construction

In this section we will briefly restate Gharibian and Le Gall's original construction [GL22]. They start their reduction from circuit evaluation.

**BQP-hard circuit:** Let $\Pi = (\Pi_{\text{yes}}, \Pi_{\text{no}})$ be a promise problem in BQP, and $x \in \{0, 1\}^n$ an input. Let $U = U_T \ldots U_1$ be a poly-time uniformly generated quantum circuit consisting of 1- and 2-qubit gates $U_i$, deciding on $\Pi$. More precisely, $U$ takes an $n$-qubit input register $A$, and a $q(n) = poly(n)$-qubit work register $B$ and outputs, upon measurement, a 1 on the first qubit with probability at least $\alpha$ (resp. at most $\beta$) if $x \in \Pi_{\text{yes}}$ (resp. $x \in \Pi_{\text{no}}$). We can assume w.l.o.g. $\alpha = 1 - 2^{-n}$, $\beta = 2^{-n}$ by the fact that BQP allows for error-reduction.

Then they use the well-known Kitaev clock construction to map the BQP-circuit to a Hamiltonian.

**Circuit to Hamiltonian mapping:** Consider Kitaev's original 5-local clock Hamiltonian [KSV02]:

$$H_{KF}^x = \Delta(H_{\mathrm{in}} + H_{\mathrm{clock}} + H_{\mathrm{prop}}) + H_{\mathrm{out}}, \tag{7.1}$$

constructed from the BQP verifier circuit $U$, where $\Delta$ is to be chosen later, with the separate terms being:

$$
\begin{aligned}
H_{\mathrm{in}} &:= (I - |x\rangle \langle x|)_A \otimes (I - |0\ldots 0\rangle \langle 0\ldots 0|)_B \otimes |0\rangle \langle 0|_C, \\
H_{\mathrm{out}} &:= |0\rangle \langle 0|_{\mathrm{out}} \otimes |T\rangle \langle T|_C, \\
H_{\mathrm{clock}} &:= \sum_{j=1}^{T} |0\rangle \langle 0|_{C_j} \otimes |1\rangle \langle 1|_{C_{j+1}}, \\
H_{\mathrm{prop}} &:= \sum_{t=1}^{T} H_t, \tag{7.2}
\end{aligned}
$$

where,

$$
\begin{aligned}
H_t :=& -\frac{1}{2} U_t \otimes |t\rangle \langle t-1|_C - \frac{1}{2} U_t^\dagger \otimes |t-1\rangle \langle t|_C \\
&+ \frac{1}{2} I \otimes |t\rangle \langle t|_C + \frac{1}{2} I \otimes |t-1\rangle \langle t-1|_C. \tag{7.3}
\end{aligned}
$$

Here $A$ is the same input register as in the BQP circuit, similarly $B$ is the work register and $C$ denotes the 'clock' register consisting of $T = poly(n)$ qubits. This Hamiltonian has several useful properties based on $U$:

- If $U$ accepts with at least probability $\alpha$, then $\lambda_0(H^{(5)}) \leq \frac{1-\alpha}{T}$.

- If $U$ accepts with at most probability $\beta$, then $\lambda_0(H^{(5)}) \geq \Omega(\frac{1-\beta}{T^3})$.

Furthermore, we can split $H_{KF}^x$ into two separate terms:

$$
\begin{aligned}
H_1 &:= H_{\mathrm{in}} + H_{\mathrm{clock}} + H_{\mathrm{prop}} \\
H_2 &:= H_{\mathrm{out}}
\end{aligned}
$$

where $H_1$ has the following property:

**7.6.1.** LEMMA (Lemma 2.2 of [GY19] (based on Lemma 3 of [GK12])). *The smallest non-zero eigenvalue of $H_1$ is at least $\pi^2/(64m^3)$ for $m \geq 1$.*

Furthermore, the null space of $H_1$ (zero-energy space) is spanned by what is known as history states:

$$|\psi_{\mathrm{hist}}\rangle = \frac{1}{\sqrt{T}} \sum_{t=1}^{T} U_t \ldots U_1 |x\rangle_A |0\ldots 0\rangle_B |t\rangle_C,. \tag{7.4}$$

In the special case of reducing from a BQP-verifier circuit there is only one such history state, because there is no witness as input for the BQP-verifier circuit. The first trick that the authors use is that changing the prefactor $\Delta$ does not increase the energy of the history state, but it does for any state that is not a history state. This allows them to increase the overlap between the history state and the ground state of the Hamiltonian (for a more precise statement see Proposition 7.7.1).

Then they construct poly-sized subset state that has significant overlap with the history state.

**Guiding state:**   Consider the following semi-classical guiding state in [GL22]

$$|u\rangle = \frac{1}{\sqrt{T}} \sum_{t=1}^{N} |x\rangle_A |0 \dots 0\rangle_B |t\rangle_C. \tag{7.5}$$

In general, this guiding state has at most $\mathcal{O}(1/(TN))$ fidelity with the history state and therefore an even smaller fidelity with the actual ground state of $H_{FK}^x$. In order to meet the promise that $\|\Pi_{H_{FK}^x} |u\rangle\|^2 \geq \zeta$ in both the YES- and NO-case, Gharibian and Le Gall use the following two tricks:

**Pre-idling trick:**   To control the YES case, they use what is known as the pre-idling trick. By *pre-idling* the circuit $U$ that is in the clock Hamiltonian – i.e. applying $N$ identity gates before the first actual gate – the fidelity between $|u\rangle$ and the history state can be increased. This increases the number of gates from $T$ to $M = T + N$. Denote the weighted and pre-idled Hamiltonian by $\hat{H}_{KF}^x$. This also slightly changes the energy bounds on $\hat{H}_{KF}^x$, redefining::

$$\hat{\alpha} = \frac{1 - \alpha}{M}$$
$$\hat{\beta} = \frac{1 - \beta}{M^3}$$

(more details in Proposition 7.7.1).

**Block-encoding the Hamiltonian:**   Finally, by block-encoding $\hat{H}_{FK}^x$ into a larger Hamiltonian $H_{FK}'^x$, which acts on $n + r + M + 1$ qubits (adding another single-qubit register $D$), one can add another Hamiltonian (in their case a scaled identity term) in another block such that the ground space in the NO-case is trivial, only increasing the locality in the construction by 1. By setting this specific qubit in the guiding state to the $|+\rangle$ state, one ensures that it has fidelity with both the NO- and YES-cases.

The final Hamiltonian is then

$$H'^x_{FK} := \frac{\hat{\alpha} + \hat{\beta}}{2} I_{ABC} \otimes |0\rangle \langle 0| \, 0_D + \hat{H}^x_{FK} \otimes |1\rangle \langle 1| \, 1_D, \qquad (7.6)$$

where $\hat{H}^x_{FK} = \Delta \left( H_{\text{in}} + H_{\text{clock}} + H_{\text{prop}} \right) + H_{\text{out}}$. The guiding state becomes

$$|u\rangle := |x\rangle_A \, |0 \ldots 0\rangle_B \left( \frac{1}{\sqrt{M}} \sum_{t=1}^M |t\rangle \right)_C |+\rangle_D. \qquad (7.7)$$

Since the overall construction starts from a 5-local Hamiltonian, the last trick increases the locality to 6 and restricts the fidelity to be at most $1/2 - \Omega(1/\text{poly}(n))$.

## 7.7 Our construction for BQP-hardness

In this section, we discuss how we achieve our three improvements in the BQP-hardness construction of [GL22]. We will build on the construction described above and prove our main theorem by proving three separate propositions.

### 7.7.1 Improved fidelity

Our first result is the improvement of the fidelity $\zeta$. The construction of [GL22] cannot exceed $\zeta = 1/2$, but our construction achieves fidelity $\zeta = 1 - 1/\text{poly}(n)$. First, let us explain why the construction of [GL22] cannot exceed fidelity $\zeta = 1/2$. In their construction, their last trick is to split the Hilbert space to deal with the YES and NO-case separately. It is clear that a ground state of $H'^x_{FK}$ is $|\psi\rangle \otimes |1\rangle$ in the YES-case, where $|\psi\rangle$ is a ground state of $H'$. For the NO case, a ground state is $|\phi\rangle \otimes |0\rangle$, where $|\phi\rangle$ can be any state. Therefore, it can then be easily observed that the optimal guiding state (i.e. the guiding state that has the maximum fidelity with ground states *in both the* YES *and the* NO*-cases*) is written as $|\phi\rangle \otimes |+\rangle$ for a certain choice of $|\phi\rangle$, which shows that the fidelity cannot exceed $1/2$ in this construction.

To overcome the problem, we realize that both in the YES and the NO-case the history state is the ground state of $H_1$, the only term giving energy to the history state is $H_2$. The main idea is to use a large energy penalty term to rule out all low-energy states which do not look like "history states", both in the YES and NO-case. We then show that the corresponding guiding state can be chosen as the semi-classical subset state introduced in [GL22] Equation 7.5. To obtain this, we notice that the ground state of our Hamiltonian is gapped and unique. This is because we are doing a reduction from BQP (as opposed to QMA). In other words, there is no QMA "proof" to be plugged into the history state construction, and therefore there is a unique low-energy history state. In sum, this allows us to remove the block encoding approach of [GL22]. Additionally, this allows us to

show that there is a gap between the ground state and first excited state of the Hamiltonian. This gives us our first proposition.

**7.7.1.** PROPOSITION. *For any $\zeta \in (0, 1 - \Omega(1/\text{poly}(n)))$, there exist $a, b \in [0, 1]$ such that $b - a = \delta$, with $\delta = 1/\text{poly}(n)$ such that the problem $\mathsf{GLH}(5, \delta, \zeta)$ is $\mathsf{BQP}$-hard. Moreover, it is still $\mathsf{BQP}$-hard with the additional two promises that*

1. *$H$ has a non-degenerate ground state separated from the first excited state by a spectral gap $\gamma \in \Omega(1/\text{poly}(n))$, with $\gamma \geq \delta$, in both cases $\lambda_0(H) \leq a$ and $\lambda_0(H) \geq b$. (We call such instances $\gamma$-gapped $\mathsf{GLH}(k, \delta, \zeta)$.)*

2. *The guiding state is restricted to be a semi-classical subset state.*

**Proof:**
We start by using the same $\mathsf{BQP}$-hard verifier circuit as in Paragraph 7.6, which we restate here for completeness. Let $\Pi = (\Pi_{\text{yes}}, \Pi_{\text{no}})$ be a promise problem in $\mathsf{BQP}$, and $x \in \{0, 1\}^n$ an input. Let $U = U_T \ldots U_1$ be a poly-time uniformly generated quantum circuit consisting of 1- and 2-qubit gates $U_i$, deciding on $\Pi$. More precisely, $U$ takes an $n$-qubit input register $A$, and a $q(n) = \text{poly}(n)$-qubit work register $B$ and outputs, upon measurement, a 1 on the first qubit with probability at least $\alpha$ (resp. at most $\beta$) if $x \in \Pi_{\text{yes}}$ (resp. $x \in \Pi_{\text{no}}$). We can assume w.l.o.g. $\alpha = 1 - 2^{-n}$, $\beta = 2^{-n}$ by the fact that $\mathsf{BQP}$ allows for error-reduction.

We will also need the following Lemma, known as the projection lemma:

**7.7.2.** LEMMA (Kempe, Kitaev, Regev [KKR06]). *Let $H = H_1 + H_2$ be the sum of two Hamiltonians operating on some Hilbert space $\mathcal{H} = \mathcal{S} + \mathcal{S}^\perp$. The Hamiltonian $H_1$ is such that $\mathcal{S}$ is a zero eigenspace and the eigenvectors in $\mathcal{S}^\perp$ have eigenvalue at least $J > 2\|H_2\|_\infty$. Then,*

$$\lambda(H_2|_{\mathcal{S}}) - \frac{\|H_2\|_\infty^2}{J - 2\|H_2\|_\infty} \leq \lambda(H) \leq \lambda(H_2|_{\mathcal{S}}),$$

*where recall $\lambda(H_2|_{\mathcal{S}})$ denotes the smallest eigenvalue of $H_2$ restricted to space $\mathcal{S}$.*

Our construction is now as follows. First, pre-idle the verifier by updating $U$ to an $M := (T + N)$-gate circuit ($N$ to be chosen shortly as needed), where the first $N$ gates are $I$, and the last $T$ gates are the original circuit $U$. Consider the 5-local clock Hamiltonian of Equation 7.1:

$$H = \Delta(H_{\text{in}} + H_{\text{clock}} + H_{\text{prop}}) + H_{\text{out}}, \tag{7.8}$$

where $H_{\text{prop}}$ is now using our new $M$-step $U$, and $\Delta \geq 0$ is some constant to be set as needed. As defined, $H$ has $\text{poly}(n)$ norm. The corresponding low-energy history state is

$$|\psi_{\text{hist}}\rangle = \frac{1}{\sqrt{M + 1}} \sum_{t=0}^{M} U_t \cdots U_1 |x\rangle_A |0 \cdots 0\rangle_B |t\rangle_C, \tag{7.9}$$

so that

$$\langle \psi_{\text{hist}} | H_{\text{out}} | \psi_{\text{hist}} \rangle = \frac{1 - p}{M + 1}, \tag{7.10}$$

for $p$ the acceptance probability of the verifier $U$.

Finally, set the semi-classical subset state as:

$$|u\rangle = |x\rangle_A |0 \cdots 0\rangle_B \left( \frac{1}{\sqrt{N}} \sum_{t=1}^{N} |t\rangle \right). \tag{7.11}$$

We will first show that, by setting $\Delta$ as needed, the history state has significant overlap with the ground state, in both the YES and NO-case. Then we will show that, by setting $N$ as needed, the history state and $|u\rangle$ have large overlap, concluding that $|u\rangle$ and the ground state have overlap $1 - 1/poly(n)$.

**Fidelity between $|\psi_{\text{hist}}\rangle$ and the ground space of $H$.** Let $\mathcal{N}$ denote the null space $\text{Null}(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}})$, where $\mathcal{N}$ is 1-dimensional (since we are embedding a BQP computation) and spanned by $|\psi_{\text{hist}}\rangle$. Let $|E_0\rangle$ be an arbitrary ground state of $H$. Write $|E_0\rangle = \eta_1 |\psi_{\text{hist}}\rangle + \eta_2 \left| \psi_{\text{hist}}^{\perp} \right\rangle$ for $|\psi_{\text{hist}}\rangle \in \mathcal{N}$, $\left| \psi_{\text{hist}}^{\perp} \right\rangle \in \mathcal{N}^{\perp}$, and $|\eta_1|^2 + |\eta_2|^2 = 1$. Then, since $H' \succeq 0$ (since $H_{\text{in}}, H_{\text{prop}}, H_{\text{out}}, H_{\text{clock}} \succeq 0$ and $\Delta \geq 0$), and where $\lambda(H)$ is the smallest eigenvalue of $H$,

$$
\begin{align}
\lambda(H) &= \langle E_0 | H | E_0 \rangle \tag{7.12} \\
&\geq \langle E_0 | \Delta(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}}) | E_0 \rangle \\
&= |\eta_1|^2 \langle \psi_{\text{hist}} | \Delta(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}}) | \psi_{\text{hist}} \rangle \tag{7.13} \\
&\quad + |\eta_2|^2 \left\langle \psi_{\text{hist}}^{\perp} \right| \Delta(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}}) \left| \psi_{\text{hist}}^{\perp} \right\rangle \\
&= |\eta_2|^2 \left\langle \psi_{\text{hist}}^{\perp} \right| \Delta(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}}) \left| \psi_{\text{hist}}^{\perp} \right\rangle \\
&\geq |\eta_2|^2 \frac{\pi^2 \Delta}{64 M^3}, \tag{7.14}
\end{align}
$$

where the second statement holds since $H_{\text{out}} \succeq 0$, the third and fourth since $(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}}) | \psi_{\text{hist}} \rangle = 0$, and the last by Lemma 7.6.1. We conclude that

$$
\begin{align}
|\langle E_0 | \psi_{\text{hist}} \rangle|^2 &\geq 1 - \frac{64 M^3 \lambda(H)}{\pi^2 \Delta} \\
&\geq 1 - \frac{64 M^3}{\pi^2 \Delta} \langle \psi_{\text{hist}} | H_{\text{out}} | \psi_{\text{hist}} \rangle \\
&= 1 - \frac{64 M^3 (1 - p)}{\pi^2 \Delta (M + 1)} \\
&\geq 1 - \frac{32 M^2 (1 - p)}{\pi^2 \Delta}, \tag{7.15}
\end{align}
$$

where the second statement holds by the fact that $\langle E_0 | H | E_0 \rangle \leq \langle \psi_{\text{hist}} | H | \psi_{\text{hist}} \rangle = \langle \psi_{\text{hist}} | H_{\text{out}} | \psi_{\text{hist}} \rangle$. The third statement follows by Equation (7.10).

**Fidelity $|u\rangle$ with the ground state.** The fidelity between $|u\rangle$ and the history state is

$$|\langle u|\psi_{\text{hist}}\rangle|^2 = \left|\sum_{t=1}^{N} \frac{1}{\sqrt{N(M+1)}}\right|^2 = \frac{N}{M+1} = \frac{N}{T+N+1}.$$

Set $N = 2T^{k+1}$, for $k \in \Theta(1)$ to be chosen shortly. Then,

$$|\langle u|\psi_{\text{hist}}\rangle|^2 = \frac{2T^{k+1}}{T + 2T^{k+1} + 1} \geq \frac{1}{1 + \frac{1}{T^k}} \geq 1 - \frac{1}{T^k}, \tag{7.16}$$

where the last bound follows from the Taylor series for $1/(1-x)$, which converges for any $|x| < 1$.

The triangle inequality combined with Equation (7.15) and Equation (7.16) now yields

$$|\langle E_0|u\rangle|^2 \geq 1 - \left(\frac{32M^2(1-p)}{\pi^2\Delta} + \frac{1}{T^k}\right)^2 \geq 1 - \left(\frac{32(T + 2T^{k+1})^2}{\pi^2\Delta} + \frac{1}{T^k}\right)^2, \tag{7.17}$$

where we have used identity $\||v\rangle\langle v| - |u\rangle\langle u|\|_{\text{tr}} = 2\sqrt{1 - |\langle u|v\rangle|^2}$, and that $p \leq 1$. We set $p = 0$, which covers both the YES and NO cases. By choosing $k \in \Theta(1)$ and $\Delta \in poly(n)$ appropriately, we can now guarantee the desired claim that $|\langle E_0|u\rangle|^2 \geq 1 - 1/r(n)$ for large enough $n$, for polynomial $r$ from the statement of the proposition.

We still need to make sure that the spectral gap scales as $\delta = 1/poly(n)$

**Correctness for YES and NO cases.** In the YES case, i.e. when $x \in \Pi_{\text{yes}}$, we have by Equation (7.10) that

$$\lambda_{\min}(H) \leq \langle\psi_{\text{hist}}| H |\psi_{\text{hist}}\rangle = \langle\psi_{\text{hist}}| H_{\text{out}} |\psi_{\text{hist}}\rangle = \frac{1-p}{M+1} \leq \frac{1}{2^n(M+1)} = a. \tag{7.18}$$

For the NO case, i.e. when $x \in \Pi_{\text{no}}$, a bit of care is needed, as the naive approach of applying Hölder's inequality to $\text{tr}(H(|E_0\rangle\langle E_0| - |\psi_{\text{hist}}\rangle\langle\psi_{\text{hist}}|))$, together with Equation 7.15, is troublesome, as both $\|H\|_\infty$ and $|\langle E_0|\psi_{\text{hist}}\rangle|^2$ increase monotonically with $\Delta$. Thus, we instead apply Lemma 7.7.2, which, for $H_1 = \Delta(H_{\text{in}} + H_{\text{prop}} + H_{\text{clock}})$ and $H_2 = H_{\text{out}}$, says,

$$\begin{aligned}
\lambda(H) &\geq \lambda(H_2|_{\mathcal{S}}) - \frac{\|H_2\|_\infty^2}{J - 2\|H_2\|_\infty} \\
&\geq \langle\psi_{\text{hist}}| H_{\text{out}} |\psi_{\text{hist}}\rangle - \frac{1}{\frac{\Delta\pi^2}{64M^3} - 2} \\
&\geq \frac{1-p}{M+1} - \frac{128M^3}{\Delta\pi^2} \\
&\geq \frac{1 - \frac{1}{2^n}}{2M+1} - \frac{128M^3}{\Delta\pi^2} = b.
\end{aligned} \tag{7.19}$$

This gives decision gap:

$$\delta = b - a = \frac{1 - \frac{1}{2^n}}{M+1} - \frac{128M^3}{\Delta\pi^2} - \frac{\frac{1}{2^n}}{(M+1)} = \frac{1 - \frac{1}{2^{n-1}}}{M+1} - \frac{128M^3}{\Delta\pi^2} \geq \frac{1}{2(M+1)} - \frac{128M^3}{\Delta\pi^2}$$

For large enough (but fixed polynomial) $\Delta = \Omega(M^3)$, we thus have that the decision gap $\delta = \mathcal{O}(1/M)$ is at least an inverse polynomial, as required.

**Spectral gap of** $H$  Let $|E_1\rangle$ be the first excited state of $H$. Then we can write $|E_1\rangle$ as $\xi_1 |\psi_{\text{hist}}\rangle + \xi_2 \left|\psi_{\text{hist}}^\perp\right\rangle$, where $|\psi_{\text{hist}}\rangle \in \mathcal{N}$ and $\left|\psi_{\text{hist}}^\perp\right\rangle \in \mathcal{N}^\perp$. It follows that:

$$|\xi_2|^2 = 1 - |\langle\psi_{\text{hist}}|E_1\rangle|^2 = 1 - |(\eta_1 \langle E_0| + \eta_2 \langle E_0^\perp|) |E_1\rangle|2 \geq 1 - 1/poly(n) \geq \frac{1}{2},$$

where we used $\langle E_0|E_1\rangle = 0$ and Equation 7.15. This gives the following lower bound on the first excited-state energy:

$$\langle E_1| H |E_1\rangle \geq |\xi_2|^2 \frac{\pi^2\Delta}{64M^3} \geq \frac{\pi^2\Delta}{128M^3},$$

following the exact same derivation as for Equation 7.14. We can use the energy of the history state to bound the energy of the ground state from above, for this we use Equation 7.10 and set $p = 0$ (accounting for both the YES and NO-case). This gives

$$\gamma = \frac{\pi^2\Delta}{128M^3} - \frac{1}{M+1},$$

which for sufficiently (but at most polynomially) large $\Delta$ scales inverse polynomially with $n$, and it holds that $\gamma \geq \delta$. What is left is to rescale $||H||$ by a sufficiently large polynomial such that $||H|| \leq 1$, completing the proof. $\square$

## 7.7.2  Extension to excited states

Our second result is to extend the GLH$(k, \delta, \zeta)$ problem to the question of excited state energy estimation, we call this the *Guided k-Local Hamiltonian Low Energy* (GLHLE$(k, c, \delta, \zeta)$) problem. In Ref. [JGL10], the authors show that determining the $c$th excited state energy of a $k$-local Hamiltonian ($k \geq 3$), where $c = poly(n)$, is QMA-complete – even if all the $c - 1$ energy eigenstates and corresponding energies are known. In their construction, they embed a $k$-local Hamiltonian $H$, encoding the QMA computation, in a Hamiltonian $H'$ living on a larger Hilbert space. This allows them to add up to a polynomial number of artificial eigenstates to $H'$ below the ground state of $H$. Finding the $c$th eigenvalue of $H'$ is then just as hard as finding the ground state of $H$. We show that this construction translates to the setting with guiding states. As a bonus, we will later show that

the unguided problem of estimating eigenstate energies is QMA-hard for $k = 2$, which was left open in [JGL10].

The next proposition extends the result to excited states, at the cost of increasing the locality of the construction by one.

**7.7.3.** PROPOSITION. *For any $\zeta \in (0, 1 - \Omega(1/poly(n)))$ there exist $a, b \in [-1, 1]$ with $b - a = \delta \in \mathcal{O}(1/poly(n))$ and some number $0 \leq c \leq poly(n)$ such that* GLHLE$(6, c, \delta, \zeta)$ *is* BQP-*hard even when,*

1. *For all $0 \leq i \leq c$, $\lambda_i(H)$ is non-degenerate and is separated by a gap $\gamma \in \Omega(1/poly(n))$ from both $\lambda_{i-1}(H)$ and $\lambda_{i+1}(H)$ with $\gamma \geq \delta$. (We call such instances $\gamma$-gapped* GLHLE$(k, c, \delta, \zeta)$*)*

2. *The guiding state is restricted to be a semi-classical subset state.*

**Proof:**
We will reduce directly from the BQP-complete Hamiltonian $H$ as defined in Eq. 7.8. Again, let $|u\rangle$ be the semi-classical guiding state as in Equation 7.11 such that $|\langle u|E_0\rangle| \geq \zeta$. Consider the following 6-local Hamiltonian $H^{(c)}$ on $m+1$ qubits[7]:

$$H^{(c)} = H^{(z)} \otimes |0\rangle \langle 0| + H^{(s)} \otimes |1\rangle \langle 1|, \tag{7.20}$$

where

$$H^{(z)} = \sum_{i=0}^{d} 2^i |1\rangle \langle 1|_i + \sum_{i=d+1}^{m} 2^{d+1} |1\rangle \langle 1|_i - \left(c - \frac{1}{2}\right) I,$$

$$H^{(s)} = \frac{1}{2} \frac{H + I/4}{\|H\| + 1/4} - \frac{1}{4}I,$$

where we have that $d = \lceil \log_2(c) \rceil$. $H^{(z)}$ has exactly $c$ states with negative energy, with the smallest eigenvalue being $-c + \frac{1}{2}$ and the largest eigenvalue at $\sum_{i=0}^{d} 2^i + \sum_{i=d+1}^{m} 2^{d+1} - \left(c - \frac{1}{2}\right) = 2^{d+1} + 2^{d+1}(m-d) - \frac{1}{2} - c$. The spectrum jumps in integer steps of 1, and has as largest negative (resp. smallest non-negative) energy value $-\frac{1}{2}$ (resp. $\frac{1}{2}$). Since eig$(H^{(s)}) \in [-1/4, 1/4]$, we must have that $H^{(s)}$ sits precisely at the $c$th excited state level (or $c+1$th eigenstate level) in $H^{(c)}$. Therefore, given a guiding state $|u\rangle$ for $H$ such that $|\langle u|E_0\rangle| \geq \zeta$, one has that the guiding state $|u^{(c)}\rangle = |u\rangle \otimes |1\rangle$ is also semi-classical and must have $|\langle u^{(c)}|E_c^{(c)}\rangle| \geq \zeta$, where $|\psi_c^{(c)}\rangle$ denotes the $c$th excited state of $H^{(c)}$. Since this construction of $H^{(c)}$ and $|u^{(c)}\rangle$ provides a polynomial time reduction from an instance of GLH$(k, \delta, \zeta)$ to one of GLHLE$(k, c, \delta, \zeta)$, whenever $c = \mathcal{O}(poly(n))$, we must have that GLHLE$(k, c, \delta, \zeta)$ is BQP-hard whenever $k \geq 6$.

---

[7]Note that this gadget can be trivially changed such that estimating the $n$ highest energy states is BQP-hard.

The gap between $\lambda_i(H^{(c)}) - \lambda_{i-1}(H^{(c)}) = 1$ and $\lambda_{i+1}(H^{(c)}) - \lambda_i(H^{(c)}) = 1$ for all $i < c-1$ by construction. The gap between $\lambda_c(H^{(c)}) - \lambda_{c-1}(H^{(c)}) = \frac{1}{4} + \lambda(H)$, and the gap between $\lambda_{c+1}(H^{(c)}) - \lambda_c(H^{(c)}) = \gamma$ as before. The norm of the new Hamiltonian is bounded by $|H^{(c)}| = \mathcal{O}(\mathrm{poly}(n))$, hence after normalization we retain $\lambda_c(H^{(c)}) - \lambda_{c-1}(H^{(c)}) \geq \lambda_{c+1}(H^{(c)}) - \lambda_c(H^{(c)}) = \Omega(1/\mathrm{poly}(n))$. $\qquad\square$

## 7.7.3 Reducing the locality

Our third result is BQP-hardness of GLHLE$(k, c, \delta, \zeta)$ for $k = 2$. We will do the reduction by making use of *perturbative gadgets*. Perturbative gadgets are standard techniques from the Hamiltonian complexity toolbox and are used to transform one Hamiltonian into another whilst approximately preserving the (low-energy) spectrum. We will use such gadgets here, and will be particularly interested in those that preserve not only the low-energy spectrum of the original Hamiltonian, but also the structure of the low-energy eigenstates. In [CMP18], the authors introduce the following definition of simulation, and demonstrate via the use of perturbative gadgets that there are families of Hamiltonians which can be 'reduced' to different families of Hamiltonians with simpler/lower locality interactions. This was later more extensively studied by [ZA21]. Note that these results originally only applied to qubits, but can be extended to qudits [PM21].

**7.7.4.** DEFINITION (Approximate Hamiltonian simulation [CMP18; ZA21]).
We say that an $m$-qubit Hamiltonian $H'$ is a $(\Delta, \eta, \epsilon)$-simulation of an $n$-qubit Hamiltonian $H$ if there exists a local encoding $\mathcal{E}(H) = V(H \otimes P + \bar{H} \otimes Q)V^\dagger$ such that

1. There exists an encoding $\tilde{\mathcal{E}}(H) = \tilde{V}(H \otimes P + \bar{H} \otimes Q)\tilde{V}^\dagger$ such that $\tilde{\mathcal{E}}(\mathbb{1}) = P_{\leq \Delta(H')}$ and $\|V - \tilde{V}\| \leq \eta$, where $P_{\leq \Delta(H')}$ is the projector onto the subspace spanned by eigenvectors of $H'$ with eigenvalue below $\Delta$,

2. $\|H'_{\leq \Delta} - \tilde{\mathcal{E}}(H)\| \leq \epsilon$, where $H'_{\leq \Delta} := P_{\leq \Delta(H')}H'$.

Here, $V$ is a local isometry that can be written as $V = \otimes_i V_i$, where each $V_i$ is an isometry acting on at most 1 qubit, and $P$ and $Q$ are locally orthogonal projectors such that $P + Q = I$, and $\bar{M}$ is the complex conjugate of $M$. Moreover, we say that the simulation is efficient if $m$ and $\|H'\|$ are at most $\mathcal{O}\left(poly(n, \eta^{-1}, \epsilon^{-1}, \Delta)\right)$, and the description of $H'$ can be computed in $poly(n)$ time given the description of $H$.

In this framework, one approximately simulates the original Hamiltonian $H$ in the low-energy subspace of $H'$. Here the different parameters are to be understood as follows: $\Delta$ is the cutoff point to which energy level the simulating Hamiltonian is accurate, $\epsilon$ is the accuracy to which each individual energy is approximated,

and $\eta$ is a parameter that controls the precision to which states are approximated. There is a corresponding encoding of a state which can be taken to be

$$\mathcal{E}_{\text{state}}(\rho) = V(\rho \otimes \sigma)V^{\dagger} \tag{7.21}$$

for $\sigma$ such that $P\sigma = \sigma$ (if $P \neq 0$). If $\rho$ is the eigenvector of $H$ with eigenvalue $\alpha$, then $\mathcal{E}_{\text{state}}(\rho)$ is approximately the eigenvector of $H'$ with eigenvalue $\alpha' \in [\alpha - \epsilon, \alpha + \epsilon]$.

In [ZA21], it is shown that there exist families of Hamiltonians that can efficiently simulate any $\mathcal{O}(1)$-local Hamiltonian. They call such families of Hamiltonians *strongly universal Hamiltonians.*[8] We use the construction of strongly universal Hamiltonians of [ZA21] to show Proposition 7.7.8. Formally, the strong (and weak) universality is defined as follows:

**7.7.5.** DEFINITION (Strong and weak universality [ZA21]). A family of Hamiltonians $\mathcal{H} = \{H_m\}$ is weakly universal if given any $\Delta, \eta, \epsilon > 0$, any $\mathcal{O}(1)$-local, $n$-qubit Hamiltonian can be $(\Delta, \eta, \epsilon)$-simulated. Such a family is strongly universal if the simulation is always efficient.

The following result is shown in [ZA21]:

**7.7.6.** THEOREM ([ZA21]). *Any non-2SLD $\mathcal{S}$-Hamiltonian on a 2D-square lattice is strongly universal.*

This result is stronger than what we require in our proof; however, their proof consists of a list of reductions where they show as an intermediate result that:

**7.7.7.** LEMMA. *Any $\mathcal{O}(1)$-local Hamiltonian can be efficiently simulated by a spatially sparse 2-local Hamiltonian with no Y-terms.*

This was already proven in [CMP18] Lemma 22, when interested in weak universality and without the Hamiltonian being spatially sparse. Beyond this framework, we are additionally interested in the mapping of Equation 7.21, and whether this mapping preserves the semi-classical subset property of our guiding states. To verify this we give an overview of the construction by [ZA21] up to these spatially sparse 2-local Hamiltonians and show that the gadgets used for every step indeed retain the semi-classical subset property as required.

The next proposition brings the locality down from $k$ ($= 6$) to 2.

**7.7.8.** PROPOSITION. *Any $\gamma$-gapped $\mathsf{GLHLE}(k, c, \delta, \zeta)$ with $k \in \mathcal{O}(1)$, $\delta = 1/\operatorname{poly}(n)$, $\zeta \in (0, 1 - \Omega(1/\operatorname{poly}(n)))$, $0 \le c \le \operatorname{poly}(n)$, and $\gamma = \Omega(1/\operatorname{poly}(n))$,*

---

[8]It would be possible to show Theorem 7.4.1 by modifying the verifier circuit $\tilde{U}_x$ following [OT08] to make the constructed Hamiltonian spatially sparse. We believe Proposition 7.7.8 is interesting because the reduction holds for arbitrary $\mathcal{O}(1)$-local Hamiltonian even if it is not originally spatially sparse.

*s.t. $\gamma \geq \delta$, with a semi-classical subset state as a guiding state can be reduced to $\gamma'$-gapped* GLHLE$(2, c, \delta', \zeta')$ *with* $\delta' = 1/\operatorname{poly}(n)$, $\zeta' = (0, 1 - \Omega(1/\operatorname{poly}(n)))$ *and* $\gamma' = \Omega(\operatorname{poly}(n))$, *and with a semi-classical encoded subset state as guiding state in polynomial time.*

**Proof:**
Let $H$ and $|u\rangle$ be arbitrary inputs of $\gamma$-gapped GLHLE$(k, c, \delta, \zeta)$ with $k \in \mathcal{O}(1)$, $\delta = 1/\operatorname{poly}(n)$, $\gamma \geq \delta$, $\zeta = (0, 1 - \Omega(1/\operatorname{poly}(n)))$, and $c = \mathcal{O}(\operatorname{poly}(n))$.

From Lemma 7.7.7, we can efficiently find a 2-local Hamiltonian $H'$ that is a strong $(\Delta, \eta, \epsilon)$-simulation of $H$ given the description of $H$. We take $\epsilon < (b-a)/2$, $b' = b - \epsilon$, $a' = a + \epsilon$ and $\Delta = \mathcal{O}(\epsilon^{-1}\|H\|^2 + \eta^{-1}\|H\|)$. Because there is a gap between energy levels $\lambda_{i+1}(H) - \lambda_i(H) \geq \gamma \geq 2\epsilon$ for any $0 \leq i \leq c$, we have that none of them mix. Also this means that $\lambda_c(H') \leq a'$ if $\lambda_c(H) \leq a$, and $\lambda_c(H') \geq b'$ if $\lambda_c(H) \geq b'$ and $b' - a' \in \mathcal{O}(1/\operatorname{poly}(n))$ as needed.

The simulation framework gives us the *existence* of desirable eigenvectors in the simulated Hamiltonian. What remains is to show that (i) the encoded state of $|u\rangle$ still has $1 - 1/\operatorname{poly}(n)$ fidelity with $c$th excited state of $H'$ and (ii) the encoded state is still a semi-classical subset state after the simulation by a 2-local Hamiltonian.

**(i) Verification of the fidelity.** The fidelity can be analyzed by the following lemma:

**7.7.9.** LEMMA (Simulation of a gapped excited state). *Suppose the $c$th excited state $|E_c\rangle$ of $H$ is non-degenerate and separated from both the $c - 1$'th excited state and $c + 1$th excited state by a gap $\gamma$. Suppose $H'$ is a $(\Delta, \eta, \epsilon)$-simulation of $H$ such that $2\epsilon < \gamma$. Then $H'$ has a non-degenerate $c$th excited state $|E'_c\rangle$ and*

$$\|\mathcal{E}_{\text{state}}(|E_c\rangle) - |E'_c\rangle\| \leq \eta + \mathcal{O}(\gamma^{-1}\epsilon).$$

**Proof:**
This is a slight modification of Lemma 2 of [BH17]. First, the non-degeneracy of the $c$th excited state of $H'$ follows because the $i$th smallest eigenvalues of $H$ and $H'$ differs by at most $\epsilon$ for all $0 \leq i \leq \dim(H) - 1$, and $\epsilon$ satisfies $2\epsilon < \gamma$. Consider $H$ as an unperturbed Hamiltonian and $V := \tilde{\mathcal{E}}^\dagger H' \tilde{\mathcal{E}} - H$ as a perturbation. Then, the perturbed Hamiltonian $H + V = \tilde{\mathcal{E}}^\dagger H' \tilde{\mathcal{E}}$ has a non-degenerate $c$th excited state $\tilde{\mathcal{E}}_{\text{state}}(|E'_c\rangle)$. The first-order perturbation theory for eigenvectors gives $\||E_c\rangle - \tilde{\mathcal{E}}^\dagger_{\text{state}}(|E'_c\rangle)\| \in \mathcal{O}(\gamma^{-1}\epsilon)$. Therefore, it follows that $\|\tilde{\mathcal{E}}_{\text{state}}(|E_c\rangle) - |E'_c\rangle\| = \|\tilde{\mathcal{E}}_{\text{state}}(|E_c\rangle) - \tilde{\mathcal{E}}_{\text{state}}(\tilde{\mathcal{E}}^\dagger_{\text{state}}(|E'_c\rangle))\| \in \mathcal{O}(\gamma^{-1}\epsilon)$ using that $\tilde{\mathcal{E}}_{\text{state}}$ is an isometry and $|E'_c\rangle \in \operatorname{Im}(\tilde{\mathcal{E}}_{\text{state}})$ . Finally, by using $\|\mathcal{E}_{\text{state}} - \tilde{\mathcal{E}}_{\text{state}}\| \leq \eta$, $\|\mathcal{E}_{\text{state}}(|E_c\rangle) - |E'_c\rangle\| \leq \eta + \mathcal{O}(\gamma^{-1}\epsilon)$ follows. $\square$

Now we can take sufficiently small $\epsilon$ and $\eta$ to ensure $\|\mathcal{E}_{\text{state}}(|u\rangle) - |E'_c\rangle\| \leq \zeta' = \zeta - 1/\operatorname{poly}(n)$. Because the Hamiltonian simulation is efficient, the operator norm $\|H'\|$ and the number of qubits of $H'$ are in $\operatorname{poly}(n)$.

**(ii) Verification of the semi-classical property.** We start from a semi-classical subset state $|u\rangle = 1/\sqrt{|S|}\sum_{x \in S}|x\rangle$. We show that after the simulation of the original $k$-local Hamiltonian $H$ where $k \in \mathcal{O}(1)$ by an 2-local Hamiltonian, the corresponding encoding $\mathcal{E}_{\text{state}}(|u\rangle)$ is a semi-classical encoded subset state. To verify that this holds, we sketch the construction of the strong Hamiltonian simulation introduced in [ZA21] up to spatially sparse 2-local Hamiltonians with no Y-terms. First, they construct a spatially sparse 5-local Hamiltonian [OT08] using a quantum phase estimation circuit and some additional modification. This procedure may be thought of as a "Hamiltonian-to-circuit" (then this circuit goes back to a Hamiltonian by circuit-to-Hamiltonian) construction. Then, they perturbatively simulate the spatially sparse Hamiltonian with known techniques in the literature [OT08; CMP18; PM17] to reduce to a 2-local Hamiltonian. We verify that the corresponding encoding of every reduction preserves the semi-classical property of states. In the following, we give an overview of their construction.

**(1) Arbitrary $k$-local Hamiltonian $\rightarrow$ spatially sparse 5-local Hamiltonian.** Let $H$ be a target $\mathcal{O}(1)$-local Hamiltonian. Assume that $H$ can be written as $H = \sum_i E_i|\psi_i\rangle\langle\psi_i|$ where $\{E_i\}$ and $\{|\psi_i\rangle\}$ are the eigenvalues and eigenvectors of $H$. In [ZA21], they showed that there is a spatially sparse quantum circuit $U_{\text{PE}}^{\text{sparse}}$ that approximately estimates the energy of $H$, i.e.

$$U_{\text{PE}}^{\text{sparse}}\sum_i c_i|\psi_i\rangle|0^m\rangle \approx \sum_i c_i|\psi_i\rangle\big|\tilde{E}_i\big\rangle|other\rangle,$$

where $\{c_i\}$ are arbitrary coefficients and $\{|\tilde{E}_i\rangle\}$ are approximations of $\{E_i\}$.

The circuit $U_{\text{PE}}^{\text{sparse}}$ is implemented first by constructing $U_{\text{NN}}^{\text{sparse}}$ that consists of 1D nearest-neighborhood interactions. Then, $U_{\text{NN}}^{\text{sparse}}$ is converted into a spatially sparse circuit using ancilla qubits and swap gates.

Then they combine uncomputation and idling to construct

$$U = (\text{Idling})(U_{\text{PE}}^{\text{sparse}})^\dagger(\text{Idling})U_{\text{PE}}^{\text{sparse}}.$$

They apply circuit-to-Hamiltonian construction for this $U$ to construct a spatially sparse 5-local Hamiltonian $H_{\text{circuit}}$. They use first-order perturbation theory to show that $H_{\text{circuit}}$ simulates $H$ in its low-energy subspace. The encoding of $H_{\text{circuit}}$ to the low energy subspace of $H$ is approximated by the map: $H \rightarrow H \otimes |\alpha\rangle\langle\alpha|$. Here, $|\alpha\rangle$ is a subset state with $poly(n)$-size subset $S'$ that is related to the history state of the idling steps after uncomputation. For details, see the proof of Proposition 2 of [ZA21]. Then, the corresponding encoding of the state is

$$|u\rangle \rightarrow |u\rangle \otimes |\alpha\rangle.$$

The encoded state is also a semi-classical subset state if $|u\rangle$ is a semi-classical subset state.

**(2) Spatially sparse 5-local Hamiltonian → Spatially sparse 10-local real Hamiltonian [CMP18] Lemma 22.** In this simulation, the state is encoded by attaching polynomially many $|+_y\rangle$ where $|+_y\rangle$ is the $+1$ eigenvector of Pauli $Y$ matrix:

$$|u\rangle \to |u\rangle \otimes |+_y\rangle \otimes \cdots \otimes |+_y\rangle. \qquad (7.22)$$

This encoding does not map a semi-classical subset state into a semi-classical state but maps into a semi-classical encoded state. The reason is as follows. Let $V_y$ be a unitary such that $|+_y\rangle = V_y |0\rangle$, and $|u\rangle = 1/\sqrt{|S|} \sum_{x \in S} |x\rangle$. Then, the right-hand side of eq. (7.22) can be written as

$$|u\rangle \otimes |+_y\rangle \otimes \cdots \otimes |+_y\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S \times \{0...0\}} I \otimes \cdots \otimes I \otimes V_y \otimes \cdots \otimes V_y |x\rangle.$$

This is a semi-classical encoded state with a subset $S \times \{0...0\}$ and a local isometry (this is indeed a local unitary) $I \otimes \cdots \otimes I \otimes V_y \otimes \cdots \otimes V_y$.

**(3) Spatially sparse 10-local real Hamiltonian → Spatially sparse 2-local Hamiltonian with no $Y$-terms [OT08; CM16].** This can be done first by simulating the 10-local real Hamiltonian with an 11-local Hamiltonian whose Pauli decomposition does not contain any Pauli $Y$ terms [CMP18, Lemma 40]. In the corresponding encoding, $|1\rangle$ states are attached for the polynomially many mediator qubits introduced in the simulation. Then, we can use subdivision gadgets and 3-to-2 gadgets [OT08]. In this simulation, polynomially many mediator qubits are introduced, and the encoding of states is just to add $|0\rangle$ states for each of the mediator qubits. The resulting Hamiltonian can be written in the form $\sum_{i<j} \alpha_{ij} A_{ij} + \sum_k (\beta_k X_k + \gamma_k Z_k)$, where $A_{ij}$ is one of the interactions of $X_i X_j$, $X_i Z_j$, $Z_i X_j$ or $Z_i Z_j$.

Clearly, by attaching polynomially many $|1\rangle$ and $|0\rangle$ states, a semi-classical encoded subset state is mapped to another semi-classical encoded subset state:

$$|u\rangle \to |u\rangle |x\rangle,$$

where $x \in \{0,1\}^{poly(n)}$ is some string of 0's and 1's corresponding to the attached mediator qubits. This concludes the proof of Proposition 7.7.8. □

# 7.8 Containment in BQP

In this section we show that $\mathsf{GLHLE}(k, c, \delta, \zeta)$ is contained in BQP – i.e. we prove Theorem 7.4.2, which we restate below for convenience. First, we recall some basic facts about combining Hamiltonian simulation with quantum phase estimation.

**7.8.1.** Lemma (Quantum eigenvalue estimation). *Let $H$ be an $\mathcal{O}(\log n)$-local Hamiltonian acting on $n$ qubits, with eigenvectors $|\psi_j\rangle$ and corresponding eigenvalues $\lambda_j \in [0, 1]$. Then there is a quantum algorithm that, given as input an eigenvector $|\psi_j\rangle$, will output with probability at least $p$ an $\epsilon$-approximation of $\lambda_j$ (i.e. an estimate $\tilde{\lambda}_j$ such that $|\tilde{\lambda}_j - \lambda_j| \le \epsilon$) in time $poly(n, 1/\epsilon, 1/p)$.*

This is by now a commonly used quantum algorithm; for details and proofs of correctness, see e.g. [GL22; CM18].

**7.4.2.** Theorem (Containment in BQP of GLHLE).

(i) GLHLE$(k, 0, \zeta, \delta)$ *is contained in* BQP *for* $k = \mathcal{O}(\log(n))$, $\zeta = \Omega(1/poly(n))$, *and* $\delta = \Omega(1/poly(n))$.

(ii) GLHLE$(k, c, \zeta, \delta)$ *for* $c \ge 1$ *is contained in* BQP *when* $k = \mathcal{O}(\log(n))$, $\zeta = \frac{1}{2} + \Omega(1/poly(n))$, *and* $\delta = \Omega(1/poly(n))$.

**Proof:**
Recall that the fidelity of the guiding state with the target eigenstate is at least $\zeta$. Containment in BQP follows from the standard quantum algorithm of Lemma 7.8.1. If we input an arbitrary $n$-qubit state $|\phi\rangle$ to the algorithm of Lemma 7.8.1, it follows that we will obtain an $\epsilon$-approximation of $\lambda_j$ with probability $\ge p \cdot |\langle \psi_j | \phi \rangle|^2$, and hence if we input the guiding state $|u\rangle$, we will obtain an $\epsilon$-approximation to the target eigenstate with probability $\ge p \cdot \zeta$. For the case $c = 0$, with $\zeta = \Omega(1/poly(n))$, we can therefore obtain an $\epsilon$-approximation to the ground state energy with probability $\ge 1/r(n)$, for $r$ some polynomial, in time $poly(n, 1/\epsilon)$. To distinguish the case that the ground state energy is $\le a$ or $\ge b$, with $b - a \ge \delta$, setting $\epsilon < \delta$ is sufficient. With $\delta = 1/\mathcal{O}(poly(n))$, this takes time $poly(n)$, proving part (i) of the theorem.

For the case $c > 0$, with $\zeta = \frac{1}{2} + \Omega(1/poly(n))$, we can choose $p > 1 - 1/poly(n)$ sufficiently large so that with probability at least $p \cdot \zeta > 1 - 1/poly(n)$ we obtain an $\epsilon$-approximation to the target energy $\lambda$. Again by choosing $\epsilon < \delta$ we can decide whether $\lambda \le a$ or $\lambda \ge b$ with probability $\ge \frac{1}{2} + 1/poly(n)$. By repeating $poly(n)$ times and taking a majority vote, we can decide which is the case with probability, say, 2/3 by a Chernoff bound, proving part (ii) of the theorem.

$\square$

# Chapter 8

# The guidable local Hamiltonian problem

In the previous chapter 7, we discussed a "practical" strategy for finding the stationary properties of a quantum mechanical system, with particular focus on the ground state of local Hamiltonians. As a reminder, we restate this strategy, which boils down to the following two-step procedure:

- Step 1 (Guiding state preparation): A classical heuristic algorithm is applied to obtain a guiding state $|u\rangle$, which is hoped to have 'good' fidelity with the ground space.

- Step 2: (Ground state energy approximation): The guiding state $|u\rangle$ is used as input to Quantum Phase Estimation (QPE) to efficiently and accurately compute the corresponding ground state energy.

In the previous chapter we studied the complexity of the second step, which can be formalized as the *Guided Local Hamiltonian Problem* (GLH). We showed that GLH is BQP-hard for a broad range of parameters (such as the fidelity between $|u\rangle$ and the ground space, and the precision $\delta$).
In this chapter we extend on the study of the GLH problem from Chapter 7, diving deeper into the different types of states we consider as "guiding states". Here we study '*Merlinized*' versions of GLH – in which guiding states are no longer given as part of the input but instead are only promised to exist – and use these as a way to gain some insight into important theoretical questions in quantum chemistry and complexity theory. In the subsequent paragraphs, we introduce some of the motivating questions guiding the study of the complexity of these so-called 'guidable' local Hamiltonian problems.

## 8.0.1   Ansätze for state preparation.

Step 1 of the aforementioned two-step procedure generally requires one to have access to classical heuristics capable of finding guiding states whose energies can

be estimated classically (as a metric to test whether candidate states are expected to be close to the actual ground state or not). Furthermore, these 'trial states' should also be preparable as quantum states on a quantum computer, so that they can be used as input to phase estimation in Step 2. In [GL22], inspired by a line of works that focused on the dequantization of quantum machine learning algorithms [Tan19; Chi+20; JLS20], a particular notion of 'sampling-access' to the guiding state $u$ is assumed. Specifically, it is assumed that one can both query the amplitude of an arbitrary basis state, and additionally that one can sample basis states according to their $l_2$ norm with respect to the overall state $u$.[1] Whilst this can be a somewhat powerful model [CHM21], it is closely related to the assumption of QRAM access to classical data, and thus in the context of quantum machine learning (where such access is commonly assumed), it makes sense to compare quantum machine learning algorithms to classical algorithms with sampling access to rule out quantum speed-ups that come merely from having access to quantum states that are constructed from exponential-size classical data.

However, for quantum chemistry and quantum many-body applications, this type of access to quantum states seems to be somewhat artificial. Furthermore, many Ansätze actually used in practice are not of the form of sampleable states. Therefore, the first question that we answer in this chapter is if there is a more natural set of "semi-classical states", which is more closely related to the heuristics used in practice (in Section 8.2).

Finally, one may ask whether the fact that the ground state preparation in Step 1 considers only *classical* heuristics might be too restrictive. *Quantum* heuristics for state preparation, such as variational quantum eigensolvers [Til+22] and adiabatic state preparation techniques [AL18], have attracted considerable attention as possible quantum approaches within the NISQ era. However, one can argue that even in the fault-tolerant setting, such heuristics will likely still be viable approaches to state preparation, in particular when used in conjunction with Quantum Phase Estimation (Corollary 8.4.2).

### 8.0.2   The quantum PCP conjecture.

Arguably the most fundamental result in classical complexity theory is the Cook-Levin Theorem [Coo71; Lev73], which states that satisfiability problems (SAT) are NP-complete. The probabilistically checkable proof (PCP) theorem [Aro+98; AS98], which originated from a long line of research on the complexity of interactive proof systems, can be viewed as a 'strengthening' of the Cook-Levin

---

[1]In this work we slightly abuse notation by making a distinction between the vector representing a quantum state, which we will denote as '$u$', and that same vector instantiated as a quantum state (e.g. living on a quantum computer), which we will denote by '$|u\rangle$'. Of course, these are the same mathematical object ($u = |u\rangle \in \mathcal{C}^{2^n}$), and we only use the different notation to make our theorem statements and proofs clearer.

theorem. In its proof-checking form, it states that all decision problems in NP can be decided, with a constant probability of error, by only checking a constant number of bits of a polynomially long proof string $y$ (selected randomly from the entries of $y$). There are also alternative equivalent formulations of the PCP theorem. One is in terms of *hardness of approximation*: it states that it remains NP-hard to decide whether an instance of constrained satisfaction problem (CSP)[2] is either completely satisfiable, or whether no more than a constant fraction of its constraints can be satisfied.[3] It is straightforward to show that this formulation is equivalent to the aforementioned proof-checking version: one simply samples a clause at random and checks whether it is satisfied, which with constant probability detects a violated clause.

Naturally, quantum complexity theorists have proposed proof-checking and hardness of approximation versions of PCP in the quantum setting. Given the close relationship between QMA and the local Hamiltonian problem, the most natural formulation is in terms of hardness of approximation: in this context, the *quantum* PCP conjecture roughly states that energy estimation of a (normalized) local Hamiltonian up to *constant* precision, relative to the operator norm of the Hamiltonian, remains QMA-hard. This conjecture is arguably one of the most important open problems in quantum complexity theory and has remained unsolved for nearly two decades. Under the assumption NP $\neq$ QMA, the quantum PCP conjecture implies that there exist Hamiltonians for which all low-energy states have no efficient classical description from which their energy can be evaluated efficiently classically. In a recent breakthrough result, the NLTS conjecture was proven to be true, which (amongst other things) means that constant-depth quantum circuits – for which the energies can be computed efficiently, as shown by a standard light cone argument – are not expressive enough to estimate the ground state energies of all Hamiltonians up to even constant precision [ABN22]. However, there have also been some no-go results: for example, a quantum PCP statement cannot hold for local Hamiltonians defined on a grid, nor on high-degree or expander graphs [BH13].

One way to shed light on the validity of the quantum PCP conjecture can be to study PCP-type conjectures for other 'Merlinized' complexity classes. Up until this point, PCP-type conjectures have not been considered for other classes besides NP and QMA.[4] However, there is the beautiful result of [AG19], which studies the possibility of a gap amplification procedure for the class MA by consid-

---

[2]A more general formulation of satisfiability where the question is whether there exists an assignment to a set of variables, each taking values from a fixed domain, that satisfies a collection of specified constraints.

[3]The transformation of a CSP to another one which is hard to approximate is generally referred to as *gap amplification*, and is realised in Dinur's proof of the PCP theorem [Din07].

[4]This is barring a result by Drucker which proves a PCP theorem for the class AM [Dru11]; though there is no direct relationship between QMA and AM and hence it is not clear whether this gives any intuition about the likely validity of the quantum PCP conjecture.

ering a particular type of Hamiltonian: uniform stoquastic local Hamiltonians. The authors show that deciding whether the energy of such a Hamiltonian is exactly zero or inverse polynomially bounded away from zero is MA-hard, but that the problem is in NP when this interval is increased to be some constant. Consequently, this implies that there can exist a gap-amplification procedure for uniform stoquastic Local Hamiltonians (in analogy to the gap amplification procedure for constraint satisfaction problems in the original PCP theorem) if and only if MA = NP – i.e. if MA can be derandomized. Since MA ⊆ QMA, this result also shows that if a gap amplification procedure for the general local Hamiltonian problem would exist that 'preserves stoquasticity', then it could also be used to derandomize MA. Here we ask a similar question about gap amplification as [AG19], however instead of considering gap amplification of stoquastic Hamiltonians we study gap amplification of what we will call *guidable Hamiltonians* with consequences for the class QCMA (Section 8.7).

# 8.1   Preliminaries

## 8.1.1   Notation

We write $\lambda_i(A)$ to denote the $i$th eigenvalue of a Hermitian matrix $A$, ordered in non-decreasing order, with $\lambda_0(A)$ denoting the smallest eigenvalue (ground state energy).

## 8.1.2   Some basic definitions and results from complexity theory

All complexity classes in this part of the thesis will be defined with respect to promise problems (and not languages). Remember that a (promise) problem $A = (A_{\text{yes}}, A_{\text{no}})$ consists of two non-intersecting sets $A_{\text{yes}}, A_{\text{no}} \subseteq \{0, 1\}^*$ (the YES and NO instances, respectively). We have that $A_{\text{inv}} = \{0, 1\}^* \setminus A_{\text{yes}} \cup A_{\text{no}}$ is the set of all invalid instances, and we do not care how a verifier behaves on problem instances $x \in A_{\text{inv}}$ (it can accept or reject arbitrarily, see the paragraph 'oracle access' for a more elaborate discussion on what this entails). All previously defined classes, such as NP, can be understood in this chapter as a promise class. We will require two additional classes in this chapter, the first of which is very similar to NP, but with the additional power that the Turing machine can run in quasi-polynomial time:

**8.1.1.** DEFINITION (NqP). A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NqP if and only if there exists a deterministic quasi-polynomial time Turing machine $M$, i.e. running in time $2^{\mathcal{O}(\log^c(n))}$ for some constant $c > 0$, and a polynomial $p$, where $M$ takes as input a string $x \in \{0, 1\}^*$ and a $p(|x|)$-bit witness $y$ and decides on acceptance or rejection of $x$ such that

- if $x \in A_{\text{yes}}$ then there exists a $y \in \{0,1\}^{p(n)}$ such that $M$ accepts $(x,y)$.

- if $x \in A_{\text{no}}$ then for every $y \in \{0,1\}^{p(n)}$ we have that $M$ rejects $(x,y)$.

The second class we require is a more specified version of QCMA, which ,following Section 2.3.2, will be defined in the quantum circuit model.

**8.1.2.** DEFINITION (UQCMA). The class UQCMA$[c,s]$ (unique QCMA) has the same definition as QCMA but with the extra constraint that if $x \in A_{\text{yes}}$ then there exists only a single string $y^*$ such that $V_n$ accepts $(x, |y^*\rangle)$ with probability $\geq c(= 2/3)$, and otherwise for all $y \neq y^*$ we have that $V$ accepts $(x, |y\rangle)$ with probability $\leq s(= 1/3)$.

Unlike QMA, it is known that a lot of the behaviours exhibited by the classical complexity classes NP and MA hold for QCMA as well. An example of this, and one that we use later, is a result from [Aha+22] stating that there exists a randomized reduction from QCMA to UQCMA, analogous to Valiant-Vazirani theorem for NP [VV85].

**8.1.3.** LEMMA (Randomized reduction from QCMA to UQCMA [Aha+22]). *Let $\langle U_n, p_1, p_2 \rangle$ describe a promise problem in QCMA, where $U_n$ is the description of a quantum circuit which takes an input $x$ of length $|x| = n$ and a witness $y$ with length $|y| = poly(n)$. Denote $p_1$ and $p_2$ with $p_1 - p_2 = 1/poly(n)$ for the completeness and soundness, respectively. Then there exists a randomized reduction to a UQCMA instance $\langle \tilde{U}_n, \tilde{p}_1, \tilde{p}_2 \rangle$, with $\tilde{p}_1 - \tilde{p}_2 = 1/poly(n)$ such that:*

- *If there exists a witness $y$ which makes $U_n$ accept $(x,y)$ with probability $\geq p_1$ then there exists a single $y^*$ which makes $\tilde{U}_n$ accept $(x, y^*)$ with probability $\geq \tilde{p}_1$ and accept $(x,y)$ for all other $y \neq y^*$ with probability $\leq \tilde{p}_2$.*

- *If $U_n$ accepts with probability $\leq p_2$ for all $y$ then $\tilde{U}_n$ accepts with probability $\leq \tilde{p}_2$ for all $y$.*

*This randomized reduction succeeds with probability $\Omega(1/|y|)$.*

Another one of these properties is the equivalence of one-sided and two-sided error in the acceptance and rejection probabilities, which just as in the MA setting holds for QCMA (assuming robustness under the choice of the universal gate-set that is used to construct the verification circuits). Formally, this is established via the following lemma.

**8.1.4.** LEMMA (Perfect completeness QCMA [Jor+12]). *Let $\mathcal{G} = \{H, X, \textit{Toffoli}\}$ be a fixed gate set. For any $c, s \in [0,1]$ satisfying $c - s := \delta \geq 1/q(n)$ for some polynomial $q : \mathbb{N} \to \mathbb{R}_{>0}$, we have that*

$$\text{QCMA}_{\mathcal{G}}[c,s] \subseteq \text{QCMA}_{\mathcal{G}}[1, s'],$$

*where $s' = 1 - \frac{1}{2}\delta^2 - \frac{1}{2}\delta^3$.*

### 8.1.3 Locality reducing perturbative gadgets

We will make use of perturbative gadgets to reduce the locality of our constructed Hamiltonian. We will rely on the same construction, based on strong Hamiltonian simulation, as we did in Chapter 7. We can summarize this construction in the following Lemma:

**8.1.5.** LEMMA ('Classical evaluability'-preserving eigenstate encodings).
*Suppose $H$ is an arbitrary $k$-local Hamiltonian on $n$ qubits with a non-degenerate ground state $|g\rangle$ separated from excited states by a gap $\gamma$. Then $H$ can be efficiently $(\Delta, \eta, \epsilon)$ simulated by a 2-local Hamiltonian $H'$ on $m = poly(n)$ qubits which has a non-degenerate ground state $|g'\rangle$, such that*

$$\|\mathcal{E}_{state}(|g\rangle) - |g'\rangle\| \leq \eta + \mathcal{O}(\gamma^{-1}\epsilon),$$

*where $\mathcal{E}_{state}(\cdot)$ appends only states of a semi-classical form as a tensor product to $|g\rangle$, i.e. preserves the classical evaluability as in Definition 8.2.2.*

**Proof:**
This follows immediately from the proof of Proposition 7.7.8 in Chapter 7, while making the observation that all encodings up to the Spatially sparse 2-local Hamiltonian (with Pauli interactions with no $Y$-terms) only append states that satisfy the definition of poly-sized subset states (see the proof of Theorem 8.5.3 in the main text) to the original eigenstate of $H$. □

## 8.2 Classically evaluatable states

Our first contribution is a new class of "semi-classical" quantum states that is more closely related to what is used in practice. Let us first introduce Gharibian and Le Gall's definition of query and sampling access to quantum states [GL22], which slightly generalizes the original definition as first proposed by Tang used to dequantize quantum algorithms for recommendation systems [Tan19].

**8.2.1.** DEFINITION (Query and sampling access, from [GL22]). We say that we have *query and $\xi$-sampling access* to a vector $u \in \mathbb{C}^N$ if the following three conditions are satisfied:

(i) we have access to an $\mathcal{O}(poly(\log(N))$-time classical algorithm $\mathcal{Q}_u$ that on input $i \in [N]$ outputs the entry $u_i$.

(ii) we have access to an $\mathcal{O}(poly(\log(N))$-time classical algorithm $\mathcal{SQ}_u$ that samples from a probability distribution $p : [N] \to [0,1]$ such that

$$p(j) \in \left[(1-\xi)\frac{|u_j|^2}{\|u\|^2}, (1+\xi)\frac{|u_j|^2}{\|u\|^2}\right]$$

for all $j \in [N]$.

(iii) we are given a real number $m$ satisfying $|m - \|u\|| \leq \xi\|u\|$.

We simply say that we have sampling access to $u$ (without specifying $\xi$) if we have 0-sampling access.

Here we propose a new class of quantum states, conceptually different from those of Definition 8.2.1, which we will call *classically evaluatable quantum states.* Our main motivations for doing so are the following:

1. It seems rather difficult to find Ansätze that are used in practice for ground state energy estimation that satisfy all conditions of Definition 8.2.1. As one of the main motivations of this work is to investigate the power of quantum versus classical state preparation when one has access to Quantum Phase Estimation, we wanted to define a class of states that can both be prepared efficiently on a quantum computer and which contains a large class of Ansätze commonly used in practice.

2. Analogous to Dinur's construction, one would expect that determining if a local Hamiltonian has ground state energy (exponentially close to) zero or some constant away from zero is QMA-hard if the quantum PCP conjecture is true. However, there are arguments from physics[5] on why one might expect this problem to be in NP [PH11]. To study the question of containment in NP it is necessary to be able to work with states within a deterministic setting, and therefore it does not make sense to rely on a form of sampling access which inherently relies on a probabilistic model of computation.

There is a third benefit of these states, which will not be discussed in this thesis, but which we will state here for completeness. Being able to study containment in NP comes with the additional advantage of being able to make statements about whether the problem admits a PCP by the classical PCP theorem. No such theorem is currently known for MA, this is exploited further in [WFC23] where the authors introduce a new type of 'quantum' PCP. This new type of quantum PCP will not be discussed in this thesis, but we would like to refer the interested reader to [WFC23].

We will define these quantum states in a slightly more general setting for completeness – by allowing for probabilistic computation of expectation values as well – but this will not be important for the remainder of this work.

**8.2.2.** DEFINITION ($\epsilon$-classically evaluatable and quantumly preparable states). We say a state $u \in \mathbb{C}^{2^n}$ is $\epsilon$**-classically evaluatable** if

---

[5]In this setting the LH problem becomes equivalent to determining whether the free energy of the system becomes negative at a finite temperature. One expects then that at such temperatures, the system loses its quantum characteristics on the large scale, making the effects of long-range entanglement become negligible. Hence, this means that the ground state of such a system should have some classical description, which places the problem in NP [Ara11].

(i) there exists a classical description of $u$, denoted as desc($u$), which requires at most $poly(n)$ bits to write down, and

(ii) there exists a classical probabilistic algorithm $\mathcal{OQ}_u$ which, given desc($u$) and the matrix elements of some $k$-local observable $O$ with $\|O\| \leq 1$, computes an estimate $\hat{z}$ such that $|\hat{z} - \langle u| O |u\rangle| \leq \epsilon$ in time $poly(n, \epsilon, 2^k)$, with success probability $\geq 2/3$.

Furthermore, we say a state $u \in \mathbb{C}^{2^n}$ is also **quantumly preparable** if

(iii) there exists a quantum circuit $V$ of at most $poly(n)$ 1- and 2-qubit gates that prepares $u$ as a quantum state, i.e. $|u\rangle$. The description of $V$ can be computed efficiently using some efficient classical algorithm $\mathcal{A}_V$, which only takes desc($u$) as an input.

Finally, if $\epsilon = 0$ and the algorithm used in (ii) is deterministic instead of probabilistic, we simply say that $u$ is **classically evaluatable**.

Note that it is not required that $u$ is normalized, however by requirement (ii) it is possible to calculate the norm of $u$. Normalization is of course required for $u$ to be quantumly preparable. Also note that if condition (iii) holds, condition (ii) (for $\epsilon > 0$) is no longer necessary in order to work with the class of states as a suitable Ansatz provided that one has access to a quantum computer, since there exist quantum algorithms to estimate the expectation values of the observables up to arbitrarily precise inverse polynomial precision. However, the current definition allows one to adopt the two-step classical-quantum procedure of *classical* Ansatz generation and *quantum* ground state preparation, as described in Section 7.2.

To demonstrate the practical relevance of Definition 8.2.2, we give four examples of Ansätze which all satisfy the required conditions to be ($\epsilon$-)classically evaluatable and quantumly preparable. The first two examples will also be perfectly samplable, as in Definition 8.2.1, of which the proofs are given in Appendix 8.A.

**8.2.3.** EXAMPLE (Matrix-product states; bounded bond and physical dimensions). Matrix-product states are quantum states of the following form

$$|u\rangle = \sum_{\{s\}} \mathrm{Tr}[A_1^{(s_1)} A_2^{(s_2)} \ldots A_n^{(s_n)}] |s_1, \ldots, s_n\rangle,$$

where $s_i$ are qudits of 'physical' dimension $p$ ($s_i \in \{0, 1, \ldots, p-1\}$), the $A_i^{(s_i)}$ are complex, square matrices of bond dimension $D$, and $n$ denotes the total number of qudits. We say that the bond dimension is bounded if it is at most polynomial in $n$, and that the physical dimension is bounded if it is taken to be some constant independent of $n$. MPS are also 0-samplable, which is shown in Appendix 8.A.

*Conditions check:*

(i) The MPS is fully determined by the set of matrices $\{A_i^{s_i}\}$, and can be described explicitly using at most $npD^2 = poly(n)$ complex numbers.

(ii) One can compute the inner product $\langle u|M|u\rangle$ in time at most $n(2D^3\chi p + D^2\chi^2 p^2)$ for any (even $n$-local) operator $M$ having a matrix product operator decomposition with bond dimension $\chi$ [Sch11; Orú14]. Since $O$ is $k$-local, it can be represented by an MPO with bond dimension at most $p^k$, and so $\langle u|O|u\rangle$ can be computed in a maximum time of $n(2D^3 p^{k+1} + D^2 p^{2k+2}) = poly(n, D, 2^k)$ when $p$ is constant.

(iii) An MPS on $n$ qubits with bond dimensions $D$ can be prepared on a quantum computer up to distance $\epsilon$ using at most $\mathcal{O}(nD\log(D)^2\log(n/\epsilon))$ 1- and 2-qubit gates and requiring $\lceil\log(D)\rceil$ additional ancilla qubits. A method for constructing such a circuit can be found in Appendix 8.B and is based on [Sch+05].

**8.2.4.** EXAMPLE (Stabilizer states). Gottesmann and Knill [Got98] showed that there exists a class of quantum states, containing states that exhibit large entanglement, that can be efficiently simulated on a classical computer. These states are called *stabilizer states* and are those generated by circuits consisting of Clifford gates, $\mathcal{C} = \langle\text{CNOT}, H, S\rangle$ where $S = \sqrt{Z}$ is a phase gate, starting on a computational basis state. Any measurement of local Pauli's on these states can be efficiently classically simulated. Amongst other things, stabilizer states have been used to formulate error correcting codes [Ste03], study entanglement [Ben+96], and in evaluating quantum hardware through randomised benchmarking [Kni+08]. Stabilizer states are also 0-samplable, again shown in Appendix 8.A.

*Conditions check:*

(i) Any stabilizer state can be described by a linear depth circuit consisting of Clifford gates starting on the $|0^n\rangle$ state [MR18]. A possible description of such a circuit is a list of tuples $(q_1, q_2, t, g)$, where $q_1$ (resp. $q_2$) denotes the first (resp. second) qubit that $g \in \mathcal{C}$ acts on at depth $t$. This description takes at most $\tilde{\mathcal{O}}(n^2)$ bits to write down.

(ii) The Gottesman-Knill [Got98] theorem shows that stabilizer states allow for strong classical simulation and efficient classical computation of probabilities for Pauli measurements. This in particular allows for the calculation of expectation values in time $poly(2^k)$.

(iii) The description is given as a quantum circuit, which can be implemented to prepare the quantum state.

We will now give two examples of Ansätze that have been shown to not be $\xi$-samplable, even up to some large constant values of $\xi$.

**8.2.5.** EXAMPLE (Constant depth quantum circuits). Constant depth quantum circuits are circuits that, given some fixed gate set $\mathcal{G}$ with just local operations, are only allowed to apply at most $t = \mathcal{O}(1)$ consecutive layers of operations from $\mathcal{G}$ on some initial quantum state, which we take to be the all-zero state $|0\dots0\rangle$. An example of constant depth quantum circuits that are used as classical Ansätze would be the simple case of the *product state Ansatz*, where one only considers one-qubit gates applied per site. Product state Ansätze are widely used in classical approximation algorithms to Local Hamiltonian problems, see for example [BH13; GP19]. In [TD04] it was shown that the ability to perform approximate weak sampling from the output of a constant depth quantum circuit up to relative error $0 < \xi < 1/3$ implies that $\mathsf{BQP} \subseteq \mathsf{AM}$, which means that it is unlikely that constant depth quantum circuits are $\xi$-samplable for any $\xi < 1/3$.

*Conditions check:*

(i) By definition.

(ii) $\langle u| O |u\rangle = \langle 0| U^\dagger O U |0\rangle$, where $U^\dagger O U$ is a $k2^t$-local observable (via a light-cone argument), and hence we can compute $\langle 0| U^\dagger O U |0\rangle$ in time $\mathcal{O}\left(2^{\mathcal{O}(k2^t)} \cdot poly(n)\right)$ which is $poly(2^k)$ if $t = \mathcal{O}(1)$.

(iii) This holds again by definition.

By combining Example 8.2.4 and Example 8.2.5 we find that any state of the form $UC |0^n\rangle$, with $U$ a constant-depth circuit and $C$ a Clifford circuit, is also classically evaluatable and quantumly preparable. Our final example is of a class of states that are not perfectly classically evaluatable, but are $\epsilon$-classically evaluatable for any $\epsilon = 1/poly(n)$.

**8.2.6.** EXAMPLE (Instantaneous quantum polynomial (IQP) circuits). IQP circuits start in $|0^n\rangle$ and apply a polynomial number of local gates that are diagonal in the $X$-basis, followed by a computational basis measurement [BJS10]. An equivalent definition would be to consider circuits with gates that are diagonal in the $Z$-basis, but then sandwiched in two layers of Hadamard gates (again followed by a measurement in the computational basis). It is well known that IQP circuits are difficult to sample from: if IQP circuits could be weakly simulated to within multiplicative error $1 \leq c < \sqrt{2}$, then the polynomial hierarchy would collapse to its third level [BJS10]. Hence, they are not $\xi$-samplable for any $\xi < \sqrt{2} - 1$. However, we will show that states generated by IQP circuits are $\epsilon$-classically evaluatable for all $\epsilon = 1/poly(n)$.

*Conditions check:*

(i) This follows by definition, since all gates are local and there are only a polynomial number of them.

(ii) This is a corollary from Theorem 3 in [BJS10], where it is shown that one can exactly sample basis states on $\mathcal{O}(\log n)$ qubits according to their $l_2$-norm. Let $C$ be an IQP-circuit of $n$ qubits which produces the state $|u\rangle = C|0^n\rangle$ before the final measurement, and let $S \subseteq [n]$ with $|S| = k$ be the qubits on which a $k$-local observable $O$ acts. Following the proof of Theorem 3 in [BJS10], the state right before the last layer of Hadamard is given by

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in S, y \in [n]\setminus S} e^{if(x,y)} |x,y\rangle,$$

where the pair $x, y \in \{0,1\}^n$ denotes the bit string state corresponding to the concatenation (with the correct indexing) of the bit strings $x \in \{0,1\}^{|S|}$ and $y \in \{0,1\}^{n-|S|}$. Here $f(x,y)$ is a phase function which can be computed efficiently, by accumulating the relevant diagonal entries of the successive commuting gates. Since $O$ does not act on the qubits with indices $[n] \setminus S$, and they only get acted upon by Hadamards, further measurements on this register should not influence any POVM that only acts on $S$ by the no-signaling principle. By this observation, the protocol is now very simple: one samples a random bit string $y' \in \{0,1\}^{n-|S|}$ and computes the random variable

$$X_i = \frac{1}{2^{|S|}} \sum_{x,x' \in \{0,1\}^{|S|}} \langle x| e^{-if(x,y')} H^{\otimes |S|} O H^{\otimes |S|} e^{if(x,y')} |x'\rangle,$$

which can be done exactly in time $poly(2^k)$ since $f(x,y')$ can be computed efficiently. Since $\|O\| \leq 1$, We have that $\mathbb{E}[X_i^2] \leq 1$ and $|\mathbb{E}[X_i]| \leq 1$, and therefore $\mathrm{Var}[X_i] = \mathbb{E}[X_i^2] - \mathbb{E}[X_i]^2 \leq 2$. Therefore, taking $s = c/\epsilon^2$ samples of $X_i$ (which are independent random variables) and computing $\hat{z} = \frac{1}{s} \sum_{i \in [s]} X_i$ ensures that

$$|\hat{z} - \langle u| O |u\rangle| \leq \epsilon,$$

with probability $\geq 2/3$, provided that $c \geq 6$. This follows from a simple application of Chebyshev's inequality.

(iii) This follows also by definition.

In general quantum states will *not* be classically evaluatable (as that would imply QMA = NP as they could be used as witnesses for the QMA-hard local Hamiltonian problem), and some other notable examples of classes of states which are not expected to be classically evaluatable are Projected Entangled Pair States (PEPS)

(since computing expectation values of local observables is #P-hard [Sch+07]) and collections of local reduced density matrices (to check whether they are consistent with a global quantum state is QMA-hard [Liu06; BG22]).

We have seen that constant-depth quantum circuits are not even approximately samplable (under the conjecture that BQP $\not\subset$ AM [TD04]). We can formalize this in the following proposition which relates $\xi$-samplable states to $\xi$-classically evaluatable states. First, we need the following Lemma which is almost a direct corollary of the proof of Theorem 4.1 in [GL22].[6]

**8.2.7.** LEMMA. *Given query access to a s-sparse Hermitian matrix $A \in \mathbb{C}^{N \times N}$ with $\|A\| \leq 1$, query and $\xi$-sampling access to a vector $u \in \mathbb{C}^N$ with $\|u\| \leq 1$ as per definition 8.2.1, for any $\xi \leq \epsilon/8$ and $\epsilon \in (0,1]$ there exists a classical randomized algorithm which with probability $\geq 1 - 1/\operatorname{poly}(N)$ outputs an estimate $\hat{z} \in \mathbb{R}$ such that*

$$\left| \hat{z} - u^\dagger A u \right| \leq \epsilon$$

*in time $\mathcal{O}^*(s/\epsilon^2)$.*

**Proof:**
Since we have query access to the entries of $A$ and those of $u$, we can compute the $i$th entry of the vector $Au$ in time $\mathcal{O}(s)$. The lemma follows then directly from the proof of Theorem 4.1 in [GL22], taking $v = u$ and replacing their $P(\sqrt{A^\dagger A})$ with our Hermitian $A$ (this also makes the estimation of the imaginary part in the proof in [GL22] redundant). $\qquad\square$

**8.2.8.** THEOREM. *For any $\xi > 0$, any $\xi$-samplable state is also $\mathcal{O}(\xi)$-classically evaluatable. On the other hand, there exist states that are perfectly classically evaluatable but not $\xi'$-samplable for all $0 < \xi' < 1/3$, unless BQP $\subseteq$ AM.*

**Proof:**
Let $u \in \mathbb{C}^N$ be a $\xi$-samplable state with $N = 2^n$. The first part of the proposition follows by checking the two conditions.

(i) $u$ is described by giving the algorithms $\mathcal{Q}_u$ and $\mathcal{SQ}_u$. Both these algorithms run in $\mathcal{O}(\operatorname{poly}(\log(N)))$-time, which implies that both have an efficient description of length at most $\mathcal{O}(\operatorname{poly}(\log(N)))$ (in terms of local classical operations, i.e. logic gates).

(ii) This follows directly from Lemma 8.2.7, since the global operator representation of a $k$-local observable $O$ acting on a $k$-subset of $n$ qubits can be written as $N \times N$ Hermitian matrix $A = O \otimes I$ where $A$ has sparsity $s = 2^k$.

---

[6]We cannot use their theorem directly, as it only works for even polynomials and we are interested in the polynomial $P(x) = x$.

This shows that any $\xi$-samplable state is at least $8\xi$-classically evaluatable. The second part follows directly from [TD04], Theorem 3, which shows that the ability to perform approximate weak sampling from the output of a constant depth quantum circuit up to relative error $0 < \xi < 1/3$ implies that $\mathsf{BQP} \subseteq \mathsf{AM}$, obstructing the ability to satisfy condition (ii) in Definition 8.2.1. By Example 8.2.5, we already showed that constant-depth quantum circuits produce classically evaluatable states, completing the proof. □

This gives rise to a (conjectured) hierarchical structure of states as depicted in Figure 8.1. An interesting observation is a supposedly significant leap in the hierarchy when we allow for a small error $\epsilon$ in the definition of $\epsilon$-classically evaluatable states. A straightforward way to explain this is by considering how it affects our ability to determine a global property of a quantum state, like its energy with respect to a Hamiltonian $H$.

Let $H$ be a sum of $m$ log-local terms, i.e. $H = \sum_{i=0}^{m-1} H_i$, satisfying $\|H\| \leq 1$. If one wants to evaluate the energy of an $\epsilon$-classically evaluatable state with respect to $H$ up to accuracy $\epsilon'$, then $\epsilon$ has to be less than $\epsilon'/m$ since in the worst case the error grows linearly with the number of terms. Instead, $\xi$-samplable states have a requirement on the accuracy of sampling, which is a property of the global state. [GL22] shows that this property can be used for energy estimation, where the requirement on $\xi$ only depends on the precision with which one wants to measure the energy. We see this reflected in Theorem 8.2.8, which shows that if a state has the property of being $\xi$-samplable this implies that the state is $\mathcal{O}(\xi)$-classically evaluatable, but not the other way around. However, we are not aware of any classes of states which are provably only $\xi$-samplable for a constant, but small, $\xi > 0$ (all examples that we give in this work are in fact 0-samplable).

For the remainder of our work, we will focus on (0-)classically evaluatable states, which by Definition 8.2.2 means that $\mathcal{OQ}_u$ is deterministic. A notable advantage of this approach, as opposed to 0-samplable states, lies in its compatibility with deterministic algorithms, allowing us to give $\mathsf{NP}$ containment results (see Section 8.6).

## 8.3 The guidable local Hamiltonian problem

Our main focus is on a new family of local Hamiltonian problems, which we call *Guidable local Hamiltonian problems.* Let us define this class of local Hamiltonian problems, which can be viewed as 'Merlinized' versions of the original guided local Hamiltonian problem. We make a distinction between different types of promises one can make with respect to the existence of guiding states: we either assume that the guiding states are of the form of Definition 8.2.2 (with or without the promise that the states are also quantumly preparable), or that there exists an efficient quantum circuit that prepares the guiding state.
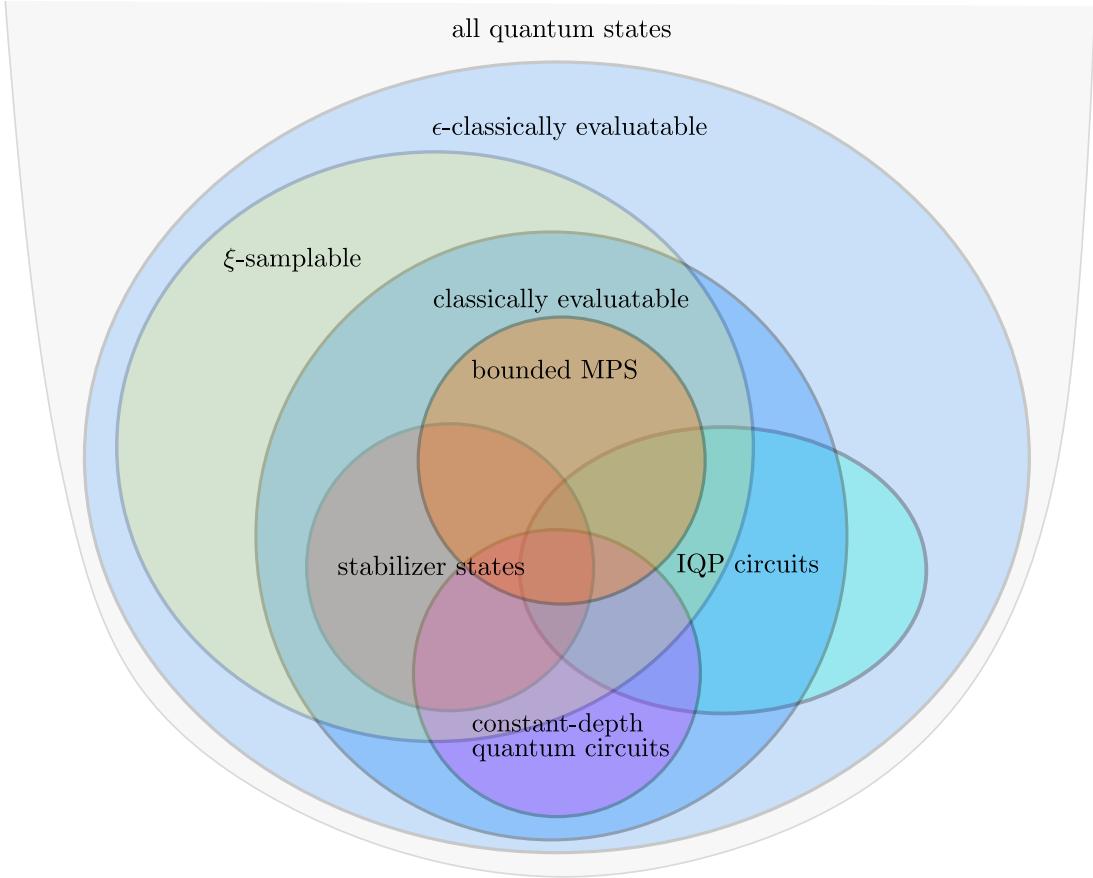
Figure 8.1: Visualization of the (conjectured) relations between classes of quantum states considered in this work, given a Hilbert space of a fixed dimension. For MPS, we only consider states with polynomially-bounded bond and local dimension. We take $\xi \leq \epsilon/8 \leq 1/3$, such that by Theorem 8.2.8 we have that (i) all $\xi$-samplable states are also $\epsilon$-classically evaluatable and (ii) constant-depth and IQP circuits are not $\xi$-samplable. One also expects that there are quantum states (which can be prepared by a polynomial time quantum circuit) which are neither classically evaluatable nor samplable, or else QMA (QCMA) would be in NP or MA, respectively.

**8.3.1.** DEFINITION (Guidable Local Hamiltonian Problems). Guidable Local Hamiltonian Problems are problems defined by having the following input, promise, one of the extra promises and output:

**Input:** A $k$-local Hamiltonian $H$ with $\|H\| \leq 1$ acting on $n$ qubits, threshold parameters $a, b \in \mathbb{R}$ such that $b - a \geq \delta > 0$ and a fidelity parameter $\zeta \in (0, 1]$.

**Promise:** We have that either $\lambda_0(H) \leq a$ or $\lambda_0(H) \geq b$ holds, where $\lambda_0(H)$ denotes the ground state energy of $H$.

**Extra promises:** Denote $\Pi_{\mathrm{gs}}$ for the projection on the subspace spanned by the ground state of $H$. Then for each problem class, we have that either one of the following promises hold:

1. There exists a classically evaluatable state $u \in \mathbb{C}^{2^n}$ for which $\|\Pi_{\mathrm{gs}} u\|^2 \geq \zeta$. Then the problem is called the **Classically Guidable Local Hamiltonian Problem**, shortened as $\mathsf{CGaLH}(k, \delta, \zeta)$. If $u$ is also quantumly preparable, we call the problem the **Classically Guidable and Quantumly Preparable Local Hamiltonian Problem**, shortened as $\mathsf{CGaLH}^*(k, \delta, \zeta)$.

2. There exists a unitary $V$ implemented by a quantum circuit composed of at most $T = poly(n)$ gates from a fixed gate set $\mathcal{G}$ that produces the state $|\phi\rangle = V|0\rangle$ (with high probability), which has $\|\Pi_{\mathrm{gs}} |\phi\rangle\|^2 \geq \zeta$. Then the problem is called the **Quantumly Guidable Local Hamiltonian problem**, shortened as $\mathsf{QGaLH}(k, \delta, \zeta)$.

**Output:**

- If $\lambda_0(H) \leq a$, output YES.

- If $\lambda_0(H) \geq b$, output NO.

A guidable local Hamiltonian problem variant for a different class of guiding states was already introduced in [GL22], where they consider the class of samplable states as in Definition 8.2.1 as guiding states. They didn't give any hardness results on this problem.

The $\mathsf{QGaLH}(k, \delta, \zeta)$ problem is very similar to the Low Complexity Low Energy States (LCSES) problem from [WJB03], but differs in some key ways. In the low complexity low energy states problem one is promised that for all states $\{|\phi\rangle\}$ that can be prepared from $|0\ldots0\rangle$ with a polynomially bounded number of gates from a fixed gate set, one has that either there exists at least one such $|\phi\rangle$ such that $\langle\phi| H |\phi\rangle \leq a$ or for all these $|\phi\rangle$ we have $\langle\phi| H |\phi\rangle \geq b$. Instead, in $\mathsf{QGaLH}(k, \delta, \zeta)$ one is promised that there exists a state $|\psi\rangle$ which can be prepared efficiently on a quantum computer that has fidelity $\zeta$ with the ground space of $H$. This promise in the fidelity does not imply that the energy of this $|\psi\rangle$ is necessarily low, as it might have a large fidelity with states in the high-energy spectrum of $H$. Nevertheless, it does imply that in the YES-case there exists a low complexity

low energy state $|\phi\rangle$. One can make use of the state $|\psi\rangle$ that has significant overlap with the ground state and use Lin and Tong's filtering method [LT20] to project $|\psi\rangle$ onto a state $|\phi\rangle$ with energy at least inverse polynomially close to the ground state (which implies $|\phi\rangle$ can be prepared by a quantum circuit). However, in the NO-case this promise on the fidelity implies that every possible state $|\psi\rangle$ has energy $\langle\psi| H |\psi\rangle \geq b - \mathcal{O}(1/\exp(n))$, as even in the NO-case it is still possible to approximate the ground state energy up to polynomial precision. This is different from the NO-case of the low complexity low energy states problem, where there might exist states with energy lower than $a$, as long as these states are not preparable by a polynomial-time quantum circuit, making the $\mathsf{QGaLH}(k, \delta, \zeta)$ problem more restrictive than the low complexity low energy states problem. In principle, this could be remedied by relaxing the requirement in $\mathsf{QGaLH}(k, \delta, \zeta)$ from having fidelity with the ground space to having fidelity with the space of states with sufficiently low energy in the YES-case only. All our results that follow would still hold, and this new problem could then be seen as a generalisation of the low complexity low energy states problem.

## 8.4   Summary of main results

### 8.4.1   QCMA completeness of the guidable local Hamiltonian problem

With these definitions we can give our main result with respect to guidable local Hamiltonian problems:

**8.4.1.** THEOREM (Complexity of guidable local Hamiltonian problems). *For $k = 2$ and $\delta = 1/\operatorname{poly}(n)$, we have that both $\mathsf{CGaLH}^*(k, \delta, \zeta)$ and $\mathsf{QGaLH}(k, \delta, \zeta)$ are $\mathsf{QCMA}$-complete when $\zeta \in (1/\operatorname{poly}(n), 1 - 1/\operatorname{poly}(n))$.*

The proof of which can be found in Corollary 8.5.4 and Theorem 8.5.6. A direct corollary of the above theorem is the following.

**8.4.2.** COROLLARY (Classical versus quantum state preparation). *When     one has access to a quantum computer (and in particular quantum phase estimation), then having the ability to prepare any quantum state preparable by a polynomially-sized quantum circuit is no more powerful than the ability to prepare states from the family of classically evaluatable and quantumly preparable states, when the task is to decide the local Hamiltonian problem with precision $1/\operatorname{poly}(n)$.*

It should be noted that our result does *not* imply that all Hamiltonians which have efficiently quantumly preparable guiding states also necessarily have guiding states that are classically evaluatable. All this result says is that for any instance

of the guidable local Hamiltonian problem with the promise that there exist guiding states that can be efficiently prepared by a quantum computer, there exists an (efficient) *mapping* to another instance of the guidable local Hamiltonian problem with the promise that there exist guiding states that are classically evaluatable and quantumly preparable. Whilst this reduction is efficient in the complexity-theoretic sense, it might not be for practical purposes, as it would likely remove all the physical structure present in the original Hamiltonian. Hence, the main implication of our result is not that these kinds of reductions are of practical merit, but that at least from a complexity-theoretic point of view the aforementioned classical-quantum hybrid approach of guiding state selection through *classical* heuristics combined with *quantum* energy estimation is at least as powerful as using quantum heuristics for state preparation instead.

## 8.4.2 Classical containment of CGaLH

We complement our quantum hardness results with classical containment results (of the classically guidable local Hamiltonian problem), obtained through a deterministic dequantized version of Lin and Tong's ground state energy estimation algorithm [LT20]. Here CGaLH is just as CGaLH* but without the promise of the guiding state being quantumly preparable.

**8.4.3.** THEOREM (Classical containment of CGaLH). *CGaLH$(k, \delta, \zeta)$ is NP-complete for $k = \mathcal{O}(\log(n))$, and constants $\delta \in (0, 1]$ and $\zeta \in (0, 1]$. Furthermore, when $\zeta = 1/poly(n)$ we have that CGaLH$(k, \delta, \zeta)$ is in NqP.*

The proof of this theorem can be found in Section 8.6. Through a more careful analysis of when exactly the quantum hardness vanishes, the picture of Figure 8.2 emerges, which characterises the complexity of CGaLH*$(k, \delta, \zeta)$ for relevant parameter settings in the desired precision and promise on the fidelity. One additional result to mention, is that when the overlap between the guiding state becomes very close to one, $\zeta = 1 - 1/\exp(n)$, the problem remains in NP even when the promise gap becomes polynomially small, $\delta = 1/poly(n)$ (Theorem 8.6.7).[7]

---

[7]After this work, Jiang published a work on a similar problem for a different class of states than we consider [Jia23]. Jiang shows that if the ground state admits a polynomial-time algorithm to compute the amplitudes, the class of states for which this is possible being called *succinct*, the corresponding local Hamiltonian problem is MA-complete even in the inverse polynomial precision setting. However, since our proof of Theorem 8.5.3 uses as a witness polynomial-sized subset states, which are succinct, it also shows that Jiang's problem is QCMA-hard when it is only promised that there exists a succinct state with only at most $1 - 1/poly(n)$ overlap with a ground state. Hence, Jiang's result is similar to our Theorem 8.6.7 in that if the ground state itself becomes (exponentially close to) a special class of states, the problem becomes classically solvable.
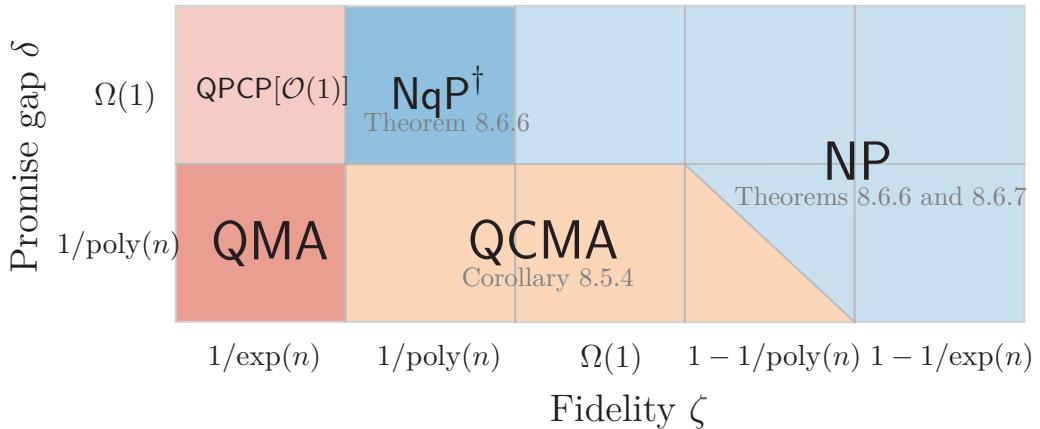
Figure 8.2: Complexity characterization of $\mathsf{CGaLH}^*(k, \delta, \zeta)$ over parameter regime $\delta$ and $\zeta$, for $k = \mathcal{O}(1)$. Any classification indicates completeness for the respective complexity class, except for $\mathsf{NqP}$, for which we only know containment (indicated by the '$\dagger$'). Here completeness for certain parameter combinations means that for all functions of the indicated form, the problem is contained in the complexity class, and for a subset of these functions the problem is also hard. The results for $\mathsf{QPCP}[\mathcal{O}(1)]$ and $\mathsf{QMA}$ follow directly from [Aha+09] and [KSV02].

### 8.4.3 Two implications for the quantum PCP conjecture

Finally, we use our obtained results on $\mathsf{CGaLH}$ to obtain an interesting result and a new conjecture related to the quantum PCP conjecture.

First, our classical containment results of $\mathsf{CGaLH}^*$ with constant promise gap can be viewed as no-go theorems for a gap amplification procedure for $\mathsf{QPCP}$ having certain properties, as illustrated by the following result.

**8.4.4.** THEOREM. *(Informal; No-go results for Hamiltonian gap amplification)*
*There cannot exist a gap amplification procedure for the local Hamiltonian problem that preserves the fidelity between the ground space of the Hamiltonian and any classically evaluatable state up to a*

- *multiplicative constant, unless $\mathsf{QCMA} = \mathsf{NP}$, or*

- *inverse-polynomial multiplicative factor, unless $\mathsf{QCMA} \subseteq \mathsf{NqP}$.*

A more precise version of Theorem 8.4.4 can be found in Section 8.7 as Theorem 8.7.5. This result is analogous to the result of [AG19], which rules out a gap amplification procedure that preserves stoquasticity under the assumption that $\mathsf{MA} \neq \mathsf{NP}$.[8] Moreover, we point out that many Hamiltonian gadget constructions *do* satisfy such fidelity-preserving conditions, and indeed are precisely those

---

[8]Or taking a different view, proving the existence of such gap amplifications would allow one to simultaneously prove that $\mathsf{MA}$ can be derandomized (or even $\mathsf{RP}$ if it exhibits some additional properties) [AG19].

that were used in Chapter 7 to improve the hardness results for the guided local Hamiltonian problem.[9]

Second, we can use our results to formulate a stronger version of the NLTS theorem (and an alternative to the NLSS conjecture [GL22]), which we will call the *No Low-energy Classically-Evaluatable States conjecture*. This conjecture can hopefully provide a new stepping stone towards proving the quantum PCP conjecture.

**8.4.5.** CONJECTURE. *(Informal; NLCES conjecture) There exists a family of local Hamiltonians $\{H_n\}_{n\in\mathbb{N}}$ on $n$ qubits, and a constant $\beta > 0$, such that for sufficiently large $n$ for every classically evaluatable state $u \in \mathbb{C}^{2^n}$ as per Definition 8.2.2, we have that*

$$\langle u | H_n | u \rangle \geq \lambda_0(H_n) + \beta \,.$$

Again, a more precise statement can be found in Section 8.7 as Conjecture 8.7.6. Just as is the case for the NLSS conjecture and the NLTS theorem, the NLCES conjecture would, if proven to be true, not necessarily imply the quantum PCP conjecture. For example, it might be that there exist states that can be efficiently described classically but for which computing expectation values is hard (just as, for example, tensor network contraction is #P-hard in the worst case [Sch+07; BMT15]). Furthermore, as we have shown in this work, states with high energy but also a large fidelity with the ground state suffice as witnesses to decision problems on Hamiltonian energies, and these would not be excluded by a proof of the NLCES conjecture above. To make this more concrete, we also formulate an even stronger version of the NLCES conjecture, which states that there must be a family of Hamiltonians, for which no classically evaluatable state has good fidelity with the low energy spectrum (Conjecture 8.7.7).

## 8.5 QCMA-completeness of guidable local Hamiltonian problems

In this section we prove that Guidable Local Hamiltonian problems are QCMA-hard in the inverse polynomial precision regime. Our construction is based on

---

[9]For a quantum version of gap amplification, one would typically expect locality-reducing Hamiltonian gadgets as part of the procedure, to compensate for a "powering step" which consists of taking powers of the Hamiltonian (which therefore increases locality). It is already known that the current best-known locality-reducing gadgets [Bra+08] cannot be used because they increase the norm of the Hamiltonian by a constant factor, which results in an unmanageable decrease of the relative promise gap. Our result shows that even if one would find better constructions that don't have this effect, they would still have to satisfy the additional constraints as described in Theorem 8.7.5.

a combination of the ideas needed to show BQP-hardness for the Guided Local Hamiltonian problem 7 and the small penalty clock construction of [DGF22].

The first obstruction one encounters in adopting the ideas from the BQP-hardness proofs of the Guided Local Hamiltonian problem to the guidable setting is the fact that QCMA verifiers, unlike BQP, have a proof register. In QCMA the promises of completeness and soundness are always with respect to computational basis state witnesses. Hence, these might no longer hold when *any* quantum state can be considered as witness: for example, in the NO-case there might be highly entangled states which are accepted with probability $\geq 2/3$. When considering a circuit problem, the verifier can easily work around this by simply measuring the witness and then proceeding to verify with the resulting computational basis state. However, there is also another trick, which retains the unitarity of the verification circuit – and which we will denote as the 'CNOT-trick' from now on – to force the witness to be classical, first used in proving QCMA-completeness of the *Low complexity low energy states* problem in [WJB03]. Since the authors do not explain the precise mechanism behind the workings of this CNOT-trick, we provide a short proof of the lemma below.

**8.5.1.** LEMMA (The 'CNOT-trick'). *Let $p(n) : \mathbb{N} \to \mathbb{R}_{>0}, q(n) : \mathbb{N} \to \mathbb{R}_{>0}$ be polynomials. Let $U_n$ be a quantum polynomial-time verifier circuit that acts on an $n$-qubit input register $A$, a $p(n)$-qubit witness register $B$ and a $q(n)$-qubit workspace register $C$, initialized to $|0\rangle^{\otimes q(n)}$. Denote $\Pi_0$ for the projection on the first qubit being zero. Let $Q$ be the Marriott-Watrous operator of the circuit, defined as*

$$Q = \left( \langle x| \otimes I_w \otimes \langle 0|^{\otimes q(n)} \right) U_n^\dagger \Pi_0 U_n \left( |x\rangle \otimes I_w \otimes |0\rangle^{\otimes q(n)} \right). \tag{8.1}$$

*Consider yet another additional $p(n)$-qubit workspace $D$ initialized to $|0\rangle^{\otimes p(n)}$, on which $U_n$ does not act. Then by prepending $U_n$ with $p(n)$ CNOT-operations, each of which is controlled by a single qubit in register $B$ and targeting the corresponding qubit in register $D$, the corresponding Marriott-Watrous operator becomes diagonal in the computational basis.*

**Proof:**
Denote $U_{\text{CNOT}}$ for the $2p(n)$ qubit operation that acts on the two registers $B$ and $D$, and that for each $l \in [p(n)]$ applies a CNOT controlled by qubit $l$ in register $B$ and targets qubit $l$ in register $D$. Consider the new verifier circuit $\tilde{U}_n = U_n U_{\text{CNOT}}$ that acts on the registers $A, B, C$ and $D$, with the corresponding Mariott-Watrous operator $\tilde{Q}$. Let $|i\rangle$ and $|j\rangle$ for $i, j \in [2^{p(n)}]$ be arbitrary computational basis

states. Then we have

$$
\begin{aligned}
\langle i| \tilde{Q} |j\rangle &= \Big( \langle x| \otimes \langle i| \otimes \langle 0|^{\otimes q(n)} \otimes \langle 0|^{\otimes p(n)} \Big) U_{\mathrm{CNOT}} U_n^\dagger \Pi_0 \\
&\qquad U_n U_{\mathrm{CNOT}} \Big( |x\rangle \otimes |j\rangle \otimes |0\rangle^{\otimes q(n)} \otimes |0\rangle^{\otimes p(n)} \Big) \\
&= \Big( \langle x| \otimes \langle i| \otimes \langle 0|^{\otimes q(n)} \otimes \langle i| \Big) U_n^\dagger \Pi_0 U_n \Big( |x\rangle \otimes |j\rangle \otimes |0\rangle^{\otimes q(n)} \otimes |j\rangle \Big) \\
&= \langle i| |j\rangle \Big( \langle x| \otimes \langle i| \otimes \langle 0|^{\otimes q(n)} \Big) U_n^\dagger \Pi_0 U_n \Big( |x\rangle \otimes |j\rangle \otimes |0\rangle^{\otimes q(n)} \Big) \\
&= \delta_{i,j} \langle i| Q |j\rangle,
\end{aligned}
$$

where we used the fact that $V$ and $\Pi_0$ themselves do not act on register $D$. Hence, the operator $\tilde{Q}$ is diagonal in the computational basis, where its entries are taken from the diagonal of $Q$.                    $\square$

The next obstruction one faces is that in the QCMA setting there might be multiple proofs which all have exponentially close, or even identical, acceptance probabilities. The analysis of the BQP-hardness proof fails to translate directly to this setting, and another technique is needed. For this, we resort to the *small-penalty clock construction* of [DGF22]. The key idea is to use a Feynman-Kiteav circuit-to-Hamiltonian mapping modified with a tunable parameter $\epsilon$, which maps a quantum verification circuit $U_n$, consisting of $T$ gates from a universal gate set of at most 2-local gates, taking input $x$ and a quantum proof $|\psi\rangle \in (\mathbb{C}^2)^{\otimes poly(n)}$ to a $k$-local Hamiltonian of the form

$$
H_{FK}^x = H_{\mathrm{in}} + H_{\mathrm{clock}} + H_{\mathrm{prop}} + \epsilon H_{\mathrm{out}}. \tag{8.2}
$$

The value of $k$ depends on the used construction. Intuitively, the first three terms check that the Hamiltonian is faithful to the computation and the last term shifts the energy level depending on the acceptance probability of the circuit. Just as in [DGF22], we will use Kempe and Regev's 3-local construction. A precise description of the individual terms in (8.2) can be found in [KR03], and will not be relevant for our work, except for the fact that the $H_{\mathrm{FK}}^x$ has a polynomially bounded operator norm. The ground state of the first three terms $H_0 = H_{\mathrm{in}} + H_{\mathrm{clock}} + H_{\mathrm{prop}}$ is given by the so-called *history state*, which is given in [KR03] by

$$
|\psi_{\mathrm{hist}}(\phi)\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t \dots U_1 |\phi\rangle |0\rangle \big|\hat{t}\big\rangle, \tag{8.3}
$$

where $|\phi\rangle$ is the quantum proof and $\hat{t}$ the unary representation of the time step of the computation given by

$$
\hat{t} = |\underbrace{1\dots1}_{t}\underbrace{0\dots0}_{T-t}\rangle.
$$

From the construction in [KR03], it is easily verified that if $U_n$ accepts $(x, |\phi\rangle)$ with probability $p$ then we have that the corresponding history state has energy

$$\langle \psi_{\text{hist}}(\phi)| H_{FK}^x |\psi_{\text{hist}}(\phi)\rangle = \epsilon \frac{1-p}{T+1}. \tag{8.4}$$

Though the core idea behind the small-penalty clock construction is identical to the one used in the BQP-hardness proof – rescaling the weight of the $H_{\text{out}}$ term as compared to the other terms in a Feynman-Kitaev circuit-to-Hamiltonian mapping – the analysis differs: using tools from the Schrieffer-Wolff transformation one can find precise bounds on intervals in which the energies in the low-energy sector must lie, gaining fine control over the relation between the acceptance probabilities of the circuit and the low-energy sector of the Hamiltonian. The main lemma we use from [DGF22] is adopted from the proof of Lemma 26 in their work.

**8.5.2.** LEMMA (Small-penalty clock construction, adopted from Lemma 26 [DGF22]). *Let $U_n$ be a quantum verification circuit for inputs $x$, $|x| = n$, where $U_n$ consists of $T = poly(n)$ gates from some universal gate-set using at most 2-local gates. Denote $P(\psi)$ for the probability that $U_n$ accepts $(x, |\phi\rangle)$, and let $H_{FK}^x$ be the corresponding 3-local Hamiltonian from the circuit-to-Hamiltonian mapping in [KR03] with a $\epsilon$-factor in front of $H_{out}$, as in Eq. (8.2). Then for all $\epsilon \leq c/T^3$ for some constant $c > 0$, we have that low-energy subspace $\mathcal{S}_\epsilon$ of $H$, i.e.*

$$S_\epsilon = span\{|\Phi\rangle : \langle \Phi| H |\Phi\rangle \leq \epsilon\}$$

*has that its eigenvalues $\lambda_i$ satisfy*

$$\lambda_i \in \left[ \epsilon \frac{1 - P(\phi_i)}{T+1} - \mathcal{O}(T^3 \epsilon^2), \epsilon \frac{1 - P(\phi_i)}{T+1} + \mathcal{O}(T^3 \epsilon^2) \right], \tag{8.5}$$

*where $\{|\psi_i\rangle\}$ are the eigenstates of the Mariott-Watrous operator of the circuit $U_n$ given by Eq. (8.1).*

Having a QCMA-verifier with the CNOT-trick of Lemma 8.5.1 ensures that in Lemma 8.5.2 all $|\phi_i\rangle$ are computational basis states, as the CNOT-trick diagonalizes the Mariott-Watrous operator. The key idea is now to exploit the fact that QCMA, unlike for what is known for QMA, is a 'well-behaved' class in the sense that is equal to UQCMA (under randomized reductions).

**8.5.3.** THEOREM. CGaLH$(k, \delta, \zeta)$ *is* QCMA-*hard under randomized reductions for $k \geq 2$, $\zeta \in (1/poly(n), 1 - 1/poly(n))$ and $\delta = 1/poly(n)$.*

**Proof:**
Let us first state a 'basic version' reduction, which uses basis states as guiding states which trivially satisfy the conditions of Definition 8.2.2, for which we prove completeness and soundness, and finally improve its parameters in terms of the achievable fidelity and locality domains.
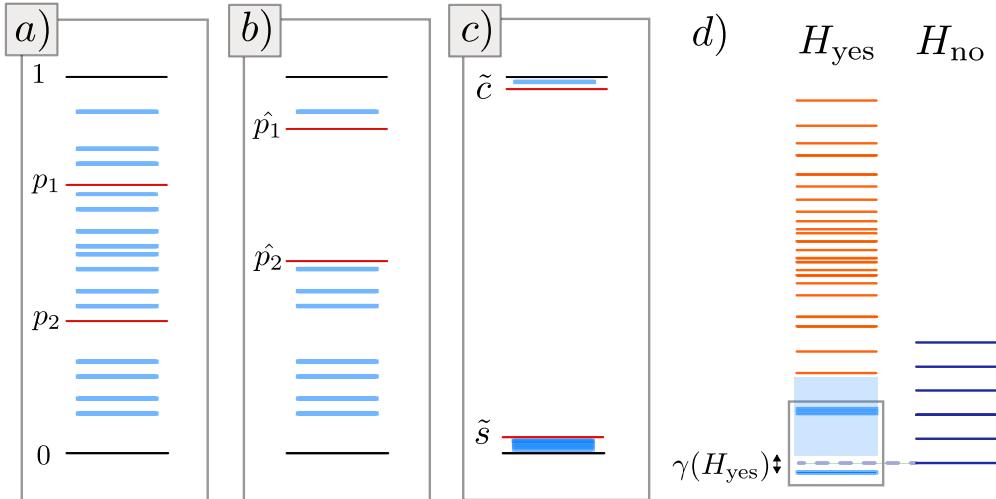
Figure 8.3: Illustration of the key ideas to construct the desired witness distribution in the YES-case in the first part of the reduction. The blue lines are witnesses, for which their position with respect to the $y$-axis represents the corresponding acceptance probabilities. The dark red lines represent the completeness and soundness parameters. $a) \to b)$ represents the randomized reduction from a QCMA-problem to a UQCMA one, $b) \to c)$ the error reduction and finally $d) \to e)$ The spectra of $H_{\mathrm{yes}}$ and $H_{\mathrm{no}}$ when $x \in A_{\mathrm{yes}}$. $H_{\mathrm{yes}}$ follows from the circuit-to-Hamiltonian mapping with the small penalty resulting in a Hamiltonian with fine control over its low-energy subspace, allowing one to ensure that its ground state is unique and can be made exponentially close to the history state corresponding to the unique accepting witness. The light blue shaded area represents the fact that we do not know the exact energy values corresponding to non-accepting witnesses, except for the fact that they are separated from $\lambda_0(H_{\mathrm{yes}})$ by at least $\gamma(H_{\mathrm{yes}}) = \Omega(1/\tilde{T}^6)$ for our choice of $\epsilon$. $H_{\mathrm{no}}$ is chosen such that its ground state energy lies exactly in the gap of $H_{\mathrm{yes}}$ in the sc Yes-case. Observe that if one was able to show that QMA $\subseteq_r$ UQMA, one could use the same proof construction to show QMA-hardness of inverse-poly-gapped Hamiltonians, for which we only yet know that they are QCMA-hard. QCMA-hardness for inverse-poly-gapped Hamiltonians was already shown in [Aha+22] (in fact they even show it for 1D Hamiltonians), and rediscovered in this work.

**The reduction** Let $\langle U_n, p_1, p_2 \rangle$ be a QCMA promise problem. By using Lemma 8.1.3, there exists randomized reduction to a UQCMA promise problem $\langle \hat{U}_n, \hat{p}_1, \hat{p}_2 \rangle$, $\hat{p}_1 - \hat{p}_2 \geq 1/q(n)$ for some polynomial $q$, which uses witnesses $y \in \{0,1\}^{p(n)}$ for some polynomial $p(n)$ and uses at most $T = poly(n)$ gates. We will now apply the following modifications to the UQCMA instance:

1. First, we force the witness to be classical by adding another register to which we 'copy' all bits of $y$ (through CNOT operations), before running the actual verification protocol – i.e. we use the CNOT trick of Lemma 8.5.1, which diagonalizes the corresponding Marriot-Watrous operator in the computational basis.

2. We apply *error reduction* to the circuit. This is done by applying the so-called "Marriot and Watrous trick" for error reduction, described in [MW04], which allows one to repeat the verification circuit several times whilst re-using the same witness. It is shown in [MW04], Theorem 3.3, that for any quantum circuit $V_n$ using $T = poly(n)$ 2-qubit gates which decides on acceptance or rejection of an input $x$, $|x| = n$, using a $p(n)$-qubit witness $|\psi\rangle$ for some polynomial $p$, satisfying completeness and soundness probabilities $c$, $s$ such that $c - s \geq 1/q(n)$ there is another circuit $\tilde{V}_n$ that again uses a $p(n)$-qubit witness $|\psi\rangle$ but has completeness and soundness $1 - 2^{-r}$ and $2^{-r}$, respectively, at the cost of using $\tilde{T} = \mathcal{O}(q^2 rT)$ gates.

Let the resulting protocol be denoted by $\langle \tilde{U}_n, \tilde{c}, \tilde{s} \rangle$, where $\hat{U}_n$ has an input register $A$, a witness register $W$ and ancilla register $B$, uses $\tilde{T} = \mathcal{O}(q^2 rT)$ gates and has completeness and soundness $C = 1 - 2^{-r}$ and $\hat{s} = 2^{-r}$. We denote $y^*$ for the (unique) witness with acceptance probability $\geq C$ in the YES-case. We keep $r$ as a parameter to be tuned later in our construction. We will also write $P(y) := \Pr[\hat{U} \text{ accepts } (y)]$. Now consider the 4-local Hamiltonian

$$H^x = H_{yes} \otimes |0\rangle \langle 0|_D + H_{no} \otimes |1\rangle \langle 1|_D, \qquad (8.6)$$

where $H_{yes} = H_{FK}^x$ is the Hamiltonian given by Eq. (8.2) using the circuit $\hat{U}_n$ and parameter $\epsilon$ and $H_{no}$ is given by

$$H_{no} = \sum_{i=0}^{R-1} |1\rangle \langle 1| \, 1_i + bI, \qquad (8.7)$$

where $R$ is the total size of the registers $A$, $W$, $B$ and the clock register $C$, and $b > 0$ is yet another tunable parameter. Note that $H_{no}$ has a *unique* ground state with energy $b$ given by the all zeros state, and the spectrum after that increases in steps of 1 (and so it in particular has a *spectral gap* of 1). We also have that $\|H_{no}\| = R + b = poly(n)$. As a guiding state in the YES-case will use the following basis state

$$|u_{yes}\rangle = |x\rangle_A |y^*\rangle_W |0 \ldots 0\rangle_B |0\rangle_C |0\rangle_D, \qquad (8.8)$$

which satisfies $(\langle\psi_{\text{hist}}(y^*)|\langle 0|_D)|u_{\text{yes}}\rangle = 1/\sqrt{(T+1)} = \mathcal{O}(1/\text{poly}(N))$, with $|\psi_{\text{hist}}(y^*)\rangle$ being the history state of witness $y^*$ for Hamiltonian $H_{yes}$. In the NO-case, we will show that the state

$$|u_{\text{no}}\rangle = |0\dots 0\rangle_{AWBC}|1\rangle_D, \tag{8.9}$$

will be in fact the ground state. We will show that setting $b := \mathcal{O}(1/\tilde{T}^7)$ and $\epsilon := \mathcal{O}(1/\tilde{T}^5)$, our reduction achieves the desired result.

**Completeness**  Let us first analyse the YES-case. By Lemma 8.5.2, we have that the eigenvalue $\lambda(y)$ corresponding to the witness $y^*$ is upper bounded by

$$\lambda(y^*) \leq \epsilon\frac{2^{-r}}{\tilde{T}+1} + \mathcal{O}(\tilde{T}^3\epsilon^2).$$

On the other hand, we have that for any $y \neq y^*$

$$\lambda(y) \geq \epsilon\frac{1-2^{-r}}{\tilde{T}+1} - \mathcal{O}(\tilde{T}^3\epsilon^2) = \Omega\left(\frac{1}{\tilde{T}^6}\right)$$

for our choice of $\epsilon$ and $r \geq 1$. Hence, for our choice of $\epsilon$ we must have that the ground state $|E_0\rangle$ of $H_{\text{yes}}$ is unique and has a spectral gap that can be bounded as

$$\gamma(H_{yes}) \geq \epsilon\frac{1-2^{-r+1}}{\tilde{T}+1} - \mathcal{O}\left(\tilde{T}^3\epsilon^2\right) = \Omega\left(\frac{1}{\tilde{T}^6}\right), \tag{8.10}$$

for some $r \geq \Omega(1)$ (we will pick $r$ to be much larger later). Let us consider the fidelity of the history state $|\psi_{\text{hist}}(y^*)\rangle$ with the actual ground state. First, we have that the energy of $|\psi_{\text{hist}}(y^*)\rangle$ is upper bounded by

$$\langle\psi_{\text{hist}}(y^*)|H_{yes}|\psi_{\text{hist}}(y^*)\rangle \leq \epsilon\frac{2^{-r}}{\tilde{T}+1} = \mathcal{O}\left(\frac{2^{-r}}{\tilde{T}^6}\right),$$

which follows directly from Eq. 8.4 and the fact that $P(y^*) \geq 1 - 2^{-r}$. We can write $|\psi_{\text{hist}}(y^*)\rangle$ in the eigenbasis of $H_{yes}$ as

$$|\psi_{\text{hist}}(y^*)\rangle = \alpha|E_0\rangle + \sqrt{1-\alpha^2}\left|E_0^\perp\right\rangle,$$

for some real number $\alpha \in [0,1]$, where $|E_0\rangle$ is the actual ground state of $H_{yes}$ and $\left|E_0^\perp\right\rangle$ another state orthogonal to $|E_0\rangle$. We have that the energy of $|\psi_{\text{hist}}(y^*)\rangle$ is upper bounded by

$$\langle\psi_{\text{hist}}(y^*)|H_{yes}|\psi_{\text{hist}}(y^*)\rangle \leq \epsilon\frac{2^{-r}}{\tilde{T}+1} = \mathcal{O}\left(\frac{2^{-r}}{\tilde{T}^6}\right).$$

On the other hand, the energy of $|\eta(y^*)\rangle$ is lower bounded by

$$\langle\psi_{\text{hist}}(y^*)|\,H_{yes}\,|\psi_{\text{hist}}(y^*)\rangle = \alpha^2\,\langle E_0|\,H_{yes}\,|E_0\rangle + (1-\alpha^2)\langle E_0^\perp|H_{yes}|E_0^\perp\rangle \geq \Omega\left(\frac{1-\alpha^2}{\tilde{T}^6}\right),$$

using the fact that $H_{yes}$ is PSD. Combining the upper and lower bounds, we find

$$\alpha^2 = |\,\langle\psi_{\text{hist}}(y^*)|E_0\rangle\,|^2 \geq 1 - \mathcal{O}\left(2^{-r}\right), \tag{8.11}$$

which can be made $\geq 1 - 2^{-c\tilde{T}}$ for some $r = c\tilde{T} + \mathcal{O}(1)$. Hence, we have that the fidelity of $|u_{\text{yes}}\rangle$ with the unique ground state of $H$ can be lower bounded as

$$|\langle u_{\text{yes}}|E_0\rangle|^2 \geq 1 - \left(\sqrt{1 - |\,\langle u_{\text{yes}}|\,(|\psi_{\text{hist}}(y^*)\rangle\,|0\rangle)|^2} + \sqrt{1 - |(\langle\psi_{\text{hist}}(y^*)|\,\langle 0|)\,|E_0\rangle|^2}\right)^2$$

$$\geq 1 - \left(\sqrt{1 - \frac{1}{\tilde{T}+1}} + 2^{-c\tilde{T}/2}\right)^2$$

$$\geq \Omega\left(\frac{1}{\tilde{T}}\right),$$

as desired.

**Soundness**   We have that all witnesses $y$ get accepted by $\hat{U}$ with at most an exponentially small probability, and hence have that $H_{yes} \succeq \Omega(1/\tilde{T}^6)$. By our choice $b$ we have therefore ensured that the ground state in the NO-case must be the state given by Eq. (8.9), which has energy $b = \Omega(1/\tilde{T}^7)$. Hence, the promise gap between YES and NO cases is $\delta = \Omega(1/\tilde{T}^7) = \Omega(1/q^2T^8) = 1/\,poly(n)$.

We will use similar tricks as used to proof Theorem 7.4.1 to improve the basic construction in terms of the fidelity range and locality.

**Increasing the fidelity range**   Note that in the NO-case we already have that the ground state is a semi-classical poly-sized subset state. However, in the YES-case, the ground state is a history state with only inverse polynomial fidelity with the state $|u_{\text{yes}}\rangle$. To work around this, we apply the same trick as for Proposition 7.7.1: by pre-idling the circuit with a polynomial number of identities, of which we denote the total number by $N$, and guiding state to

$$\left|u_{\text{yes}}^{\text{new}}\right\rangle = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} |x\rangle_A\,|y^*\rangle_W\,|0\ldots0\rangle_B\,|t\rangle_C\,|0\rangle_D\,, \tag{8.12}$$

which satisfies

$$|\left\langle u_{\text{yes}}^{\text{new}}\right|\,(|\psi_{\text{hist}}(y^*)\rangle\,|0\rangle)|^2 = \frac{N}{N + \tilde{T} + 1}.$$

Since the history state itself has an exponentially close fidelity with the ground state by equation Eq. (8.11), we have that the guiding state itself has an inverse polynomially close to 1 fidelity with the unique ground state $|\psi\rangle$ of large enough $N$. For the new pre-idled circuit we have to replace in all our results throughout our construction $\tilde{T}$ by $\tilde{T} + N$ and we have that the fidelity becomes

$$\left|\left\langle u_{\text{yes}}^{\text{new}}\middle|E_0\right\rangle\right|^2 \geq 1 - \left(\sqrt{1 - |\left\langle u_{\text{yes}}^{\text{new}}\middle|(|\psi_{\text{hist}}(y^*)\rangle\,|0\rangle)|^2} + \sqrt{1 - |(\langle\psi_{\text{hist}}(y^*)|\,\langle 0|)\,|E_0\rangle|^2}\right)^2$$

$$\geq 1 - \left(1 - \frac{N}{N + \tilde{T} + 1} + 2^{-c(\tilde{T}+N)/2}\right)^2$$

$$\geq 1 - \frac{1}{r(n)},$$

for any positive polynomial $r$ for some choice of $N \in poly(\tilde{T})$.

**Classical evaluatability and quantum preparability** We will check each condition of Definition 8.2.2 for $\left|u_{\text{yes}}^{\text{new}}\right\rangle$. Condition (i) follows directly from the definition of polynomially-sized subset states. For condition (ii) we have that $\langle u|O|u\rangle = \frac{1}{|S|}\sum_{i,j\in S}\langle i|O|j\rangle$ can be computed efficiently for any $O$ for which we have query access to its matrix elements, since each $\langle i|O|j\rangle$ corresponds to a query to the element $O_{i,j}$. Hence, when $|S| = poly(n)$ this can be done efficiently for any $k$. Finally, for condition (iii), we have that such states can be trivially prepared using $poly(n)$ quantum gates by using a series of controlled rotations on each qubit at a time. For instance, a very simple application of the algorithm from Grover-Rudolph [GR02] would suffice.

**Reducing the locality** Finally, we show how to reduce the locality of the constructed Hamiltonian. Assume that we have already increased the fidelity as above, so that the number of gates in the circuit is now $M = \tilde{T} + N$. The spectral gap of $H$, denoted as $\gamma(H)$, can be lower bounded as

$$\gamma(H) \geq \min\left[\gamma(H|_{x\in A_{yes}}), \gamma(H|_{x\in A_{no}})\right]$$

$$= \min\left[\delta, \Omega\left(\frac{1}{M^6}\right) - \Omega\left(\frac{1}{M^7}\right)\right]$$

$$= \delta = \Omega\left(\frac{1}{M^7}\right) = 1/poly(n).$$

Since the ground state is unique and inverse-polynomially gapped (in both the YES- and NO-case), we can apply Lemma 8.1.5 to obtain a 2-local Hamiltonian $H'$ which $(\Delta, \eta, \epsilon)$-simulates $H$, where we can take $\Delta = 1/poly(n) \geq \gamma(H)$ sufficiently large, $\eta, \epsilon = 1/poly(n) \leq \delta \leq \gamma(H)$ sufficiently small to ensure that the ground energy remains below some $a'$ in a YES instance and above some $b'$ in a NO instance, such that $b' - a' = \delta = 1/poly(n)$ and so that

$\|\mathcal{E}_{\text{state}}(|g\rangle) - |g'\rangle\| \leq \eta + \mathcal{O}(\gamma^{-1}\epsilon)$, where $|g\rangle$ is $|u_{\text{yes}}\rangle$ in a YES instance or $|u_{\text{no}}\rangle$ in a NO instance, $|g'\rangle$ is the ground state of $H'$, and $\mathcal{E}_{\text{state}}(|g\rangle)$ is as in Lemma 8.1.5. That is, $H'$ approximates $H$ in the low energy spectrum (below $\Delta$) in a way that the eigenvalues are perturbed by at most some small inverse-polynomial, and where the ground state can be approximated by the old ground state, plus some semi-classical state added as a tensor product. Finally, note that we can obtain $\|H\| \leq 1$ by simply scaling down by some polynomial, as required by the problem definition. Note that this will not change any of the statements as all relevant parameters and coefficients are (inverse) polynomials in $n$, albeit of very large degree.

Finally, we can ensure $\|H\| \leq$ by scaling $H$ with an inverse polynomially large factor.                                                                          □

Since polynomially-sized subset states are also samplable (see [GL22]), our proof would also go through if one considers a variant of the guidable local Hamiltonian problem which considers samplable states as in Definition 8.2.1 instead. We have the following corollary.

**8.5.4.** COROLLARY. $\mathsf{CGaLH}^*(k, \delta, \zeta)$ *is* QCMA-*complete, where the hardness is under randomized reductions, for $k \geq 2$, $\zeta \in (1/\operatorname{poly}(n), 1 - 1/\operatorname{poly}(n))$ and $\delta = 1/\operatorname{poly}(n)$.*

**Proof:**
Hardness follows from Theorem 8.5.3. Containment follows trivially from the fact that the YES-and NO-cases can be distinguished by using $\operatorname{desc}(u)$ as a witness, and a verifier circuit that prepares the quantum state $|u\rangle$ (which can be done efficiently possible because of the extra condition on $u$) followed by quantum phase estimation to an accuracy strictly smaller than the promise gap $\delta$, see Theorem 2 in [CFW22].                                                      □

Now that we have established QCMA-completeness for $\mathsf{CGaLH}^*$, we get QCMA-completeness for $\mathsf{QGaLH}$ *for free* for the same range of parameter settings, as the latter is a generalization of the former (containing $\mathsf{CGaLH}^*$ as a special case), and containment holds by the same argument as used in the proof of Corollary 8.5.4. However, with just a little bit of more work we can see that QCMA-hardness for $\mathsf{QGaLH}$ actually persists for a larger range of parameter settings. For this, we will use the following lemma by [LT20].

**8.5.5.** LEMMA (Ground state preparation with a-priori ground energy bound [LT20]). *Suppose we have a Hamiltonian $H = \sum_k \lambda_k(H) |\phi_q\rangle \langle\phi_q|$, where $\lambda_k(H) \leq \lambda_{k+1}(H)$, given through its $(\alpha, m, 0)$-block-encoding $U_H$. That is, we have access to a $(n + m)$-qubit unitary operator $U$ such that that*

$$\alpha(\langle 0|^{\otimes m} \otimes I)U(|0\rangle^{\otimes m} \otimes I) = H.$$

*Also suppose we have an initial state $|\psi\rangle$ prepared by some circuit $U_{prep}$ with the promise that $|\langle\phi_0||\psi\rangle|^2 \geq \Gamma$, and that we have the following bounds on the*

*ground energy and spectral gap:* $\lambda_0(H) \le \mu - \Delta/2 < \mu + \Delta/2 \le \lambda_1(H)$, *for some* $\mu, \Delta \in \mathbb{R}$. *Then the ground state can be prepared to fidelity* $1 - \varepsilon$ *with probability* $1 - \nu$ *with the following costs:*

1. *Query complexity:* $\mathcal{O}\left(\frac{\alpha}{\Gamma\Delta} \left(\log\left(\frac{\alpha}{\Delta}\right) \log\left(\frac{1}{\Gamma}\right) \log\left(\frac{\log(\alpha/\Delta)}{\nu}\right) + \log\left(\frac{1}{\varepsilon}\right)\right)\right)$ *queries to* $U_H$ *and* $\mathcal{O}\left(\frac{1}{\Gamma}\log\left(\frac{\alpha}{\Delta}\right)\log\left(\frac{\alpha/\Delta}{\nu}\right)\right)$ *queries to* $U_{prep}$,

2. *Number of qubits:* $\mathcal{O}\left(n + m + \log(\frac{1}{\Gamma})\right)$,

3. *Other one- and two-qubit gates:* $\mathcal{O}\left(\frac{m\alpha}{\Gamma\Delta}\left(\log\left(\frac{\alpha}{\Delta}\right)\log\left(\frac{1}{\Gamma}\right)\log\left(\frac{\log(\alpha/\Delta)}{\nu}\right) + \log\left(\frac{1}{\varepsilon}\right)\right)\right)$

**8.5.6.** Theorem. $\mathsf{QGaLH}(k, \delta)$ *is* QCMA-*complete for* $k \ge 2$, $\delta = 1/\operatorname{poly}(n)$ *and* $\zeta \in (1/\operatorname{poly}(n), 1 - 1/\exp(n))$.

**Proof:**
This follows immediately from the proof of Theorem 8.5.3, where the used history states themselves can be prepared by a quantum circuit of at most $\operatorname{poly}(n)$ gates. In the locality reduction, we can only ensure that we remain inverse polynomially close to the original ground state. However, due to Lemma 8.5.5, this fidelity is enough to guarantee the existence of a quantum circuit, still polynomial in $n$, that produces a new quantum state which is inverse exponentially close to the actual ground state. Let $U_{\text{prep}}$ be the quantum circuit that creates $|u\rangle = U_{\text{prep}}|0\rangle$. Let us assume the worst case setting in our construction of Theorem 8.5.3, where we have that $\Gamma = \zeta = 1/\operatorname{poly}(n)$, $\alpha = \mathcal{O}(1)$, $\Delta = \gamma(H) = \Omega(1/M^7)$, and $\mu = \lambda_0(H_{no}) = \Omega(1/M^7)$. Let $m = \mathcal{O}(1)$. Since the inverse fidelity $\varepsilon$ appears only logarithmically in Lemma 8.5.5, we can prepare a state that is exponentially-close in fidelity with (exponentially) high probability $1 - \nu$ in

$$\mathcal{O}\left(\operatorname{poly}(n)M^7\left(\log\left(\operatorname{poly}(n)M^7\right)\log\left(\operatorname{poly}(n)\right)\log\left(\frac{\log(M^7)}{\nu}\right) + \log\left(\exp(n)\right)\right)\right)$$
$$= \tilde{\mathcal{O}}\left(\operatorname{poly}(n)\right)$$

queries to $U_H$ (the block encoding of $H$) and single qubit gates, as well as

$$\mathcal{O}\left(\operatorname{poly}(n)\log\left(M^7\right)\log\left(\frac{M^7}{\nu}\right)\right) = \tilde{\mathcal{O}}\left(\operatorname{poly}(n)\right)$$

queries to $U_{\text{prep}}$. $\qquad\square$

# 8.6 Classical containment via spectral amplification

To complement our quantum hardness results with classical containment results in certain parameter regimes, we will use a technique based on the dequantization of the quantum singular value transformation as described in [GL22]. Our algorithm differs conceptually from the one proposed in [GL22] in the following ways:

- We consider a different (and less restrictive) input model: whereas [GL22] considers access to states of the form of Definition 8.2.1, we use states that adhere to the requirements as in Definition 8.2.2.

- For our purposes, we only consider local Hamiltonians (which are Hermitian sparse matrices) and not arbitrary sparse complex matrices. This simplifies the algorithm in the sense that we can view functions on these Hamiltonians as acting on the *spectrum* instead of the *singular values*.

- We also simplify the algorithm by tailoring it exactly to ground state *decision* instead of *estimation* problems, which allows us to use a different function acting on $H$ as compared to [GL22] to solve the relevant problems.

Let us introduce and prove bounds on the complexity of the *spectral amplification* algorithm in the next subsection. In the subsequent subsection, we will utilize this algorithm to put classical complexity upper bounds on $\mathsf{CGaLH}(k, \delta, \zeta)$ in specific parameter regimes.

## 8.6.1 Spectral amplification

Let $H = \sum_{i=0}^{m-1} H_i$ be a Hamiltonian on $n$ qubits which is a sum of $k$-local terms $H_i$, which satisfies $\|H\| \leq 1$. Since $H$ is Hermitian, we can write $H$ as

$$H = \sum_{i=0}^{2^n-1} \lambda_i |\psi_i\rangle \langle\psi_i|,$$

where $\lambda_i \in [-1, 1]$ (by assumption on the operator norm) denotes the $i$'th eigenvalue of $H$ with corresponding eigenvector $|\psi_i\rangle$. Consider a polynomial $P \in \mathbb{R}[x]$ of degree $d$, and write

$$P(x) = a_0 + a_1 x + \cdots + a_d x^d.$$

The *polynomial spectral amplification* of $H$ for $P$ is then defined as

$$\begin{aligned}
P(H) &= a_0 I + a_1 H + \cdots + a_d H^d \\
&= a_0 I + a_1 \sum_{i=0}^{2^n-1} \lambda_i |\psi_i\rangle \langle\psi_i| + \cdots + a_d \sum_{i=0}^{2^n-1} \lambda_i^d |\psi_i\rangle \langle\psi_i| \\
&= \sum_{i=0}^{2^n-1} P(\lambda_i) |\psi_i\rangle \langle\psi_i|.
\end{aligned}$$

Now for $\alpha \in [-1, 1]$, denote

$$\Pi_\alpha = \sum_{\{i:\lambda_i \leq \alpha\}} |\psi_i\rangle \langle\psi_i| \tag{8.13}$$

for the projection on all eigenstates of $H$ which have eigenvalues at most $\alpha$, which we will call a *low-energy projector* of $H$. Note that for any $\alpha \geq \lambda_0$, we must have that $\Pi_{\mathrm{gs}}\Pi_\alpha = \Pi_\alpha\Pi_{\mathrm{gs}} = \Pi_{\mathrm{gs}}$. We can utilize such a projector to solve $\mathsf{CGaLH}(k, \delta, \zeta)$, simply by computing $\|\Pi_\alpha |u\rangle\|$ for $\alpha = a$ given a classically evaluatable state $u$. To see why this works, note that in the YES-case, for the witness $\mathrm{desc}(u)$ we have that $\|\Pi_a |u\rangle\| \geq \|\Pi_{\mathrm{gs}} |u\rangle\| \geq \sqrt{\zeta}$ and in the NO-case we have that $\|\Pi_a |v\rangle\| = 0$ for all states, which means that the two cases are separated by $\sqrt{\zeta}$. However, it is unlikely that an efficient description exists of $\Pi_a$, and even if it did, it would not be $k$-local and therefore $\|\Pi_a |u\rangle\|$ would not even be necessarily efficiently computable.

The idea is now to approximate this low-energy projector $\Pi_\alpha$ by a polynomial in $H$. To see this, note that $\Pi_\alpha$ can be written exactly as

$$\Pi_\alpha = \frac{1}{2}\left(1 - \mathrm{sgn}(H - \alpha I)\right),$$

where $\mathrm{sgn}(x)$ is the sign function, which for our purposes is defined on $\mathbb{R} :\to \mathbb{R}$ as

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{if } x \leq 0. \end{cases}$$

From [HC17] we can then use the polynomial approximation of the sign function, which can subsequently be shifted to obtain the desired approximate low-energy projector $\tilde{\Pi}_a$.

**8.6.1.** LEMMA (Polynomial approximation to the sign function, from [HC17]). *For all $\delta' > 0, \epsilon' \in (0, 1/2)$ there exists an efficiently computable odd polynomial $P \in \mathbb{R}[x]$ of degree $d = \mathcal{O}\left(\frac{\log(1/\epsilon')}{\delta'}\right)$, such that*

- *for all $x \in [-2, 2] : |P(x)| \leq 1$, and*

- *for all $x \in [-2, 2] \setminus (-\delta', \delta') : |P(x) - sgn(x)| \leq \epsilon'$.*

Lemma 8.6.1 is, for the required conditions, optimal in its parameters $\delta'$, and $\epsilon'$ [HC17]. Since Lemma 8.6.1 holds on the entire interval $[-2, 2]$, choosing any $\alpha \in [-1, 1]$ and scaling the $\text{sgn}(x)$ function with the factor $1/2$ will ensure that the error, as in the lemma, will be $\leq \epsilon/2$. Let $q_\alpha(x) : \mathbb{R} \to [0, 1]$ defined as $q_\alpha(x) = \frac{1}{2}(1 - \text{sgn}(x - \alpha))$ be this function, with polynomial approximation $Q_\alpha \in \mathbb{R}[x]$ of degree $d$. Note that $Q_\alpha$ can be written as a function of $P$ as $Q_\alpha(x) = \frac{1}{2}(1 - P(x - \alpha))$. We will write $\tilde{\Pi}_\alpha = Q_\alpha(H)$ for the corresponding polynomial approximation of the approximate low-energy ground state "projector". Note that $\tilde{\Pi}_\alpha$ is Hermitian (since $H$ is Hermitian), but that $\tilde{\Pi}_\alpha$ is no longer necessarily a projector and therefore $\tilde{\Pi}_\alpha^2 \neq \tilde{\Pi}_\alpha$. If we now replace $\Pi_\alpha$ in $\|\Pi_\alpha |u\rangle\|$ by $\tilde{\Pi}_\alpha$, we get $\|\tilde{\Pi}_\alpha |u\rangle\| = \sqrt{\langle u| \tilde{\Pi}_\alpha^\dagger \tilde{\Pi}_\alpha |u\rangle} = \sqrt{\langle u| \tilde{\Pi}_\alpha^2 |u\rangle} = \sqrt{\langle u| (Q_\alpha(H))^2 |u\rangle}$, which means that we have to evaluate up to degree $2d$ powers of $H$. The next lemma will give an upper bound on the number of expectation values that have to be computed when evaluating a polynomial of $H$ of degree $d$.

**8.6.2.** LEMMA. *Given access to a classically evaluatable state $u$, a Hamiltonian $H = \sum_{i=0}^{m-1} H_i$, where each $H_i$ acts on at most $k$ qubits non-trivially, and a polynomial $P[x]$ of degree $d$, there exists a classical algorithm that computes $\langle u| P(H) |u\rangle$ in $\mathcal{O}(m^d)$ computations of $\langle u| O_i |u\rangle$, where the observables $\{O_i\}$ are at most $kd$-local.*

**Proof:**
We have that

$$\langle u| P(H) |u\rangle = \langle u| \left(a_0 I + a_1 H + \cdots + a_d H^d\right) |u\rangle$$
$$= a_0 + a_1 \langle u| H |u\rangle + \cdots + a_d \langle u| H^d |u\rangle.$$

Let $l \in [d]$ be the different powers for which we have to compute $\langle u| H^l |u\rangle$. We have that for each $l$ that

$$H^l = \left(\sum_{i=0}^{m-1} H_i\right)^l$$

consists of at $m^l$ terms when fully expanded and is Hermitian. However, when the sum is expanded, not every summand is necessarily Hermitian and therefore a local observable. However, we do have that for every $kl$-local summand $Q_{j,l}$ there exists another summand $Q_{j',l}$ which has all the terms of $Q_j$ but in reverse order, unless the $Q_j$ contains only powers of a single term. Grouping those together, we can write

$$H^l = \sum_{j=1}^{m + \frac{m^l - m}{2}} \hat{Q}_{j,l},$$

where each $\hat{Q}_{j,l}$ is $2l$ local and has $\|\hat{Q}_{j,l}\| \leq 2$. We can now simply absorb the factor 2 into the coefficients $a_{l'}$, $l' \in \{0, 1, \ldots, d\}$, such that all $\hat{Q}_{j,l}$ satisfy $\|\hat{Q}_{j,l}\| \leq 1$. The total number of local observables that have to be computed is now equal to

$$\sum_{l=1}^{d} m + \frac{m^l - m}{2} = \frac{m(m^d + dm - d - 1)}{2(m-1)} = \mathcal{O}(m^d),$$

completing the proof. □

All that remains to show is that for constant promise gap $\delta$, using a good enough approximation $\tilde{\Pi}_\alpha$ with a suitable choice of $\alpha$, will ensure that we can still distinguish the two cases in the $\mathsf{CGaLH}(k, \delta, \zeta)$ problem in a polynomial (resp. quasi-polynomial number of computations in $m$ when $\zeta = \Omega(1)$ (resp. $\zeta = 1/\operatorname{poly}(n)$).

**8.6.3. THEOREM.** *Let $H = \sum_{i=0}^{m-1} H_i$ be some Hamiltonian, and $\operatorname{desc}(u)$ be a description of a classically evaluatable state $u \in \mathbb{C}^{2^n}$. Let $a, b \in [-1, 1]$ such that $b - a \geq \delta$, where $\delta > 0$ and let $\zeta \in (0, 1]$. Consider the following two cases of $H$, with the promise that either one holds:*

*(i) $H$ has an eigenvalue $\leq a$, and $\|\Pi_{gs} |u\rangle\|^2 \geq \zeta$ holds, or*

*(ii) all eigenvalues of $H$ are $\geq b$.*

*Then there exists a classical algorithm that is able to distinguish between cases (i) and (ii) using*

$$\mathcal{O}\left(m^{c\left(\log(1/\sqrt{\zeta}))/\delta\right)}\right)$$

*computations of local expectation values, for some constant $c > 0$.*

**Proof:**
Let $\tilde{\Pi}_\alpha := Q_\alpha(H)$, where $Q$ is a polynomial of degree $d$, be the approximate low-energy projector that approximates $\Pi_\alpha = \frac{1}{2}(1 - \operatorname{sgn}(H - (\alpha I)))$. We set $\alpha := \frac{a+b}{2}$, $\delta' := \delta/2$ and $\epsilon' = 1/10$. We propose the following algorithm:

1. Compute $\|\tilde{\Pi}_a |u\rangle\|$ using a polynomial of degree $2d$ where $d = \mathcal{O}(\log(1/\epsilon'))/\delta'$, for $\epsilon' := \frac{1}{10}\sqrt{\zeta}$ and $\delta' = \delta/2$.

2. If $\|\tilde{\Pi}_\alpha |u\rangle\| \geq \frac{9}{10}\sqrt{\zeta}$, output (i) and output (ii) else.

Clearly, by Lemma 8.6.2, we have that this can be done in at most

$$\mathcal{O}\left(m^{c\left(\log(1/\sqrt{\zeta}))/\delta\right)}\right)$$

computations of expectation values of local observables, for some constant $c$. Let us now prove the correctness of the algorithm. Note that we can write $\tilde{\Pi}_\alpha$ as

$$\tilde{\Pi}_\alpha = \sum_{i=0}^{2^n-1} Q(\lambda_i) \left|\psi_i\right\rangle \left\langle\psi_i\right|,$$

where we have that

$$\begin{cases} 1 - \sqrt{\zeta}/2 \leq Q(\lambda_i) \leq 1 & \text{if } \lambda_i \leq a, \\ 0 \leq Q(\lambda_i) \leq \zeta/2 & \text{if } \lambda_i \geq b, \\ 0 \leq Q(\lambda_i) \leq 1 & \text{else}, \end{cases}$$

by Lemma 8.6.1. Let us analyse both case (i) and (ii) separately.

(i) $H$ has an eigenvalue $\leq a$, and $\|\Pi_{\mathrm{gs}}\left|u\right\rangle\|^2 \geq \zeta$ holds:

$$\begin{aligned}
\|\tilde{\Pi}_\alpha \left|u\right\rangle\| &\geq \|\tilde{\Pi}_\alpha \Pi_{\mathrm{gs}} \left|u\right\rangle\| \\
&= \|\Pi_\alpha \Pi_{\mathrm{gs}} \left|u\right\rangle - (\Pi_\alpha - \tilde{\Pi}_\alpha)\Pi_{\mathrm{gs}} \left|u\right\rangle\| \\
&= \left\|\Pi_{\mathrm{gs}} \left|u\right\rangle - \left(\sum_{i:\lambda_i \leq \alpha} \left|\psi_i\right\rangle \left\langle\psi_i\right| - \sum_{i=0}^{2^n-1} Q(\lambda_i) \left|\psi_i\right\rangle \left\langle\psi_i\right|\right) \Pi_{\mathrm{gs}} \left|u\right\rangle\right\| \\
&= \left\|\Pi_{\mathrm{gs}} \left|u\right\rangle - \left(\sum_{i:\lambda_i \leq \alpha} (1 - Q(\lambda_i)) \left|\psi_i\right\rangle \left\langle\psi_i\right| - \sum_{i:\lambda_i > \alpha} Q(\lambda_i) \left|\psi_i\right\rangle \left\langle\psi_i\right|\right) \Pi_{\mathrm{gs}} \left|u\right\rangle\right\| \\
&\geq \left\|\Pi_{\mathrm{gs}} \left|u\right\rangle - \left(\sum_{i:\lambda_i \leq \alpha} \frac{1}{10} \left|\psi_i\right\rangle \left\langle\psi_i\right|\right) \Pi_{\mathrm{gs}} \left|u\right\rangle\right\| \\
&= \left\|\Pi_{\mathrm{gs}} \left|u\right\rangle - \frac{1}{10}\left(\sum_{i:\lambda_i \leq \alpha} \left|\psi_i\right\rangle \left\langle\psi_i\right|\right) \Pi_{\mathrm{gs}} \left|u\right\rangle\right\| \\
&= \left\|\Pi_{\mathrm{gs}} \left|u\right\rangle - \frac{1}{10}\Pi_\alpha \Pi_{\mathrm{gs}} \left|u\right\rangle\right\| \\
&= \left\|\Pi_{\mathrm{gs}} \left|u\right\rangle - \frac{1}{10}\Pi_{\mathrm{gs}} \left|u\right\rangle\right\| \\
&= (1 - \frac{1}{10})\|\Pi_{\mathrm{gs}} \left|u\right\rangle\| \\
&\geq \frac{9}{10}\sqrt{\zeta}.
\end{aligned}$$

(ii) all eigenvalues of $H$ are $\geq b$:
We must have that

$$\|\tilde{\Pi}_\alpha \left|u\right\rangle\| \leq \frac{1}{2}\sqrt{\zeta},$$

since $\lambda_i \geq b$ for all $i \in \{0, \ldots, 2^n - 1\}$.

Hence, we have that the promise gap between both cases is lower bounded by

$$\frac{9}{10}\sqrt{\zeta} - \frac{1}{2}\sqrt{\zeta} = \frac{2}{5}\sqrt{\zeta},$$

which is $1/poly(n)$ when $\zeta \geq 1/poly(n)$. $\qquad\square$

**8.6.4.** REMARK. It should be straightforward to adopt the same derivation as above to a more general setting by considering sparse matrices, a promise with respect to fidelity with the low-energy subspace (i.e. all states with energy $\leq \lambda_0 + \gamma$ for some small $\gamma$), as well as $\epsilon > 0$ for $\epsilon$-classically evaluatable states (see Definition 8.2.2). However, this would likely put constraints on $\gamma$ and $\epsilon$, where $\epsilon$ in principle has to scale inversely proportional to the number of local terms in the Hamiltonian.
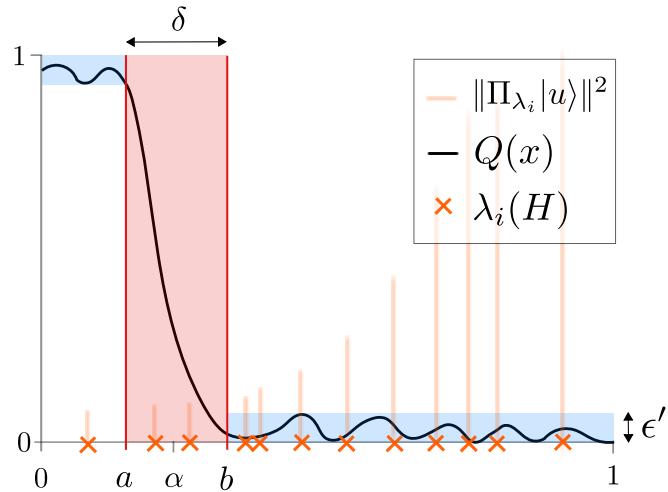


Figure 8.4: Illustration of the approximate low-energy projector $\Pi_\alpha$ in both the YES-case with $\alpha = \frac{a+b}{2}$. The orange crosses correspond to the energy values, and the attached shaded lines indicate the fidelity of the guiding state with the space spanned by all eigenstates $|\psi_l\rangle$ of $H$ that have energy at most $\lambda_i$. The polynomial approximation of the shifted sign function is displayed as $Q_\alpha(x)$, and the $\epsilon'$-error approximation regimes are indicated with the blue-shaded areas. In the red regime we do not have tight bounds on the error, except that the function values are in $[0, 1]$. For small enough $\epsilon$, in the YES-case the contribution of the ground state to the value of $\|\tilde{\Pi}_\alpha |u\rangle\|^2$ should be larger than that computed in the NO-case due to contributions of higher energy values, as a result from an inexact implementation of the low-energy projector. In the NO-case, all energy values will be larger than $b$.

### 8.6.2   Classical hardness and containment

All results in this section also hold when 'CGaLH' is replaced by 'CGaLH$^*$', as the containment trivially follows since CGaLH generalises CGaLH$^*$ and the hardness construction uses a diagonal (i.e. classical) Hamiltonian, of which the ground states are basis states and can thus be prepared on a quantum computer. To be able to make completeness statements when we consider NP, let us start by first proving a (straightforward) hardness result.

**8.6.5.** Lemma. $\mathsf{CGaLH}(k, \delta, \zeta)$ *is* NP-*hard for* $k \geq 2$, $\delta \leq \mathcal{O}(1)$ *and* $\zeta \leq 1$, *where* $k, \delta, \zeta$ *can also be functions of* $n$.

**Proof:**
We will prove this by a reduction from gapped 3-SAT. Let $\gamma$-3-SAT be a promise decision problem where we are given a formula $\phi(x) = \frac{1}{m} \sum_{i=0}^{m-1} C_i$ with $C_i = x_{i_1} \vee x_{i_2} \vee x_{i_3}$, with the promise that either $\phi(x) = 1$ (output YES) or $\phi(x) \leq \gamma$ (output NO), where $\gamma \in (0, 1)$. From (one of) the (equivalent) PCP theorem(s) we know that there exists a constant $\gamma \in (0, 1)$ for which deciding on the correct output (we are allowed to output anything if the promise doesn't hold) is NP-hard [Hås01]. Next, we apply the gadget from [AL21], which maps $\phi(x)$ to a 2-SAT instance with formula $\phi'(x) = \frac{1}{10m} \sum_{i=0}^{m-1} \sum_{j \in [10]} C'_{i,j}$. Here we have that $\phi'(x)$ has the property that for every clause $C_i$ there are 10 corresponding clauses $C'_{i,j}$, $j \in [10]$ such that, if a given assignment $x$ satisfies a clause $C_i$ of $\phi(x)$, then exactly 7 clauses of $C'_{i,j}$ can be satisfied, and for those $C_i$ that are not satisfied by $x$, at most 6 clauses of $C'_{i,j}$ are satisfied. Note that if $\phi(x) = 1$, we then must have that $\phi'(x) = 7/10$, and that if $\phi(x) \leq \gamma$, we have that $\phi'(x) \leq 7\gamma/10 + 6(1 - \gamma)/10 = (\gamma + 6)/10$. Hence, it is still NP-hard to distinguish between those cases, and the promise gap between the YES- and the NO-case is $\frac{1}{10}(1 - \gamma) =: \gamma'$, which is some constant. Let us now map $\phi'(x)$ into a 2-local diagonal Hamiltonian $H'$ such that $\langle x | H' | x \rangle = \phi(x)$, which can be done by a representation of the clauses as diagonal matrices. By our choice of $\phi'(x)$, we have already ensured that the Hamiltonian is (sub)-normalized. To turn the problem into a minimization problem, one can simply invert the spectrum by letting $H = I - H'$ (note that $H' \succeq 0$). The eigenvectors of $H$ are basis vectors – and thus themselves classically evaluatable states for which $\zeta = 1 = \mathcal{O}(1)$ – and its eigenvalues are precisely the function evaluations of $1 - \phi'(x)$. Hence, setting $a := 3/10$ and $b := (4 - \gamma)/10$ gives us $\delta = \gamma' = \mathcal{O}(1)$.                        □

Theorem 8.6.3 now gives us a very easy way to establish the following upper bounds on $\mathsf{CGaLH}(k, \delta, \zeta)$ when the required precision $\delta$ is only constant. Combined with Lemma 8.6.5, we obtain the following result, reminiscent of Theorem 5 in [GL22].

**8.6.6.** Theorem. $\mathsf{CGaLH}(k, \delta, \zeta)$ *is* NP-*complete for* $k = \mathcal{O}(\log(n))$, *and constants* $\delta \in (0, 1]$ *and* $\zeta \in (0, 1]$. *Furthermore, when* $\zeta = 1/poly(n)$ *we have that* $\mathsf{CGaLH}(k, \delta, \zeta)$ *is in* NqP.

**Proof:**

NP-Hardness follows from Lemma 8.6.5. The containment statements follow from Theorem 8.6.3, in which the proposed algorithm for $m = \mathcal{O}(n^k)$, $\delta \in (0,1]$ constant runs in polynomial time when $\zeta$ is constant and in quasi-polynomial time when $\zeta = 1/poly(n)$ (using the fact that $\mathcal{O}(n^{\log(n)}) = 2^{\mathcal{O}(\log^c(n))}$ for some constant $c > 0$). $\qquad\square$

Moreover, by a little more careful inspection one can show that the problem's hardness depends on how $\zeta$ and $\delta$ relate to one another, as shown in the following theorem.

**8.6.7.** THEOREM. *Let $f(n) : \mathbb{N} \to \mathbb{R}_{>0}$, $g(n) : \mathbb{N} \to \mathbb{R}_{>0}$ be some functions with the property that there exists some constant $n_0$ (which is known), such that for all $n \geq n_0$ we have that $1/g(n) - 1/f(n) > 0$. Then we have that $\mathsf{CGaLH}(k, \delta, \zeta)$ is NP-complete for $k \geq 2$, $\delta = 1/g(n)$ and $\zeta \geq 1 - 1/f(n)$.*

**Proof:**

*Hardness:* NP-hardness follows again trivially from Lemma 8.6.5, by taking $g(n) = \mathcal{O}(1)$ and letting $f(n)$ be some arbitrarily large function such that $f(n) \gg g(n)$, which gives a loose lower bound on $\zeta$ which can be as large as exactly 1.

*Containment:* To prove this we split the regime into the $n \geq n_0$ case and the $n < n_0$ case, giving separate algorithms for both cases.

$n < n_0$ **case:** In this setting we have that $n \in [1, n_0]$, which is a constant. Hence, we can simply diagonalize the full Hamiltonian and compute its ground state energy in time upper bounded by some constant.

$n \geq n_0$ **case:** As always, denote $\Pi_{\mathrm{gs}}$ for the projector on the ground space of $H$. The verifier expects to be given a $\mathrm{desc}(u)$ such that $\|\Pi_{\mathrm{gs}} |u\rangle\|^2 \geq 1 - 1/f(n)$ and checks if $\langle u| H |u\rangle \leq a + 1/f(n)$. Let us now check completeness and soundness of this simple protocol. By the definition of the problem, we must have that $\|H\| \leq 1$. We can write $|u\rangle = \alpha_1 |\phi_0\rangle + \alpha_2 \left|\phi_0^\perp\right\rangle$ for $|\alpha_1|^2 + |\alpha_2|^2 = 1$. Here $|\phi_0\rangle$ lives in the ground space of $H$ and $\left|\phi_0^\perp\right\rangle$ in the subspace orthogonal to the ground state. Note that $|\alpha_2|^2 = 1 - \|\Pi_{\mathrm{gs}} |u\rangle\|^2 \leq \frac{1}{f(n)}$. Therefore, we must have that in the YES-case

$$
\begin{aligned}
\langle u| H |u\rangle &= |\alpha_1|^2 \langle \phi_0| H |\phi_0\rangle + |\alpha_2|^2 \left\langle \phi_0^\perp \right| H \left|\phi_0^\perp\right\rangle \\
&\leq a + |\alpha_2|^2 \left\langle \phi_0^\perp \right| H \left|\phi_0^\perp\right\rangle \\
&\leq a + \|H\|/f(n) \\
&\leq a + 1/f(n).
\end{aligned}
$$

In the NO-case, we can simply evoke the variational principle

$$\langle u| H |u \rangle \geq \lambda_0(H) \geq b = a + 1/g(n) > a + 1/f(n),$$

for all $|u\rangle$, by the assumption of the functions for $n \geq n_0$. Therefore, the two cases are separated, and can therefore be distinguished from one another (using the fact that for our definition of classically evaluatable states the expectation of local observables can be computed exactly[10]). $\qquad\square$

## 8.7   Implications to the quantum PCP conjecture

In this final section, we consider some implications from all previous sections to the quantum PCP conjecture. We find that our results allow us to give a no-go result for the existence of quantum gap amplification procedures exhibiting certain properties (unless QCMA = NP or QCMA $\subseteq$ NqP), and our results allow us to pose a conjecture which generalizes NLTS (now a theorem [ABN22]) and provides an alternative to NLSS [GL22].[11]

### 8.7.1   Definition of QPCP

For completeness, we will start by giving the definition of quantum probabilistically checkable proof systems, or QPCP, following the definition of [Aha+09; AAV13]. Then we will state the QPCP conjecture. We will be mostly interested in the gap amplification variant of the QPCP conjecture, which we state at the end of this section.

**8.7.1.** DEFINITION (Quantum Probabilistically Checkable Proofs (QPCP)). Let $n \in \mathbb{N}$ be the input size and $p, q : \mathbb{N} \to \mathbb{N}$, $c, s : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with $c - s > 0$. A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ has a $(p(n), q(n), c, s)$-QPCP-verifier if there exists a quantum algorithm $V$ which acts on an input $|x\rangle$ and a polynomial number of ancilla qubits, and takes as additional input a quantum state $|\xi\rangle \in (\mathbb{C}^2)^{\otimes p(n)}$, from which it is allowed to access at most $q(n)$ qubits, followed by a measurement of the first qubit after which it accepts only if the outcome is $|1\rangle$, such that

> **Completeness.** If $x \in A_{\text{yes}}$, then there is a quantum state $|\xi\rangle$ such that the verifier accepts with probability at least $c$,

---

[10] In practice, one would want the difference to be large enough to be able to detect them with machine precision.

[11] See also [Cob+23], which proposes a closely-related conjecture independently of this work.

**Soundness.** If $x \in A_{\mathrm{no}}$, then for all quantum states $|\xi\rangle$ the verifier accepts with probability at most $s$.

A promise problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ belongs to $\mathsf{QPCP}[p, q, c, s]$ if it has a $(p(n), q(n), c, s)$-$\mathsf{QPCP}$ verifier. If $p(n) \leq poly(n)$, $c = 2/3$, and $s = 1/3$, we simply write $\mathsf{QPCP}[q]$.

Using the above notion quantum probabilistically checkable proofs (QPCPs) one can formulate the *quantum* PCP conjecture.

**8.7.2.** CONJECTURE (QPCP conjecture - proof verification version). *There exists a constant $q \in \mathbb{N}$ such that*

$$\mathsf{QMA} = \mathsf{QPCP}[q].$$

The celebrated PCP theorem states that any $\mathsf{NP}$ problem can be reformulated such that a probabilistically checkable proof system, can verify the problem by only checking a constant number of bits from a polynomially sized witness [Aro+98; AS98]. It also implies that there exists a constant $\alpha$ such that it is $\mathsf{NP}$-hard to decide for a constraint satisfaction problem (CSP) with 'promise gap' $\alpha$. Dinur [Din07] showed that this implication can be obtained directly, by reducing from a CSP with inverse polynomial promise gap to one with a constant promise gap, whilst retaining $\mathsf{NP}$-hardness. This type of reduction is commonly referred to as *gap amplification*.

As in the classical PCP theorem, there exist a gap amplification formulations of the QPCP conjecture. In the context of the quantum complexity classes (notably $\mathsf{QMA}$), 'quantum' CSPs are generalized by local Hamiltonian problems. The formulation of the quantum PCP conjecture in terms of inapproximability of local Hamiltonians is:

**8.7.3.** CONJECTURE (QPCP conjecture - gap amplification version). *There exists a (quantum) reduction from the local Hamiltonian problem with promise gap $1/poly(n)$ to another instance of the local Hamiltonian problem with promise gap $\Omega(1)$.*

It is well known that, at least under *quantum reductions*, both conjectures are in fact equivalent:

**8.7.4.** FACT ([Aha+09]). Conjecture 8.7.2 holds if and only if conjecture 8.7.3 holds.

## 8.7.2   Gap amplifications

We now consider the implications of Section 8.6 to gap amplifications of guidable local Hamiltonian problems.

**8.7.5.** THEOREM (No-go's for quantum-classical gap amplification). *There cannot exist*

1. *A polynomial time classical reduction from an instance of* $\mathsf{CGaLH}(k, \zeta, \delta)$ *with* $k \geq 2$, *some constant* $\zeta > 0$, *and* $\delta = 1/poly(n)$ *to some* $\mathsf{CGaLH}(k', \zeta', \delta')$ *with* $k' \geq 2$, *some constant* $\zeta' > 0$, *and* $\delta' = \Omega(1)$,

*unless* $\mathsf{QCMA} = \mathsf{NP}$, *and*

2. *A quasi-polynomial time classical reduction from an instance of* $\mathsf{CGaLH}(k, \zeta, \delta)$ *with* $k \geq 2$, $\zeta = 1/poly(n)$, *and* $\delta = 1/poly(n)$ *to some* $\mathsf{CGaLH}(k', \zeta', \delta')$ *with* $k' \geq 2$, $\zeta' = 1/poly(n)$, *and* $\delta' = \Omega(1)$,

*unless* $\mathsf{QCMA} \subseteq \mathsf{NqP}$.

**Proof:**
These all follow directly from Theorem 8.6.6.                                    □

One can also interpret the no-go results for gap amplifications (points 1 and 2 in the above theorem) in a more general setting: if one wants to prove the QPCP conjecture through a gap amplification procedure a la Dinur, the procedure needs to have the property that it doesn't preserve 'classically evaluatable' properties of eigenstates (it cannot even maintain an inverse polynomial fidelity with such states) unless at the same time showing that $\mathsf{QCMA} = \mathsf{NP}$ (or $\mathsf{QCMA} \subseteq \mathsf{NqP}$, which is also very unlikely)! Hence, this result can be viewed as a 'QCMA-analogy' to the result from [AG19], where the authors showed that the existence of quantum gap amplifications that preserve stoquasticity of Hamiltonians would imply that $\mathsf{NP} = \mathsf{MA}$. We also point out that it is possible that – even though the complexity of $\mathsf{QGaLH}$ and $\mathsf{CGaLH}^*$ was in the *inverse polynomial* precision regime the same when $1/poly(n) \leq \zeta \leq 1 - 1/poly(n)$ – it might very well be that their complexities will differ when considering a *constant* precision, as our containment results of Section 8.6 crucially use the properties of classically evaluatable states.

## 8.7.3   Classically evaluatable states and QPCP

Finally, we close by formulating a new conjecture which can be viewed as a strengthening of the NLTS theorem, or as an alternative to the NLSS conjecture of [GL22] in light of our results, and which must hold if the quantum PCP conjecture is true and $\mathsf{QMA} \neq \mathsf{NP}$.

**8.7.6.** CONJECTURE (No Low-Energy Classically-Evaluatable States (NLCES )). *There exists a family of local Hamiltonians $\{H_n\}_{n\in\mathbb{N}}$, where each $H_n$ acts on $n$ qubits, and a constant $\beta > 0$, such that for sufficiently large $n$ we have that for all classically evaluatable states $|u\rangle \in \mathbb{C}^{2^n}$ as in Definition 8.2.2 it holds that $\langle u| H |u\rangle \geq \lambda_0(H_n) + \beta$.*

Taking into account our results about the containment of the constant-gapped classically guidable local Hamiltonian problem in NP – namely the insight that what really matters is the *fidelity* of a classically evaluatable state with the low-lying energy subspace of the Hamiltonian, and not the energy of the classically evaluatable state itself – we can also define a stronger version of the NLCES conjecture, which must hold if the quantum PCP conjecture holds.

**8.7.7.** CONJECTURE (Strong-NLCES conjecture). *There exists a family of local Hamiltonians $\{H_n\}_{n\in\mathbb{N}}$, where each $H_n$ acts on $n$ qubits, and a constant $\beta > 0$, such that for sufficiently large $n$ we have that for all classically evaluatable states $|u\rangle \in \mathbb{C}^{2^n}$, as in Definition 8.2.2, we have that $\|\Pi_{\lambda_0(H_n)+\beta} |u\rangle \|^2 = o(1/\operatorname{poly}(n))$. Here $\Pi_{\lambda_0(H_n)+\beta}$ is the projector onto the space spanned by eigenvectors of $H$ with energy less than $\lambda_0(H_n) + \beta$.*

Note that the NLCES Conjecture is strictly weaker than the Strong-NLCES conjecture, and that both do not necessarily imply the QPCP conjecture.

## 8.7.4 Open questions and future work

**The (strong) NLCES conjecture.** It would be interesting to see whether the family of Hamiltonians used to prove the well-known NLTS conjecture, or constructions inspired by the proof thereof (in particular Hamiltonians that arise from error-correcting codes), can also be used to prove (weaker versions of) our NLCES conjecture (see Conjecture 8.7.6). Note that our NLCES conjecture is strictly stronger than NLTS, since it includes all states that can be prepared by constant depth quantum circuits (i.e. those states covered by the NLTS conjecture), but also includes states that require super-constant quantum depth, for example arbitrary Clifford circuits[12], matrix-product states, etc.

**The classical guiding state existence assumption.** As discussed in [Cad+23a], the existence of practical quantum advantage based on the previously mentioned two-step procedure is only expected if there exist guiding states, quantum or classical, that have not too much (exponentially close) but also not too little (exponentially small) fidelity with the ground space of the Hamiltonian under study. Whilst there is some literature that (partially) explores this direction [Bur+21; Tub+18; Lee+23], it would be useful and interesting to study this

---

[12]This has in fact recently been proven for Clifford circuits, see [Cob+23].

assumption in the special case of Ansätze that describe classically evaluatable and quantumly preparable states. This could provide numerical evidence to support the results that we have shown from a complexity-theoretic perspective: that classical heuristics combined with quantum phase estimation is indeed the right way to approach fault-tolerant quantum advantage in chemistry.

# 8.A    Perfect sampling access of MPS and stabilizer states

In this appendix we show that both matrix product states (MPS) and stabilizer states are samplable states, by checking all three conditions of Definition 8.2.1.

**Matrix product states:**    Let $u$ be a $N = 2^n$-dimensional vector described by an MPS of $n$ particles, bounded bond dimension $D$ and local particle dimension $d$.

(i) Let $\hat{i}$ be the bit representation of $i$. The algorithm $Q_u$ can simply be the evaluation of $\mathrm{Tr}[A_1^{(s_1)} A_2^{(s_2)} \ldots A_n^{(s_n)}]$ for $s = \hat{i}$, which can be done via a naive matrix multiplication algorithm in time $\mathcal{O}(nD^3)$, and thus clearly runs in time $\mathcal{O}\left(poly(\log(N))\right)$ when $d = \mathcal{O}(poly(n)$, $D = \mathcal{O}(poly(n)$.

(ii) We will use that expectation values that are a tensor product of 1-local observables can be computed efficiently for a MPS im time $\mathcal{O}(nd^2D^3)$ [VMC08]. We assume that $m$ is already known (see item (iii)), and that our MPS is therefore normalized. The algorithm $\mathcal{SQ}_u$ works as follows: one computes the probability that the first qubit is 1 by computing the expectation value of the 1-local projector $\Pi_1 = |1\rangle \langle 1|_1$. Let $p_1 = \langle u| \Pi_1 |u\rangle$ and $p_0 = 1 - p_1$. The algorithm now samples a bit $j_1 \in \{0,1\}$ according to distribution $\{p_0, p_1\}$, and computes the expectation value of the 2-local projector $\Pi_{j_1 1}$ to obtain $p_{j_1,1}$ and $p_{j_1,0}$, from which again a bit is sampled according to the distribution $p_{j_1,0}, p_{j_1,1}$. This procedure is repeated for all $n - 2$ remaining sites, which yields a sample $j$ with probability $|u_j|^2$. The total time complexity of this procedure is $\mathcal{O}(n^2d^2D^3) = \mathcal{O}(poly(\log N))$, when $d = \mathcal{O}(poly(n))$ and $D = \mathcal{O}(poly(n))$, as desired.

(iii) $m$ can easily be computed by considering the overlap of the MPS with itself, which can be done in time $\mathcal{O}(npD^3)$ as the overlap can be viewed as the expectation value of a 0-local observable.

**Stabilizer states:**    Let $u \in \mathbb{C}^{2^n}$, $N = 2^n$, be a stabilizer state on $n$ qubits.

(i) This follows from the fact that basis states are stabilizer states, and that there exists an algorithm $\mathcal{Q}_u$ that computes inner products between stabilizer states in time $\mathcal{O}(n^3) = \mathcal{O}(poly(\log N))$ [AG04].

(ii) This follows from the fact that stabilizer states can be strongly simulated (i.e. marginals can be computed), which allows for weak simulation as shown in [TD04] at overhead $n$ for the cost of strong simulation. Using the strong simuation algorithm as in [AG04], this gives an algorithm $\mathcal{SQ}_u$ that runs in time $\mathcal{O}(n^3) = \mathcal{O}(poly(\log N))$.

(iii) $m = 1$ by definition.

## 8.B MPS to circuit construction

In this section, we show that any MPS on $n$ qubits with bond dimension $D$ can be implemented on a quantum computer up to distance $\epsilon$, with respect to the 2-norm, in $\mathcal{O}(nD \log(D)^2 \log(Dn/\epsilon))$ one- and two-qubit gates and a $\mathcal{O}(npoly(D))$-time classical pre-calculation. The result is based on a result from [Sch+05]. For completeness we will first repeat their result. Let $\mathcal{H}_A = \mathbb{C}^D$ and $\mathcal{H}^B = \mathbb{C}^2$ be the Hilbert spaces characterising a $D$-dimensional ancillary system and a single qubit, respectively. Then every MPS of the form

$$|\psi\rangle = \langle \phi_F | V_n \dots V_1 |\phi_I\rangle$$

with arbitrary maps $V_k : \mathcal{H}_A \mapsto \mathcal{H}_A \otimes \mathcal{H}_B$, and $|\phi_I\rangle, |\phi_F\rangle \in \mathcal{H}_A$ is equivalent to a state

$$|\psi\rangle = \langle \tilde{\phi}_F | \tilde{V}_n \dots \tilde{V}_1 |\tilde{\phi}_I\rangle$$

with $\tilde{V}_k : \mathcal{H}_A \mapsto \mathcal{H}_A \otimes \mathcal{H}_B$ isometries and such that the ancillary register decouples in the last step

$$\tilde{V}_n \dots \tilde{V}_1 |\tilde{\phi}_I\rangle = |\phi_F\rangle \otimes |\psi\rangle.$$

Note that this is the canonical form of the MPS and can be found using $\mathcal{O}(npoly(D))$ classical pre-calculation time. The isometries are of size $2D \times D$ acting on the auxiliary system sequentially and create one qubit each. Every $\tilde{V}_k$ can be embedded into a unitary $U_k : \mathcal{H}_A \otimes \mathcal{H}_B \mapsto \mathcal{H}_A \otimes \mathcal{H}_B$ of size $2D \times 2D$, acting on the auxiliary system and a qubit initialised in $|0\rangle$ such that $U_k |\tilde{\phi}_k\rangle \otimes |0\rangle = \tilde{V}_k |\tilde{\phi}_k\rangle$. This gives the quantum circuit

$$U_n \dots U_1 |\tilde{\phi}_I\rangle |0\rangle^{\otimes n} = |\tilde{\phi}_F\rangle |\psi\rangle.$$

$|\tilde{\phi}_I\rangle$ is a state in $\mathcal{H}_A$ which can be generated on $\lceil \log(D) \rceil$ qubits, up to normalisation. By the Solovay-Kitaev theorem, this state can be prepared up to distance $\epsilon$ by a circuit of

$$\mathcal{O}(\lceil \log(D) \rceil^2 D \log(\lceil \log(D) \rceil^2 D/\epsilon)) = \mathcal{O}(D \log(D)^2 \log(1/\epsilon))$$

two and one-qubit gates. The unitaries $U_k$ act on $\lceil \log(D) \rceil + 1$ qubits hence they can be approximated up to error $\epsilon$ in

$$\mathcal{O}((\lceil \log(D) \rceil + 1)^2 (D+1) \log((\lceil \log(D) \rceil + 1)^2 (D+1)/\epsilon)) = \mathcal{O}(D \log(D)^2 \log(D/\epsilon)).$$

Note that because every unitary incurs an error $\epsilon$ the entire error can be bounded by $n\epsilon$, setting individual error to $\epsilon' = \frac{\epsilon}{n}$ ensures that the generated state is at most $\epsilon$ far from the desired state. This results in a circuit of complexity: $\mathcal{O}(nD\log(D)^2\log(nD/\epsilon))$ generating the MPS up to normalisation.

# Bibliography

[Aar09]     S. Aaronson. "Computational complexity: Why quantum chemistry is hard". In: *Nature Physics* 5 (Oct. 2009), pp. 707–708. DOI: 10.1038/nphys1415.

[AAS20]     S. Aaronson, Y. Atia, and L. Susskind. "On the Hardness of Detecting Macroscopic Superpositions". In: *arXiv* (2020). DOI: 10.48550/arXiv.2009.07450.

[AAV13]     D. Aharonov, I. Arad, and T. Vidick. "Guest column: the quantum PCP conjecture". In: *SIGACT News* 44.2 (June 2013). arXiv:1309.7495, pp. 47–79. DOI: 10.1145/2491533.2491549.

[AB09]      S. Arora and B. Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[AB99]      D. Aharonov and M. Ben-Or. "Fault-Tolerant Quantum Computation With Constant Error Rate". In: *arXiv* (1999). DOI: 10.48550/arXiv.quant-ph/9906129.

[ABN22]     A. Anshu, N. P. Breuckmann, and C. Nirkhe. "NLTS Hamiltonians from good quantum codes". In: *arXiv preprint* (June 2022). arXiv:2206.13228.

[Ach+25]    R. Acharya et al. "Quantum error correction below the surface code threshold". In: *Nature* 638.8052 (2025), pp. 920–926. DOI: 10.1038/s41586-024-08449-y.

[AG04]      S. Aaronson and D. Gottesman. "Improved simulation of stabilizer circuits". In: *Physical Review A* 70.5 (2004). arXiv:quant-ph/0406196, p. 052328. DOI: 10.1103/PhysRevA.70.052328.

[AG19]      D. Aharonov and A. B. Grilo. "Stoquastic PCP vs. Random-ness". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. `arXiv:1901.05270`. 2019, pp. 1000–1023. DOI: `10.1109/FOCS.2019.00065`.

[Aha+09]    D. Aharonov et al. "The detectability lemma and quantum gap amplification". In: *ACM Symposium on Theory of Computing (STOC)*. `arXiv:0811.3412`. New York, NY, USA, 2009, pp. 417–426. DOI: `10.1145/1536414.1536472`.

[Aha+22]    D. Aharonov et al. "The Pursuit of Uniqueness: Extending Valiant-Vazirani Theorem to the Probabilistic and Quantum Settings". In: *Quantum* 6 (Mar. 2022). `arXiv:0810.4840`, p. 668. DOI: `10.22331/q-2022-03-17-668`.

[AK07]      S. Aaronson and G. Kuperberg. "Quantum versus Classical Proofs and Advice". In: *IEEE Conference on Computational Complexity (CCC)*. 2007, pp. 115–128. DOI: `10.1109/CCC.2007.27`.

[AL18]      T. Albash and D. A. Lidar. "Adiabatic quantum computation". In: *Reviews of Modern Physics* 90 (1 Jan. 2018). `arXiv:1611.04471` , p. 015002. DOI: `10.1103/RevModPhys.90.015002`.

[AL21]      C. Ansótegui and J. Levy. "Reducing SAT to Max2SAT". In: *IJCAI* (Aug. 2021), pp. 1367–1373. DOI: `10.24963/ijcai.2021/189`.

[Alm+17]    C. G. Almudever et al. "The engineering challenges in quantum computing". In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2017, pp. 836–845. DOI: `10.23919/DATE.2017.7927104`.

[Amy+23]    M. Amy et al. "Catalytic embeddings of quantum circuits". In: *ArXiv* (2023). DOI: `10.48550/arXiv.2305.07720`.

[Ara11]     I. Arad. "A note about a partial no-go theorem for quantum PCP". In: *Quantum Information and Computing* 11.11–12 (Nov. 2011). `arXiv:1012.3319`, pp. 1019–1027.

[Aro+98]    S. Arora et al. "Proof Verification and the Hardness of Approximation Problems". In: *Journal of the ACM (J.ACM)* 45.3 (May 1998), pp. 501–555. DOI: `10.1145/278298.278306`.

[AS98]      S. Arora and S. Safra. "Probabilistic checking of proofs: a new characterization of NP". In: *Journal of the ACM (J.ACM)* 45.1 (Jan. 1998), pp. 70–122. DOI: `10.1145/273865.273901`.

[Bar89]     D. A. M. Barrington. "Bounded-width polynomial-size branching programs recognize exactly those languages in NC$^1$". In: *Journal of Computer and System Sciences (J.CSS)* 38.1 (1989), pp. 150–164. DOI: 10.1016/0022-0000(89)90037-8.

[Bau+20]   B. Bauer et al. "Quantum algorithms for quantum chemistry and quantum materials science". In: *Chemical Reviews* 120.22 (Oct. 2020). arXiv:2001.03685, pp. 12685–12717. DOI: 10.1021/acs.chemrev.9b00829.

[BC92]      M. Ben-Or and R. Cleve. "Computing algebraic formulas using a constant number of registers". In: *SIAM Journal on Computing (SICOMP)* 21.1 (1992), pp. 54–58. DOI: 10.1137/0221006.

[BCP83]     A. Borodin, S. Cook, and N. Pippenger. "Parallel computation for well-endowed rings and space-bounded probabilistic machines". In: *Information and Control* 58.1 (1983), pp. 113–136. DOI: https://doi.org/10.1016/S0019-9958(83)80060-6.

[BDS22]     S. Bisoyi, K. Dinesh, and J. Sarma. "On pure space vs catalytic space". In: *Theoretical Computer Science (TCS)* 921 (2022), pp. 112–126. DOI: 10.1016/J.TCS.2022.04.005.

[BE19]      A. Bärtschi and S. Eidenbenz. "Deterministic preparation of Dicke states". In: *International Symposium on Fundamentals of Computation Theory*. Springer. 2019, pp. 126–139. DOI: 10.1007/978-3-030-25027-0_9.

[BE22]      A. Bärtschi and S. Eidenbenz. "Short-Depth Circuits for Dicke State Preparation". In: (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.

[Ben+96]    C. H. Bennett et al. "Mixed-state entanglement and quantum error correction". In: *Physical Review A* 54.5 (Nov. 1996). arXiv:quant-ph/9604024, pp. 3824–3851. DOI: 10.1103/physreva.54.3824.

[Ben80]     P. Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22.5 (May 1980), pp. 563–591. DOI: https://doi.org/10.1007/BF01011339.

[BG22]      A. Broadbent and A. B. Grilo. "QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge". In: *SIAM Journal on Computing (SICOMP)* 51.4 (2022). arXiv:1911.07782, pp. 1400–1450.

[BH13]     F. G. Brandao and A. W. Harrow. "Product-State Approximations to Quantum Ground States". In: *ACM Symposium on Theory of Computing (STOC)* (June 2013). `arXiv:1310.0017`, pp. 871–880.

[BH17]     S. Bravyi and M. Hastings. "On complexity of the quantum Ising model". In: *Communications in Mathematical Physics* 349.1 (2017), pp. 1–45. DOI: `10.1007/s00220-016-2787-4`.

[Bis+24]   S. Bisoyi et al. "Almost-catalytic and property-catalytic Computation". In: *ArXiv* (2024). DOI: `10.48550/arXiv.2409.07208`.

[BJS10]    M. J. Bremner, R. Jozsa, and D. J. Shepherd. "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467.2126 (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 (TQC)*. 2011, pp. 35–46. DOI: `10.1007/978-3-642-18073-6_4`.

[BMT15]    J. D. Biamonte, J. Morton, and J. W. Turner. "Tensor network contractions for #SAT". In: *Journal of Statistical Physics* 160 (June 2015). `arXiv:1405.7375`, pp. 1389–1404. DOI: `https://doi.org/10.1007/s10955-015-1276-z`.

[BR60]     R. C. Bose and D. K. Ray-Chaudhuri. "On a class of error correcting binary group codes". In: *Information and control* 3.1 (1960), pp. 68–79. DOI: `10.1016/S0019-9958(60)90287-4`.

[Bra+08]   S. Bravyi et al. "Quantum Simulation of Many-Body Hamiltonians Using Perturbation Theory with Bounded-Strength Interactions". In: *Physical Review Letters* 101 (7 Aug. 2008). `arXiv:0803.2686`, p. 070503.

[Buh+14]   H. Buhrman et al. "Computing with a full memory: catalytic space". In: *ACM Symposium on Theory of Computing (STOC)*. 2014, pp. 857–866. DOI: `10.1145/2591796.2591874`.

[Buh+18]   H. Buhrman et al. "Catalytic Space: Non-determinism and Hierarchy". In: *Theory of Computing Systems (TOCS)* 62.1 (2018), pp. 116–135. DOI: `10.1007/S00224-017-9784-7`.

[Buh+24a]  H. Buhrman et al. "Quantum Catalytic Computation". In: *ArXiv* (2024). DOI: `10.48550/arXiv.2506.16324`.

[Buh+24b]   H. Buhrman et al. "State preparation by shallow circuits using feed forward". In: *Quantum* 8 (2024), p. 1552. DOI: `10.22331/q-2024-12-09-1552`.

[Bur+21]    V. v. Burg et al. "Quantum computing enhanced computational catalysis". In: *Physical Review Research* 3.3 (July 2021). `arXiv:2007.14460`, p. 033055. DOI: `10.1103/PhysRevResearch.3.033055`.

[BV93]      E. S. Bernstein and U. V. Vazirani. "Quantum complexity theory". In: *SIAM Journal on Computing (SICOMP)* 26 (1993), pp. 1411–1473.

[BV97]      E. Bernstein and U. Vazirani. "Quantum Complexity Theory". In: *SIAM Journal on Computing (SICOMP)* 26.5 (1997), pp. 1411–1473. DOI: `10.1137/s0097539796300921`.

[Cad+23a]   C. Cade et al. "Improved Hardness Results for the Guided Local Hamiltonian Problem". In: *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*. Vol. 261. `arXiv:2207.10250`. 2023, 32:1–32:19. DOI: `10.4230/LIPIcs.ICALP.2023.32`.

[Cad+23b]   C. Cade et al. "Quantifying Grover speed-ups beyond asymptotic analysis". In: *Quantum* 7 (Oct. 2023), p. 1133. DOI: `10.22331/q-2023-10-10-1133`.

[Cad+24]    C. Cade et al. "Quantum algorithms for community detection and their empirical run-times". In: *Quantum Information & Computation* 24 (2024). DOI: `10.26421/QIC24.5-6-1`.

[CFW22]     C. Cade, M. Folkertsma, and J. Weggemans. "Complexity of the Guided Local Hamiltonian Problem: Improved Parameters and Extension to Excited States". In: (July 2022). `arXiv:2207.10097`.

[Chi+20]    N.-H. Chia et al. "Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing Quantum machine learning". In: *ACM Symposium on Theory of Computing (STOC)*. `arXiv:1910.06151`. Chicago, IL, USA: Association for Computing Machinery, 2020, pp. 387–400. DOI: `10.1145/3357713.3384314`.

[CHM21]     J. Cotler, H.-Y. Huang, and J. R. McClean. "Revisiting dequantization and quantum advantage in learning tasks". In: *ArXiv* (Dec. 2021). `arXiv:2112.00811`.

[CJL08]     S. Clark, R. Jozsa, and N. Linden. "Generalized Clifford groups and simulation of associated quantum circuits". In: *Quantum Information and Computing* 8.1 (2008), pp. 106–126. DOI: `10.26421/QIC8.1-2-8`.

[CLS25]     R. Chab, F. Li, and S. Setia. "Algorithmic Techniques for GPU Scheduling: A Comprehensive Survey". In: *Algorithms* 18.7 (2025), p. 385. DOI: `10.3390/a18070385`.

[CM16]      T. Cubitt and A. Montanaro. "Complexity classification of local Hamiltonian problems". In: *SIAM Journal on Computing (SICOMP)* 45.2 (2016), pp. 268–316. DOI: `10.1137/140998287`.

[CM18]      C. Cade and A. Montanaro. "The Quantum Complexity of Computing Schatten p-norms". In: *Theory of Quantum Computation, Communication and Cryptography (TQC).* `arXiv:1706.09279`. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2018. DOI: `10.4230/LIPIcs.TQC.2018.4`.

[CM22]      J. Cook and I. Mertz. "Trading Time and Space in Catalytic Branching Programs". In: *IEEE Conference on Computational Complexity (CCC).* Vol. 234. Leibniz International Proceedings in Informatics (LIPIcs). 2022, 8:1–8:21. DOI: `10.4230/LIPIcs.CCC.2022.8`.

[CM24]      J. Cook and I. Mertz. "Tree Evaluation Is in Space O(log n · log log n)". In: *ACM Symposium on Theory of Computing (STOC).* ACM, 2024, pp. 1268–1278. DOI: `10.1145/3618260.3649664`.

[CMP18]     T. S. Cubitt, A. Montanaro, and S. Piddock. "Universal quantum Hamiltonians". In: *The National Academy of Science* 115.38 (Sept. 2018). `arXiv:1701.05182`, pp. 9497–9502. DOI: `https://doi.org/10.1073/pnas.1804949115`.

[Cob+23]    N. J. Coble et al. "Local Hamiltonians with no low-energy stabilizer states". In: *arXiv preprint arXiv:2302.14755* (Feb. 2023). `arXiv:2302.14755`.

[Coo+25]    J. Cook et al. "The Structure of Catalytic Space: Capturing Randomness and Time via Compression". In: *ACM Symposium on Theory of Computing (STOC).* 2025.

[Coo71]     S. A. Cook. "The complexity of theorem-proving procedures". In: *ACM Symposium on Theory of Computing (STOC).* 1971, pp. 151–158. DOI: `10.1145/800157.805047`.

[Dag+25]     L. Daguerre et al. "Experimental demonstration of high-fidelity logical magic states from code switching". In: *ArXiv* (2025). DOI: 10.48550/arXiv.2506.14169.

[Dat+20]     S. Datta et al. "Randomized and Symmetric Catalytic Computation". In: *CSR*. Vol. 12159. Lecture Notes in Computer Science (LNCS). Springer, 2020, pp. 211–223. DOI: 10.1007/978-3-030-50026-9\_15.

[Deu85]      D. Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[DGF22]      A. Deshpande, A. V. Gorshkov, and B. Fefferman. "Importance of the Spectral gap in Estimating Ground-State Energies". In: *PRX Quantum* 3 (4 Dec. 2022). arXiv:2007.11582, p. 040327. DOI: 10.1103/PRXQuantum.3.040327.

[Din07]      I. Dinur. "The PCP theorem by gap amplification". In: *Journal of the ACM (J.ACM)* 54.3 (June 2007), 12–es. DOI: 10.1145/1236457.1236459.

[DN06]       C. M. Dawson and M. A. Nielsen. "The Solovay-Kitaev algorithm". In: *Quantum Information and Computing* 6.1 (Jan. 2006), pp. 81–95.

[Dod+06]     Y. Dodis et al. "Syndrome Encoding and Decoding of BCH Codes in Sublinear Time". In: (2006). https://www.cs.bu.edu/ reyzin/code/bch-excerpt.pdf.

[DRS04]      Y. Dodis, L. Reyzin, and A. Smith. "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data". In: *Advances in Cryptology - EUROCRYPT 2004*. 2004.

[Dru11]      A. Drucker. "A PCP Characterization of AM". In: *ICALP* (July 2011). arXiv:1002.3664, pp. 581–592. DOI: 10.1007/978-3-642-22006-7_49.

[Dul15]      Y. Dulek. "Catalytic space: on reversibility and multiple-access randomness". In: *Personal communication* (2015).

[Fey82]      R. P. Feynman. "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21.6/7 (1982). DOI: 10.1007/BF02650179.

[Fol+25]     M. Folkertsma et al. "Fully Characterizing Lossy Catalytic Computation". In: *Innovations in Theoretical Computer Science Conference (ITCS)*. Vol. 325. 2025, 50:1–50:13. DOI: 10.4230/LIPICS.ITCS.2025.50.

[Fos+24]   M. Foss-Feig et al. "Progress in Trapped-Ion Quantum Simulation". In: (2024). arXiv: 2409.02990 [quant-ph].

[FR21]     B. Fefferman and Z. Remscrim. "Eliminating intermediate measurements in space-bounded Quantum computation". In: *ACM Symposium on Theory of Computing (STOC)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1343–1356. DOI: 10.1145/3406325.3451051.

[GC99]     D. Gottesman and I. L. Chuang. "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations". In: *Nature* 402.6760 (1999), pp. 390–393. DOI: 10.1038/46503.

[Gid25]    C. Gidney. "How to factor 2048 bit RSA integers with less than a million noisy qubits". In: (2025). arXiv: 2505.15917 [quant-ph].

[GK12]     S. Gharibian and J. Kempe. "Hardness of approximation for quantum problems". In: *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP 2012)*. 2012, pp. 387–398.

[GKS15]    A. B. Grilo, I. Kerenidis, and J. Sikora. "QMA with subset state witnesses". In: *International Symposium on Mathematical Foundations of Computer Science*. arXiv:1410.2882. Springer. 2015, pp. 163–174. DOI: 10.1007/978-3-662-48054-0_14.

[GL22]     S. Gharibian and F. Le Gall. "Dequantizing the Quantum singular value transformation: hardness and applications to Quantum chemistry and the Quantum PCP conjecture". In: *ACM Symposium on Theory of Computing (STOC)*. arXiv:2111.09079. Rome, Italy: Association for Computing Machinery, 2022, pp. 19–32. DOI: 10.1145/3519935.3519991.

[Goo+16]   I. Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[Got98]    D. Gottesman. "The Heisenberg representation of quantum computers". In: *arXiv preprint* (1998). arXiv:quant-ph/9807006.

[GP19]     S. Gharibian and O. Parekh. "Almost Optimal Classical Approximation Algorithms for a Quantum Generalization of Max-Cut". In: *LIPIcs* 145 (Oct. 2019). arXiv:1909.08846, 31:1–31:17.

[GR02]      L. Grover and T. Rudolph. "Creating superpositions that correspond to efficiently integrable probability distributions". In: *arXiv preprint* (Aug. 2002). `arXiv:quant-ph/0208112`.

[Gre+02]    F. Green et al. "Counting, fanout and the complexity of quantum ACC". In: *Quantum Information and Computing* 2.1 (2002), pp. 35–65. DOI: `10.26421/QIC2.1-3`.

[Gro96]     L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *ACM Symposium on Theory of Computing (STOC)*. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. DOI: `10.1145/237814.237866`.

[Gup+24]    C. Gupta et al. "Lossy Catalytic Computation". In: *Computing Research Repository (CoRR)* abs/2408.14670 (2024).

[Gus+23]    E. J. Gustafson et al. "Preparing quantum many-body scar states on quantum computers". In: *Quantum* 7 (Nov. 2023), p. 1171. DOI: `10.22331/q-2023-11-07-1171`.

[GY19]      S. Gharibian and J. Yirka. "The complexity of simulating local measurements on quantum systems". In: *Quantum* 3 (2019), p. 189. DOI: `10.22331/q-2019-09-30-189`.

[Hås01]     J. Håstad. "Some Optimal Inapproximability Results". In: *Journal of the ACM* 48.4 (July 2001), pp. 798–859. DOI: `10.1145/258533.25853`.

[HC17]      G. Hao Low and I. L. Chuang. "Hamiltonian Simulation by Uniform Spectral Amplification". In: (July 2017). `arXiv:1707.05391`.

[Hoc59]     A. Hocquenghem. "Codes correcteurs d'erreurs". In: *Chiffers* 2 (1959), pp. 147–156.

[HŠ05]      P. Høyer and R. Špalek. "Quantum Fan-out is Powerful". In: *Theory of Computing* 1.5 (2005), pp. 81–103. DOI: `10.4086/toc.2005.v001a005`.

[Ifr00]     G. Ifrah. *The universal history of computing: From the abacus to quantum computing*. John Wiley & Sons, Inc., 2000.

[IS20]      T. Iadecola and M. Schecter. "Quantum many-body scar states with emergent kinetic constraints and finite-entanglement revivals". In: *Physical Review B* 101 (2 Jan. 2020), p. 024306. DOI: `10.1103/PhysRevB.101.024306`.

[JGL10]     S. P. Jordan, D. Gosset, and P. J. Love. "Quantum-Merlin-Arthur–complete problems for stoquastic Hamiltonians and Markov matrices". In: *Physical Review A* 81 (3 Mar. 2010). `arXiv:0905.4755`, p. 032331. DOI: `10.1103/PhysRevA.81.032331`.

[Jia23]     J. Jiang. "Local Hamiltonian Problem with succinct ground state is MA-Complete". In: *ArXiv* (Sept. 2023). `arXiv:2309.10155`.

[JLS20]     D. Jethwani, F. Le Gall, and S. K. Singh. "Quantum-Inspired Classical Algorithms for Singular Value Transformation". In: *Symposium on Mathematical Foundations of Computer Science (MFCS)*. Vol. 170. Leibniz International Proceedings in Informatics (LIPIcs). `arXiv:1910.05699`. Dagstuhl, Germany, 2020, 53:1–53:14. DOI: `10.4230/LIPIcs.MFCS.2020.53`.

[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`.

[Jor+12]    S. P. Jordan et al. "Achieving Perfect Completeness in Classical-Witness Quantum Merlin-Arthur Proof Systems". In: *Quantum Information & Computation* 12.5-6 (May 2012). `arXiv:1111.5306`, pp. 461–471.

[Jou+17]    N. P. Jouppi et al. "In-Datacenter Performance Analysis of a Tensor Processing Unit". In: *CoRR* abs/1704.04760 (2017). arXiv: `1704.04760`.

[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. DOI: `10.48550/arXiv.quant-ph/0508124`.

[Kit97]     A. Y. Kitaev. "Quantum computations: algorithms and error correction". In: *Russian Mathematical Surveys* 52.6 (Dec. 1997), pp. 1191–1249. DOI: `10.1070/rm1997v052n06abeh002155`.

[KKR06]     J. Kempe, A. Y. Kitaev, and O. Regev. "The complexity of the local Hamiltonian problem". In: *SIAM Journal on Computing (SICOMP)* 35.5 (2006), pp. 1070–1097.

[KL98]      E. Knill and R. Laflamme. "Power of One Bit of Quantum Information". In: *Physical Review Letters* 81.25 (Dec. 1998), pp. 5672–5675. DOI: `10.1103/physrevlett.81.5672`.

[Kni+08]    E. Knill et al. "Randomized benchmarking of quantum gates". In: *Physical Review A* 77.1 (Jan. 2008). `arXiv:0707.0963`. DOI: `10.1103/physreva.77.012307`.

[Kou+25]    M. Koucký et al. "Collapsing Catalytic Classes". In: *Electronic Colloquium on Computational Complexity (ECCC)* TR25-018 (2025). DOI: `10.48550/arXiv.2504.08444`.

[KR03]      J. Kempe and O. Regev. "3-local Hamitonian is QMA-complete". In: *Quantum Info. Comput.* 3.3 (May 2003). `arXiv:0302079`, pp. 258–264.

[Kra+19]    P. Krantz et al. "A quantum engineer's guide to superconducting qubits". In: *Applied physics reviews* 6.2 (2019). DOI: `10.1063/1.5089550`.

[KSV02]     A. Y. Kitaev, A. Shen, and M. N. Vyalyi. *Classical and quantum computation*. 47. American Mathematical Society, 2002.

[Lee+23]    S. Lee et al. "Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry". In: *Nature communications* 14.1 (2023). `arXiv:2208.02199`, p. 1952.

[Lev73]     L. A. Levin. "Universal sequential search problems". In: *Problemy peredachi informatsii* 9.3 (1973), pp. 115–116.

[Liu+22]    H. Liu et al. "Prospects of quantum computing for molecular sciences". In: *Materials Theory* 6.1 (Mar. 2022). `arXiv:2102.10081`, pp. 1–17. DOI: `10.1186/s41313-021-00039-z`.

[Liu06]     Y.-K. Liu. "Consistency of Local Density Matrices is QMA-complete". In: (Apr. 2006). `arXiv:quant-ph/0604166`.

[LMT00]     K. Lange, P. McKenzie, and A. Tapp. "Reversible Space Equals Deterministic Space". In: *Journal of Computer and System Sciences* 60.2 (2000), pp. 354–367. DOI: `10.1006/jcss.1999.1672`.

[Lon01]     G. L. Long. "Grover algorithm with zero theoretical failure rate". In: *Physical Review A* 64 (2 July 2001), p. 022307. DOI: `10.1103/PhysRevA.64.022307`.

[LT20]      L. Lin and Y. Tong. "Near-optimal ground state preparation". In: *Quantum* 4 (Dec. 2020). `arXiv:2002.12508`, p. 372. DOI: `10.22331/q-2020-12-14-372`.

[LWN24]     P. Lipka-Bartosik, H. Wilming, and N. H. Ng. "Catalysis in quantum information theory". In: *Reviews of Modern Physics* 96.2 (2024), p. 025005.

[Mer23]    I. Mertz. "Reusing Space: Techniques and Open Problems". In: *Bulletin of the EATCS (B.EATCS)* 141 (2023), pp. 57–106.

[Moo+65]   G. E. Moore et al. "Cramming more components onto integrated circuits". In: (1965).

[Moo99]    C. Moore. "Quantum Circuits: Fanout, Parity, and Counting". In: *arXiv preprint arXiv:quant-ph/9903046* (1999). DOI: `10.48550/arXiv.quant-ph/9903046`. eprint: `arXiv:quant-ph/9903046`.

[MR18]     D. Maslov and M. Roetteler. "Shorter Stabilizer Circuits via Bruhat Decomposition and Quantum Circuit Transformations". In: *IEEE Transactions on Information Theory* 64.7 (2018), pp. 4729–4738. DOI: `10.1109/TIT.2018.2825602`.

[MW04]     C. Marriott and J. Watrous. "Quantum Arthur-Merlin Games". In: *CCC* (June 2004). `arXiv:cs/0506068`, pp. 275–285. DOI: `10.1007/s00037-005-0194-x`.

[MY23]     T. Metger and H. Yuen. "stateQIP = statePSPACE". In: (2023), pp. 1349–1356. DOI: `10.1109/FOCS57990.2023.00082`.

[NC10]     M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. DOI: `10.1017/CBO9780511976667`.

[NM14]     Y. Nakata and M. Murao. "Diagonal quantum circuits: Their computational power and applications". In: *The European Physical Journal Plus* 129.7 (July 2014). DOI: `10.1140/epjp/i2014-14152-9`.

[NO02]     H. Nishimura and M. Ozawa. "Computational complexity of uniform quantum circuit families and quantum Turing machines". In: *Theor. Comput. Sci.* 276.1–2 (Apr. 2002), pp. 147–181. DOI: `10.1016/S0304-3975(01)00111-6`.

[NO09]     H. Nishimura and M. Ozawa. "Perfect computational equivalence between quantum Turing machines and finitely generated uniform quantum circuit families". In: *Quantum Information Processing* 8.1 (Jan. 2009), pp. 13–24. DOI: `10.1007/s11128-008-0091-8`.

[OGo+22]   B. O'Gorman et al. "Intractability of Electronic Structure in a Fixed Basis". In: *PRX Quantum* 3 (2 May 2022). `arXiv:2103.08215`, p. 020322. DOI: `10.1103/PRXQuantum.3.020322`.

[Orú14]    R. Orús. "A practical introduction to tensor networks: Matrix product states and projected entangled pair states". In: *Annals of physics* 349 (Oct. 2014). arXiv:1306.2164, pp. 117–158.

[OT08]     R. Oliveira and B. M. Terhal. "The complexity of quantum spin systems on a two-dimensional square lattice". In: *Quantum Information and Computation* 8.10 (2008), pp. 0900–0924. DOI: 10.5555/2016985.2016987.

[PH11]     D. Poulin and M. B. Hastings. "Markov Entropy Decomposition: A Variational Dual for Quantum Belief Propagation". In: *Physical Review Letters* 106 (8 Feb. 2011). arXiv:1012.2050, p. 080403. DOI: 10.1103/PhysRevLett.106.080403.

[PM17]     S. Piddock and A. Montanaro. "The complexity of antiferromagnetic interactions and 2D lattices". In: *Quantum Information & Computation* 17.7-8 (2017), pp. 636–672. DOI: 10.48550/arXiv.1506.04014.

[PM21]     S. Piddock and A. Montanaro. "Universal qudit hamiltonians". In: *Communications in Mathematical Physics* 382 (2021). arXiv:1802.07130, pp. 721–771. DOI: 10.1007/s00220-021-03940-3.

[Pot17]    A. Potechin. "A Note on Amortized Branching Program Complexity". In: *IEEE Conference on Computational Complexity (CCC)*. Vol. 79. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 4:1–4:12. DOI: 10.4230/LIPIcs.CCC.2017.4.

[PS13]     P. Pham and K. M. Svore. "A 2D nearest-neighbor quantum architecture for factoring in polylogarithmic depth". In: *Quantum Inf. Comput.* 13.11-12 (2013), pp. 937–962. DOI: 10.26421/QIC13.11-12-3.

[PSC21]    L. Piroli, G. Styliaris, and J. I. Cirac. "Quantum Circuits Assisted by Local Operations and Classical Communication: Transformations and Phases of Matter". In: *Physical Review Letters* 127 (22 Nov. 2021), p. 220503. DOI: 10.1103/PhysRevLett.127.220503.

[PSC24]    L. Piroli, G. Styliaris, and J. I. Cirac. "Approximating Many-Body Quantum States with Quantum Circuits and Measurements". In: *Physical Review Letters* 133.23 (Dec. 2024). DOI: 10.1103/physrevlett.133.230401.

[RB01]     R. Raussendorf and H. J. Briegel. "A One-Way Quantum Computer". In: *Physical Review Letters* 86 (22 May 2001), pp. 5188–5191. DOI: 10.1103/PhysRevLett.86.5188.

[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)*. Vol. 215. Leibniz International Proceedings in Informatics (LIPIcs). 2022, 112:1–112:4. DOI: `10.4230/LIPIcs.ITCS.2022.112`.

[SB09]    D. J. Shepherd and M. J. Bremner. "Temporally unstructured quantum computation". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 465 (2009), pp. 1413–1439. DOI: `10.1098/rspa.2008.0443`.

[Sch+05]  C. Schön et al. "Sequential generation of entangled multiqubit states". In: *Physical review letters* 95.11 (Sept. 2005). `arXiv:quant-ph/0501096`, p. 110503.

[Sch+07]  N. Schuch et al. "Computational Complexity of Projected Entangled Pair States". In: *Physical Review Letters* 98 (14 Apr. 2007). `arXiv:0611050`, p. 140506. DOI: `10.1103/PhysRevLett.98.140506`.

[Sch11]   U. Schollwöck. "The density-matrix renormalization group: a short introduction". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 369.1946 (July 2011), pp. 2643–2661. DOI: `10.1098/rsta.2010.0382`.

[She06]   D. Shepherd. "Computation with Unitaries and One Pure Qubit". In: *ArXiv* (2006). arXiv: `quant-ph/0608132 [quant-ph]`.

[Sho95]   P. W. Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Physical Review A* 52 (4 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 J. Comput.* 26.5 (1997), pp. 1484–1509. DOI: `10.1137/S0097539795293172`.

[Sim97]   D. R. Simon. "On the power of quantum computation". In: *SIAM Journal on Computing (SICOMP)* 26.5 (1997), pp. 1474–1483. DOI: `10.1137/S0097539796298637`.

[SJ08]    P. W. Shor and S. P. Jordan. "Estimating Jones polynomials is a complete problem for one clean qubit". In: (2008). arXiv: `0707.2831 [quant-ph]`.

[Smi+23]  K. C. Smith et al. "Deterministic Constant-Depth Preparation of the AKLT State on a Quantum Processor Using Fusion Measurements". In: *PRX Quantum* 4 (2 Apr. 2023), p. 020315. DOI: `10.1103/PRXQuantum.4.020315`.

[Sop+22]  A. Sopena et al. "Algebraic Bethe circuits". In: *Quantum* 6 (2022), p. 796. DOI: `10.22331/q-2022-09-08-796`.

[Ste03]  A. M. Steane. "Quantum Computing and Error Correction". In: (Apr. 2003). arXiv:`quant-ph/0304016`.

[Ste96]  A. M. Steane. "Error Correcting Codes in Quantum Theory". In: *Physical Review Letters* 77 (5 July 1996), pp. 793–797. DOI: `10.1103/PhysRevLett.77.793`.

[Sus18]  L. Susskind. "Three Lectures on Complexity and Black Holes". In: (2018). DOI: `10.48550/arXiv.1810.11563`. arXiv: `1810.11563 [hep-th]`.

[Tan+24]  N. Tantivasadakarn et al. "Long-Range Entanglement from Measuring Symmetry-Protected Topological Phases". In: *Physical Review X* 14 (2 June 2024), p. 021040. DOI: `10.1103/PhysRevX.14.021040`.

[Tan19]  E. Tang. "A quantum-inspired classical algorithm for recommendation systems". In: *ACM Symposium on Theory of Computing (STOC)*. arXiv:`1807.04271`. New York, NY, USA: Association for Computing Machinery, 2019, pp. 217–228. DOI: `10.1145/3313276.3316310`.

[TD04]  B. M. Terhal and D. P. DiVincenzo. "Adaptive Quantum Computation, Constant Depth Quantum Circuits and Arthur-Merlin Games". In: *Quantum Information & Computation* 4.2 (Mar. 2004). arXiv:`quant-ph/0205133`, pp. 134–145. DOI: `10.26421/QIC4.2-5`.

[Til+22]  J. Tilly et al. "The variational quantum eigensolver: a review of methods and best practices". In: *Physics Reports* 986 (Nov. 2022). arXiv:`2111.05176`, pp. 1–128. DOI: `10.1016/j.physrep.2022.08.003`.

[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`.

[Tub+18]  N. M. Tubman et al. "Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices". In: *arXiv preprint* (Sept. 2018). arXiv:`1809.05523`.

[Tur+18]    C. J. Turner et al. "Weak ergodicity breaking from quantum many-body scars". In: *Nature Physics* 14.7 (2018), pp. 745–749. DOI: 10.1038/s41567-018-0137-5.

[Tur+36]    A. M. Turing et al. "On computable numbers, with an application to the Entscheidungsproblem". In: *Journal of Math* 58.345-363 (1936), p. 5.

[TVV23a]    N. Tantivasadakarn, R. Verresen, and A. Vishwanath. "Shortest Route to Non-Abelian Topological Order on a Quantum Processor". In: *Physical Review Letters* 131 (6 Aug. 2023), p. 060405. DOI: 10.1103/PhysRevLett.131.060405.

[TVV23b]    N. Tantivasadakarn, A. Vishwanath, and R. Verresen. "Hierarchy of Topological Order From Finite-Depth Unitaries, Measurement, and Feedforward". In: *PRX Quantum* 4.2 (June 2023). DOI: 10.1103/prxquantum.4.020339.

[VMC08]     F. Verstraete, V. Murg, and J. I. Cirac. "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems". In: *Advances in physics* 57.2 (2008). arXiv:0907.2796, pp. 143–224. DOI: 10.1080/14789940801912366.

[Vol99]     H. Vollmer. *Introduction to circuit complexity: a uniform approach*. Springer, 1999. DOI: 10.1007/978-3-662-03927-4.

[VV85]      L. G. Valiant and V. V. Vazirani. "NP is as easy as detecting unique solutions". In: *ACM Symposium on Theory of Computing (STOC)*. New York, NY, USA: Association for Computing Machinery, 1985, pp. 458–463. DOI: 10.1145/22145.22196.

[Wat+19]    A. B. Watts et al. "Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits". In: *ACM Symposium on Theory of Computing (SIGACT)*. June 2019, pp. 515–526. DOI: 10.1145/3313276.3316404.

[Wat98]     J. H. Watrous. *Space-bounded quantum computation*. The University of Wisconsin-Madison, 1998.

[WFC23]     J. Weggemans, M. Folkertsma, and C. Cade. "Guidable Local Hamiltonian Problems with Implications to Heuristic Ansatze State Preparation and the Quantum PCP Conjecture". In: (Feb. 2023). arXiv:302.11578.

[Wil25]     R. Williams. "Simulating Time in Square-Root Space". In: *ACM Symposium on Theory of Computing (STOC)* (2025). DOI: 10.1145/3717823.3718225.

[Win+23]   K. Wintersperger et al. "Neutral atom quantum computing hardware: performance and end-user perspective". In: *EPJ Quantum Technology* 10.1 (2023), p. 32. DOI: `10.1140/epjqt/s40507-023-00190-1`.

[WJB03]    P. Wocjan, D. Janzing, and T. Beth. "Two QCMA-complete problems". In: *Quantum Information & Computation* 3.6 (Nov. 2003). `arXiv:quant-ph/0305090`, pp. 635–643.

[Wol19]    R. de Wolf. "Quantum computing: Lecture notes". In: *ArXiv* (2019). DOI: `10.48550/arXiv.1907.0941`.

[Yao93]    A. C. Yao. "Quantum Circuit Complexity". In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, Nov. 1993, pp. 352–361. DOI: `10.1109/SFCS.1993.366852`.

[ZA21]     L. Zhou and D. Aharonov. "Strongly universal Hamiltonian simulators". In: *arXiv preprint* (2021). DOI: `10.48550/arXiv.2102.02991`.

[Zha24]    M. Zhandry. "Toward Separating QMA from QCMA with a Classical Oracle". In: (2024). arXiv: `2411.01718` `[quant-ph]`.

# Abstract

## Empowering Quantum Computation with: Measurements, Catalysts, and Guiding States

Over the past decade, there has been significant progress in the development of physical quantum computational devices. As of this year (2025), we are beginning to see the first implementations of quantum devices with active error correction. Nevertheless, many milestones remain before fully error-corrected quantum computation becomes available. A natural question is whether additional resources might aid in the development of these devices. In this thesis, we identify three stages in the progression of quantum devices and propose three corresponding resources that can enhance their computational capabilities. While these resources are motivated by specific stages of device development, their applicability extends beyond these regimes.

In Part One, we study the pre–error-correction regime, where computations are performed directly on physical qubits without error correction. As errors accumulate with circuit depth, this regime is effectively restricted to constant-depth circuits, severely limiting computational power (e.g., long-range entanglement generation). To mitigate this, we introduce the model of *Local Alternating Quantum–Classical Computations* (LAQCC), which augments constant-depth quantum circuits with *intermediate measurements and fast intermediate classical computations*. We show that LAQCC substantially extends the range of feasible computations beyond that of bare constant-depth circuits.

In Part Two, we consider the early fault-tolerance regime, where computations are constrained by the number of available *logical* qubits rather than runtime. Motivated by the classical catalytic space model, we define a quantum analogue in which a space-bounded quantum machine is given access to an auxiliary catalytic register, initialized in an arbitrary quantum state, which can be altered during the computation, as long as it is restored at the end of the computation. We

show that this catalytic resource extends the computational power of quantum space-bounded machines.

In Part Three, we study the problem of estimating the ground-state energy of a *Local Hamiltonian*, a central task in quantum chemistry. A common approach is to first generate a guiding state—a state with nontrivial overlap with the ground space—via a classical heuristic, and then apply Quantum Phase Estimation to approximate the smallest eigenvalue. The second step of this procedure, estimating the ground state energy given a guiding state, has been formalized as the *Guided Local Hamiltonian* problem [GL22], which is known to be BQP-hard for certain parameter regimes. We extend this result by showing that hardness persists over a broader range of parameters. Then we study an alternative version of this problem, the *Guidable Local Hamiltonian* problem, in which one is not given a guiding state, but instead only promised that it exists. We use this to give complexity-theoretic evidence that classical heuristics for generating guiding states are, in this setting, as powerful as quantum heuristics. Furthermore, we use this problem to give restrictions on possible gap-amplification procedures, analogous to [AG19], required for proving the quantum PCP (probabilistically checkable proofs) conjecture.

# Samenvatting

In de afgelopen tien jaar is er flinke vooruitgang geboekt in de ontwikkeling van zogeheten quantumcomputers. Computers die gebruikmaken van het *spooky* gedrag van de allerkleinste deeltjes. Dit jaar zijn bijvoorbeeld de eerste experimentele ruisonderdrukkende systemen geïmplementeerd. In deze systemen worden fouten die tijdens een berekening ontstaan gedetecteerd en gecorrigeerd. Zulke fouten komen vaak voor bij quantumsystemen, doordat quantuminformatie inherent kwetsbaar is.

Toch zijn we nog ver verwijderd van een volledig stabiele quantumcomputer, die gebruikt kan worden om volledige quantumberekeningen uit te voeren. Dit roept de volgende vraag op:

> Zijn er computationele hulpmiddelen die de ontwikkeling van
> quantumcomputers kunnen vergemakkelijken?

In deze thesis identificeren we drie specifieke fases in de ontwikkeling van quantumcomputers. Voor elk van die drie fases stellen we een hulpmiddel voor dat de rekenkracht van de quantumcomputer kan versterken. We onderzoeken deze hulpmiddelen met behulp van de complexiteitstheorie. Hoewel we deze hulpmiddelen in hun context bestuderen, zijn ze breder inzetbaar in de ontwikkeling van quantumcomputers.

In het eerste deel van dit proefschrift kijken we naar quantumberekeningen die nog geen gebruik kunnen maken van ruisonderdrukking. Deze fase heeft twee kenmerken. Ten eerste kunnen alle fysiek aanwezige qubits worden gebruikt om een berekening uit te voeren. Ten tweede is de toegestane diepte van de berekeningen zeer beperkt, doordat in diepere berekeningen te veel ruis ontstaat. Door deze beperking is het alleen mogelijk om constantedieptecircuits uit te voeren. Dit type circuits is zeer beperkt in zijn computationele kracht. Het is bijvoorbeeld niet mogelijk om qubits die niet bij elkaar in de buurt liggen te verstrengelen.

Het eerste hulpmiddel dat we bespreken in dit proefschrift is erop gericht om de computationele kracht van deze constantedieptecircuits te versterken. We

versterken deze circuits door ze af te wisselen met tussentijdse metingen van
een deel van de qubits en snelle tussentijdse klassieke berekeningen toegepast
op de uitkomst van deze metingen. Om de impact van dit nieuwe hulpmiddel
te bestuderen, introduceren we een nieuw computationeel model geheten: *Lo-
cal Alternating Quantum Classical Computations* (afgekort LAQCC). In dit deel
laten we zien dat er veel meer berekeningen mogelijk zijn in LAQCC ten opzichte
van constantedieptecircuits, wat aantoont dat dit hulpmiddel de computationele
kracht van constantedieptecircuits inderdaad aanzienlijk vergroot.

In het tweede deel van dit proefschrift kijken we naar de kracht van quantum-
computers die toegang hebben tot ruisonderdrukking die zich nog in een vroeg
stadium bevindt. Deze fase wordt gekenmerkt door stabiele qubits, waarop lange
berekeningen kunnen worden uitgevoerd zonder dat er fouten ontstaan, met als
limiterende factor dat een machine alleen toegang heeft tot een gering aantal
van dit soort qubits. Berekeningen in deze fase zijn niet zozeer gelimiteerd door
de tijd die het kost om ze uit te voeren, maar door het geheugen dat ervoor
nodig is. Het hulpmiddel dat we bekijken in deze situatie is het toevoegen van
een zogeheten geheugencatalysator. Dat is een extra stuk geheugen dat zich aan
het begin van een berekening in een arbitraire quantumtoestand bevindt. Deze
toestand mag door de quantumcomputer worden aangepast tijdens de bereken-
ing, maar moet aan het eind van de berekening hersteld zijn. Dit hulpmiddel
is gebaseerd op een klassieke versie hiervan, het zogeheten catalytische ruimte-
model. Om deze geheugencatalysator te bestuderen introduceren we het quan-
tumcatalytische ruimtemodel. In dit deel laten we zien dat de toevoeging van een
geheugencatalysator de kracht van quantumcomputers met een beperkt geheugen
vergroot.

In het derde en laatste deel van dit proefschrift bestuderen we een compu-
tationeel probleem dat belangrijk is in de quantumchemie, het vinden van de
grondtoestandenergie van een lokale Hamiltoniaan. Een standaardmanier om dit
probleem op te lossen is als volgt: Eerst gebruikt men een klassieke heuristische
methode om een zogeheten *guiding state* te vinden. Dit is een quantumtoestand
die een significante overlap heeft met de grondtoestand. De tweede stap is om
het quantumfasebenaderingsalgoritme toe te passen op deze toestand. De kans
dat dit resulteert in het vinden van de grondtoestandenergie is afhankelijk van de
overlap tussen de guiding state en de grondtoestand. In dit deel van het proef-
schrift bestuderen we de complexiteit van de tweede stap van deze aanpak: we
onderzoeken hoe moeilijk het is om de grondtoestandsenergie van de Hamilto-
niaan te vinden, gegeven een guiding state. Dit probleem heet de *Guided Local
Hamiltonian problem*. De complexiteit hiervan is voor het eerst bestudeerd door
Gharibian en Le Gall [GL22]. Zij tonen aan dat dit probleem BQP-moeilijk is
voor een brede set aan parameteres. In dit proefschrift laten we zien dat BQP-
moeilijkheid geldt voor een nog grotere set aan parameters.

Wij bestuderen ook het *Guidable Local Hamitonian Problem*. In dit probleem
wordt de guiding state niet gegeven als input, maar wordt wel beloofd dat zo'n

toestand bestaat. We laten zien dat dit probleem QCMA-moeilijk is. Dit geeft complexiteitstheoretisch bewijs dat klassieke heuristische methodes even goed werken voor het vinden van een guiding state quantumheuristische methodes. Verder gebruiken we dit probleem om te bewijzen dat er beperkingen zijn aan de technieken die men gebruikt om te proberen de *quantum PCP (probabilistically checkable proofs) conjecture* te bewijzen.

# Acknowledgments

I want to start by expressing my gratitude to my promoter, Harry Buhrman, for taking me on as his student, for the many interesting discussions we had, for his guidance, and for the opportunities he provided during the last four years. I would also like to extend this gratitude to my co-promoter, Kareljan Schoutens, who was especially helpful during the later stages of my PhD and played an essential role in the final organization of this thesis.

I thank the members of my doctoral committee Jonas Helsen, Christian Schaffner, Florian Schreck, Florian Speelman, and Michael Walter for taking the time to review my thesis and participate in its evaluation. I am grateful for their careful reading, thoughtful questions, and valuable feedback.

I want to express my utmost gratitude to my co-authors, for all the sparring sessions we had in front of the whiteboard, and for doing a much needed final spelling check after my writing. Ido, Chris, Jordi, Niels, Bruno, Harry, Sevag, François, Ryu, Tomoyuki, Ian, Quinten, Florian, Sergii and Sathya thank you for embarking on this journey with me. Sharing knowledge and developing new ideas together is what made science come to life.

I want to thank all those that helped me in creating this thesis. I want to thank Jasmijn for helping me with designing the cover of this thesis, Renée for helping me write the introduction and Dutch summary, and my parents, Eelco and Liesbeth, for taking the valiant effort of reading through my thesis to help me out with my spelling, even though any understanding of the content was lost after the introduction.

I want to thank all the colleagues and friends I met at QuSoft, CWI, KDVI, and Leiden over the years. You created a warm work environment where I got to pick everyone's brain for some free knowledge. I want to thank *Adam, Ailsa, Akshay, Ake, Aldo, Alicja, Aljosja, Alvaro, Amira, Anna, Arie, Arjan, Carla, ChrisC, ChrisS, DaanP, DaanS, Davi, Dima, Doutzen, Dyon, Emiel, Erik, Filippo, Florian, Fons, Fran, Freek, Galina, Garazi, Giada, Gina, Hani, Harold, Harry, Hema, Ido, Jana, Jelena, Jeroen, John, Jonas, Jordi, Jop, Joppe, Joran,*

*Julian, Kareljan, KoenG, KoenL, Krystal, Laurens, Léo, Lisa, Llorenç, Lorenzo, Luca, Ludo, Ludovico, Lynn, Mani, Marc, Māris, Martine, Matteo, Max, Maxim, Mehrdad, Michael, Minnie, Niels, Nicolas, Nikhil, Peter, Philip, Phillipe, Poojth, Quinten, Randy, Remco, René, Ronald, Salvatore, Sander, Sanne, Sarah, Sebastian V, SebastianZ, Seenivasan, Shane, Simona, Sophie, Stacey, Subha, Susanne, VictorL, VictorS, Vlad, Yanlin, Yaroslav* and *Zongbo*

I am especially grateful to my paranymphs, Llorenç and Jelena, for supporting me in the final stages of my PhD. Llorenç, from the moment we shared nachos at the Polder it was clear to me that you would become a dear friend. I want to thank you for your ever-present joyful spirit, for inviting me into the Catalan tradition of the Tió, and for all the fun times and adventures we shared, from playing table tennis with cardboard during Covid, to escaping winter to Gran Canaria and our surf trip to Morocco. Jelena, I thank you for your constant enthusiasm—for science, for languages, for cultures—and for sharing this with me. I always enjoy our spontaneous coffee dates, the walks through the park, the Easter lunches with caviar (a Russian tradition), and hopefully soon we'll be sharing a new hobby of kitesurfing.

I also want to thank Fran for his ever-present support and kindness, and for sharing his three recipes on our storm-cruising campervan trip. I thank Adam for his calming presence and our shared conferences in the Austrian and Spanish mountain ranges, especially for those delicious fruits he got from the market in Benasque. I want to thank Sebastian for answering all my little questions at work, but mostly for our friendship over the past ten years and our coffee support sessions in times of need. I want to thank Doutzen for pointing out the QSC grant to me, which gives me the opportunity to continue pursuing a scientific career. I want to thank those with whom I shared an office directly: Niels, Jordi, Lynn, Dyon, Gina, and Daan, for making it a vibrant and open working space. I also want to thank Emiel and Alicja for reminding me of some practicality in research and for inviting me into the Leiden group. Finally, I want to thank Team V (the KDVI/CWI volleyball group), who made writing the thesis more bearable by combining days of writing with a game of beach volleyball.

I want to thank my friends and family for supporting me, enjoying life with me, and being there when needed. In particular, I want to thank my parents, Eelco and Liesbeth, for always being there when I need them and for the love they always show me. I am grateful for all the enthusiasm you show when I come home with something new that interests me, and for sharing your own passions, your creativity, and your perception of the world around you.

I want to thank Jeroen, my oldest, most loyal, and best friend[13]. We have known each other for over 30 years and have always been an integral part of each other's lives. From the early childhood sleepovers every Friday to our summer

---

[13]Even though I had to fight for the best-friend part and might have been second choice when we were 2 :P.

holiday trips now, I am very grateful to have you as my friend. I want to thank Marion for all the movie nights and toothbrush parties we shared over the years; Daniel for the creative projects and experimental cooking sessions; Syrka for cultural expanses to the theater; Madelon for reconnecting our friendship; Nenna for reigniting my passion for climbing; Chelsea for the creative outbursts we shared during our brief period of being paranymphs; Marcus for reminding me to stay grounded and yet always keep on wondering; and Simon for escaping winter with me to La Gomera and for our shared love of board games.

Finally, I want to thank my girlfriend Renée for her constant support during my difficult years, and even more for celebrating all my successes, both the little ones and the big ones. I am deeply grateful for all the time we share together, from playing board games and ordering in on January 1st to surfing with snowy mountain peaks in the background. Your enthusiasm for the small wonders of everyday life, the birds in the sky, the autumn leaves, the evening light on your yellow walls, reminds me to slow down and appreciate the world around me. Thank you for all the wonderful years we have shared together.

During my PhD, I went through a particularly rough patch of my life, as I contracted Covid and developed Long Covid. This significantly impacted my work and, to an even greater extent, my daily life, as my activity levels dropped to almost non-existent. There are several people I want to thank for helping me through this difficult period.

First of all I want to thank all of my friends and family who supported me through this difficult period. I want to thank Irma van Luntheren for helping me navigate the bureaucratic hurdles involved in taking a prolonged period of sick leave; she took most of these tasks off my hands and was always there to answer my questions. I want to thank Ronald de Wolf for giving me the extra time I needed to successfully finish my thesis. I am grateful to my mother, Liesbeth, for encouraging me to start with Zelfzorg aan Zee. Even though I initially resisted, believing I would not be able to endure the therapy, she kept insisting after seeing the positive results experienced by a friend's daughter.

Most of all, I want to thank Tijs van Bezij, Rosalie Denneman, the volunteers in the program, and all the other participants of Zelfzorg aan Zee, for not seeing me as a patient but as a person, and for guiding me through their innovative surfing-therapy program. This program set me on the road to recovery: it gave me the tools to keep progressing, taught me the value of the sea, instilled in me a passion for surfing, and, above all, gave me my life back. Without this unique therapy, I might still be in the situation I was in over a year ago, and this thesis would not exist.